Toward Unsupervised Human Activity and Gesture Recognition in Videos

Farhood Negin

UNIVERSITÉ COTE D'AZUR

ECOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE

pour l'obtention du grade de

Docteur en Sciences

de l'Université Cote d'Azur
Mention: INFORMATIQUE

presentée et soutenue par

Farhood NEGIN

Vers une reconnaissance des activités humaines non supervisées et des gestes dans les vidéos

Thèse dirigée par: François BRÉMOND

INRIA Sophia Antipolis, STARS

soutenue le 15/10/2018

Jury :

Président :	Frédéric PRECIOSO	-	University of Cote d'Azur
Rapporteur :	Christian WOLF	-	INSA de Lyon
Rapporteur :	François CHARPILLET	-	INRIA Lorraine
Examinateur :	Matthieu CORD	-	UPMC - Sorbonne Universities
Directeur de thèse :	François BRÉMOND	-	INRIA (STARS)

UNIVERSITY COTE D'AZUR

DOCTORAL SCHOOL STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

P H D T H E S I S

to obtain the title of

PhD of Science

of the University Cote d'Azur

Specialty : COMPUTER SCIENCE

Defended by

Farhood NEGIN

Toward Unsupervised Human Activity and Gesture Recognition in Videos

Thesis Advisor: Francois BREMOND

prepared at INRIA Sophia Antipolis, STARS Team

defended on October 15, 2018

Jury :

President :	Frederic PRECIOSO	-	University of Cote d'Azur
Reviewer :	Christian WOLF	-	INSA de Lyon
Reviewer :	Francois CHARPILLET	-	INRIA Lorraine
Examinator :	Matthieu CORD	-	UPMC - Sorbonne Universities
Advisor :	Francois BREMOND	-	INRIA (STARS)

VERS UNE RECONNAISSANCE DES ACTIVITÉS HUMAINES NON SUPERVISÉES ET DES GESTES DANS LES VIDÉOS

Farhood Negin
Directeur de thèse: Francois Bremond
STARS, Inria Sophia Antipolis, France

RÉSUMÉ

L'objectif principal de cette thèse est de proposer un framework complet pour une découverte, modélisation et reconnaissance automatiques des activités humaines dans les vidéos. La reconnaissance d'activités humaines est la capacité à détecter et analyser automatiquement des activités humaines à partir des informations extraites et requises par des capteurs (par exemple: une séquence d'images capturées par une caméra RGB). Malgré les efforts énormes, la reconnaissance d'activités reste un domaine de recherche dynamique en raison des défis/challenges majeurs qui restent à surmonter. Parmi les défis à relever, on cite la grande complexité des activités humaines, où la variabilité du point de vue, l'apparence, et les variations de modèles de mouvement sont les problèmes les plus importants. La segmentation temporelle et spatiale d'une action dans les vidéos, la modélisation sémantique des activités et des sous-activités ainsi que l'obtention et le traitement de données sont d'autres défis notables. D'abord, nous examinons et évaluons les techniques déjà existantes dans le domaine (état-de-l'art). Ensuite, nous proposons notre framework de reconnaissance d'activités supervisée, développé en fonction des caractéristiques géométriques, des descripteurs locaux et des caractéristiques profondes, pour produire des lignes de base. Ces frameworks suivent des mots-clés/groupes de mots conventionnels et des pipelines basés sur des vecteurs de Fisher pour représenter et modéliser les activités. Afin de modéliser et de reconnaître des activités dans des vidéos à long terme, nous proposons aussi un framework qui combine des informations perceptuelles globales et locales issues de la scène, et qui construit, en conséquence, des modèles d'activités hiérarchiques. Tout d'abord, nous créons un modèle de scène basé sur les informations de trajectoires acquises. Les modèles de scène contiennent des informations contextuelles décrivant des régions intéressantes dans un environnement avec une sémantique spatiale basique. En utilisant les modèles de scène créés, nous construisons un niveau intermédiaire d'événements primitifs pour permettre l'interprétation des informations de bas niveau. La représentation de l'événement primitif fournit une description significative du mouvement global dans la scène. En se basant sur des modèles de scène créés dans plusieurs résolutions et représentations d'événements primitifs, une méthode basée sur des modèles est utilisée pour découvrir les activités de niveau supérieur. Pour compléter la procédure de modélisation, nous extrayons plusieurs descripteurs et les combinons avec les informations collectées concernant les activités découvertes. Nous proposons deux catégories du framework pour combiner ces informations. Dans la première catégorie du framework, en utilisant les descripteurs extraits, un classificateur supervisé basé sur le vecteur de Fisher est formé et les étiquettes sémantiques prédites sont intégrées dans les modèles hiérarchiques construits. Dans la seconde catégorie, pour avoir un framework complètement non supervisé, plutôt que d'incorporer les étiquettes sémantiques, les codes visuels formés sont stockés dans les modèles. Nous proposons une méthode de reconnaissance probabiliste qui trouve les occurrences d'activités similaires aux activités modélisées dans des vidéos non vues. Enfin, nous évaluons les frameworks proposés sur deux ensembles de données réalistes sur les activités de la vie quotidienne (Activity Daily Living) enregistrées auprés des patients dans un environnement hospitalier. En outre, pour modéliser des mouvements fins du corps humain, nous proposons quatre différents frameworks de reconnaissance de gestes où chaque framework accepte une ou une combinaison de différentes modalités de données en entrée. Nous évaluons les frameworks développés dans le contexte du

test de diagnostic médical, appelé Praxis.

Le test Praxis est un test diagnostique basé sur les gestes, il a été accepté comme diagnostic révélateur de pathologies corticales telles que la maladie d'Alzheimer. Malgré sa simplicité, ce test est souvent sauté par les cliniciens. Avec les méthodes proposées, nous étudions les gestes statiques et dynamiques du haut du corps basés sur le test de Praxis et leur potentiel dans un framework médical pour automatiser les procédures de test, pour l'évaluation cognitive des personnes âgées assistée par ordinateur. Afin de réaliser la reconnaissance gestuelle ainsi que l'évaluation correcte des performances, nous avons collecté un nouveau groupe de données vidéo gestuelle RGB-D relevé par Kinect V.2, qui contient 29 gestes spécifiques suggérés par les cliniciens et enregistrés à la fois par des experts et des patients effectuant ce jeu de gestes. Avec cet ensemble de données, nous proposons un nouveau défi dans la reconnaissance gestuelle qui consiste à obtenir une opinion objective sur les performances correctes et incorrectes de gestes très similaires. Notre framework basé sur l'apprentissage en profondeur (Deep Learning) proposé, apprend la dynamique des gestes du haut du corps en considérant les vidéos comme des séquences de clips à court terme des gestes. Au début, notre approche utilise la détection des parties du corps pour extraire les morceaux d'image entourant les mains. Ensuite, à l'aide d'un modèle de réseau neuronal convolutif (CNN) affiné, il apprend des caractéristiques de main profonde qui sont ensuite liées à une longue mémoire à court terme pour capturer les dépendances temporelles entre les trames vidéo. Nous rapportons les résultats des expériences sur quatre méthodes développées. Les expériences montrent l'efficacité de notre approche basée sur l'apprentissage en profondeur dans la reconnaissance des gestes et les tâches d'évaluation de la performance. La satisfaction des cliniciens à partir des rapports d'évaluation indique un fort impact du framework correspondant au diagnostic.

TOWARD UNSUPERVISED HUMAN ACTIVITY AND GESTURE RECOGNITION IN VIDEOS

by
Farhood Negin
Supervisor: Francois Bremond
STARS, Inria Sophia Antipolis, France

ABSTRACT

The main goal of this thesis is to propose a complete framework for automatic discovery, modeling and recognition of human activities in videos. Human activity recognition is the ability to automatically detect and analyze human activities from extracted information captured by sensors (E.g. sequence of images captured by RGB camera). In spite of the enormous efforts, activity recognition still remains as a dynamic research field due to the major challenges which yet to be overcome. Among the challenges being faced, the high complexity of human activities such as variability in view point, appearance and motion pattern variations are the most important ones. Moreover, temporal and spatial segmentation of an action in videos, semantic modeling of the activities and sub-activities as well as obtaining and handling data are other notable challenges.

First, we review and evaluate the prominent and state-of-the-art techniques in the filed. Then, we propose our supervised activity recognition framework developed based on the geometrical features, local descriptors and deep features to produce baselines. These frameworks follow conventional bag-of-words and Fisher vector based pipelines to represent and model the activities.

In order to model and recognize activities in long-term videos, we propose a framework that combines global and local perceptual information from the scene and accordingly constructs hierarchical activity models. First, we create a scene model based on the acquired trajectory information. The scene models contain contextual information describing interesting regions in an environment with basic spatial semantics. Using the created scene models, we build an intermediate level of Primitive events to enable interpretation of low-level information. The Primitive event representation provides a meaningful description of the global motion in the scene. Based on the created scene models in multiple resolutions and Primitive event representations, a pattern-based method is used for discovering higher level activities. To complete the modeling procedure, we extract multiple descriptors and combine them with the collected information regarding discovered activities. We propose two variations of the framework to combine this information. In the first variation of the framework, using the extracted descriptors, a supervised classifier based on Fisher vector is trained and the predicted semantic labels are embedded in the constructed hierarchical models. In the second variation, to have a completely unsupervised framework, rather than embedding the semantic labels, the trained visual codebooks are stored in the models. We propose a probabilistic recognition method that finds occurrences of similar activities to the modelled activities in unseen videos. The proposed frameworks are capable of online recognition of activities thanks to the learned scene regions. Finally, we evaluate the proposed frameworks on two realistic Activities of Daily Living (ADL) datasets recorded from patients in a hospital environment.

Furthermore, to model fine motions of human body, we propose four different gesture recognition frameworks where each framework accepts one or combination of different data modalities as input. We evaluate the developed frameworks in the context of medical diagnostic test namely Praxis. Praxis test is a gesture-based diagnostic test which has been accepted as a diagnostically indicative of cortical pathologies such as Alzheimer's disease. Despite being simple, this test is oftentimes skipped by the clinicians. With the proposed methods, we investigate the static and dynamic upper-body gestures based on the Praxis test and their potential in a medical framework to automatize the test procedures for computer-assisted cognitive assessment of older adults.

In order to carry out gesture recognition as well as correctness assessment of the performances, we have collected a novel challenging RGB-D gesture video dataset recorded by Kinect V.2, which contains 29 specific gestures suggested by clinicians and recorded from both experts and patients performing the gesture set. With this dataset, we suggest a new challenge in gesture recognition which is to obtain an objective opinion about correct and incorrect performances of very similar gestures.

Our proposed deep learning based framework learns the dynamics of upper-body gestures by considering the videos as sequences of short-term clips of the gestures. At first, our approach uses body part detection to extract image patches surrounding the hands. afterwards, by means of a fine-tuned convolutional neural network (CNN) model, it learns deep hand features which are then linked to a long short-term memory to capture the temporal dependencies among the video frames. We report the results of the experiments on four developed methods. The experiments show effectiveness of our deep learning based approach in gesture recognition and performance assessment tasks. Satisfaction of clinicians from the assessment reports indicates a high impact of the framework corresponding to the diagnosis.

Dedicated to
my wife Masoumeh, who made a big sacrifice and postponed her Ph.D. to accompany me on this journey
and to my family for their love and support.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my supervisor François Brémond for the patience, constant support, encouragement and advice he has provided throughout my Ph.D. I feel extremely lucky to have him during this time not only as my supervisor but also as a mentor and a friend. I am so grateful to him for giving me this opportunity and for his trust in my work.

I would also like to thank my thesis reviewers, Christian Wolf, François Charpillet and my examinator Matthieu Cord, that kindly agreed to review the manuscript and provided me with their valuable opinion and suggestions. Many thanks to president of the jury, Frédéric Precioso, for his time and availability.

A part of this Ph.D. thesis has been done in Computer Vision Center (CVC) at Universitat Autònoma de Barcelona (UAB). I appreciate all the help and support provided by my host associate professor Jordi Gonzales and the technical support by Pau Rodriguez during my visit from their lab.

A big part of experiments in this manuscript is conducted on data that is provided by Institute Claude Pompidou and Nice Hospital. I would like to thank professor Philip Robert and doctor Jérémy Bourgeois for their help and assist in the medical aspect of the work and also data acquisition part.

A special thank to Michal Koperski, Serhan Cosar, Carlos Crispim-Junior and Adlen Kerboa for their technical and scientific help and support. Another special thank goes to all of my colleagues in the STARS team at INRIA for their collaboration and friendship.

Last but not least, my sincere gratitude goes to my family for their encouragement to follow my dreams. I would like to thank my wife, Masoumeh who postponed her Ph.D. and moved with me to France. I am so happy that she will continue her studies. I want to thank her for all the love and support she put during this time. A special thank goes to my older brother assistant professor Masoud Negin, the first person who taught me to have a scientific world-view and inspired me to pursuit a Ph.D. in science. Finally, I would like to thank my other brother and my parents for their unconditional love and constant support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

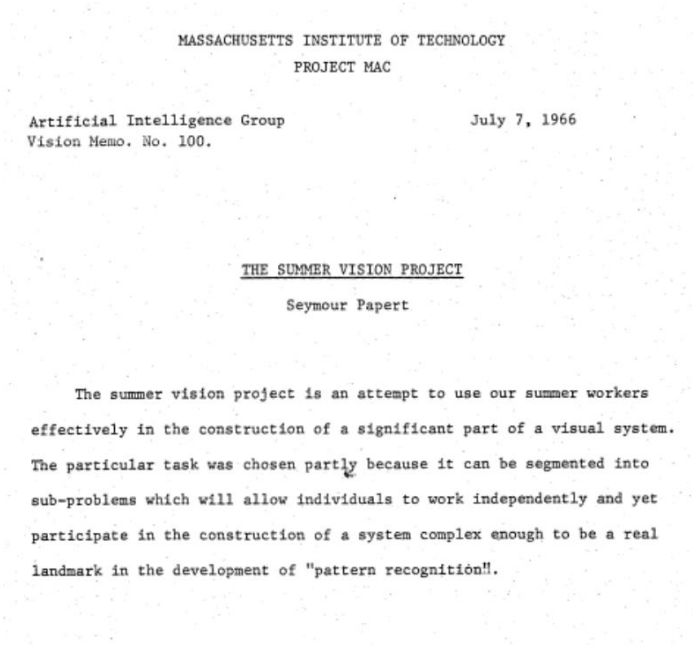
"The goal is to turn data into information, and information into insight."
- Carly Fiorina

Contents

In this chapter, we introduce the topic of this Ph.D. thesis which is modeling of human activities in videos using hybrid methods combining supervised and unsupervised methods. Another related topic investigated in this manuscript is analyzing human gestures (Section 1.1). We present the motivation for the current work in section 1.2 and the problem statement in section 1.3. Then, we explain the current research challenges in activity and gesture recognition (Section 1.4), and we also present our main contributions regarding these topics (1.5). Finally, we conclude this chapter with the thesis structure by giving a brief explanation of the upcoming chapters (Section 1.6).

1.1 Introduction

In recent years, there are a lot of discussions about artificial intelligence which transforms our lives and the world around us by automating physical and perceptual tasks. To bring



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Figure 1.1: The first page of the MIT summer vision project in 1966 describing the assignments and goals of the project.

such systems into existence, a computer must be taught to "see" its surroundings. It seems trivial to equip a machine with sight by simply attaching a webcam to it. However, it turned out that vision is human's most complex cognitive ability. Therefore, to develop an intelligent system, the machine must not only be capable of seeing things in the surrounding environment but to "understand" it as well. The machine must be able to extract an entirely new high-level of perceptual information from the raw low-level data that it receives. This understanding must be robust and reliable as the machine should make decisions based on them and accordingly, act upon those decisions.

Most of the human experience of the surrounding world comes intensely through the visual system. Researchers found out that almost half of the power used by the brain is spent on the processing of perceptual information in the visual cortex[1].

The quest for designing such systems started in 1966 when MIT professor Seymour Papert asked his students in a summer project [1] to connect a camera to a computer and construct a system of programs describing the recordings by dividing images into "likely objects", "likely background areas", and "chaos" (Figure 1.1).

[1] http://news.mit.edu/1996/visualprocessing

Nowadays, knowing the big impact of visual communication, a world without videos has become unimaginable. Videos are ubiquitous, popular, and easily accessible. Moreover, affordable and high-quality video cameras have increasingly become an integral part of our lives. Videos therefore can convey information in a clear, consistent, and unified way. Moreover, sharing and retention of information become more convenient.

Until recently, most of the video contents were inefficiently handled by supervision of human operators. With the rapid growth in the number of stored video content and easier access to enormous amounts of video data, devising an automatic and efficient solution for understanding their content becomes indispensable. Consequently, cognitive systems based on visual understanding can take their place in the central position of an intelligent system providing qualitative analysis of the perception of information. In collaboration with other interdisciplinary domains such as machine learning and data mining, these systems can overcome the limitation of human agents in analyzing and recognizing activities in videos. The technical and scientific progress have been resulted in a rapid pace of evolving computer chips which are available at cheaper prices and faster processing powers. Such systems are currently applicable and can perform video understanding and activity recognition with much faster rate than any human can do.

To address these challenges, in this thesis, we propose a complete framework for recognizing human activities in long-term videos. The framework enables automatic discovery and modeling of human activities from training videos and accordingly, recognition and evaluation of unseen and diverse test videos recorded in realistic settings. We investigate all the constituent steps composing an automatic recognition system by tying low-level visual features to semantic interpretation of activities. Additionally, our contribution has a social impact mainly by aiming to resolve a real-world problem regarding health care of older people. The developed framework aims to improve their quality of life by monitoring subjects in nursing homes as well as to help doctors in the early diagnosis of cognitive disorders. Our ultimate goal is to apply our research to real social problems.

1.2 Motivation

Human action and activity recognition have gained lots of attention owing to its extensive domain of applicability. It can be employed as a solution for the problems arising in Video Retrieval, Video Surveillance, Health Care, Human-Computer Interaction as well as in Entertainment Industry.

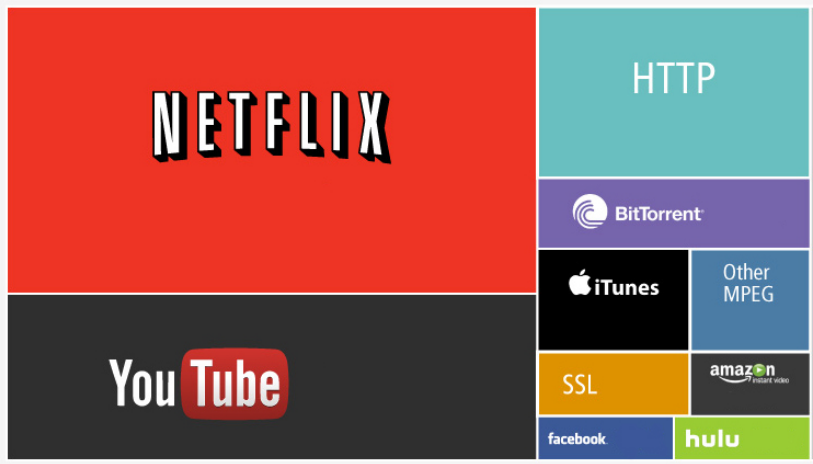


Figure 1.2: The rise of video streaming services has led to exponential growth of data traffic in the past few years. Services such as Netflix, YouTube, iTunes, Amazon and Hulu are among the biggest traffic hogs, accounting for almost 74% of global IP traffic [44].

Video Retrieval

In the past decade, there has been a significant change in the digital content landscape. Online video consumption has become the most popular internet activity worldwide. Today, online video is accounted for about 74% of all online traffic and it is projected that by 2019, it will claim more than 80% of all the web traffic. Plus, the most important strategy in today's content marketing is video. Along with that, the landscape of digital world is also changing. People are phasing out desktop and laptop computers to more portable devices such as mobile phones or tablets. These cheap and all-present devices coupled with high-speed internet access, virtually define no boundaries on how, when, and where the user can interact with the content (Figure 1.2). This phenomena makes uploading and sharing the video content appealing and popular while 300 hours worth of video content are pushed to YouTube every minute. YouTubers watch 6 billion hours of videos every month that is equivalent to watch 4 billion videos every single day [2]. Thanks to its sharing capabilities and news feed content update algorithms, Facebook surpasses YouTube with an overwhelming number of 4 billion daily video streams.

As the size of these video data banks grows, video understanding helps to effectively organize this huge data collection and assists users in quick retrieval of the content by putting forward favorable suggestions. Automatic video content analysis and retrieval, in particular activity analysis, have become extremely important in designing and implementation of such systems.

[2] https://www.youtube.com/yt/about/press/

Figure 1.3: Shows different types of surveillance cameras installed in public and private spaces in the cities and a sample of a monitoring center.

Video Surveillance

Surveillance cameras or closed-circuit televisions (CCTV) are becoming an inseparable feature of our lives. Driven by fear of terrorism, violence, disorder, and theft, and availability of low-cost hardware, video surveillance systems are widely deployed all over the places from metro stations and airports to banks and shopping malls (Figure 1.3). According to IHS [96], in 2014, there were 245 million active and operational professionally installed video surveillance cameras globally. Currently, it is estimated that only in London, there are 500,000 surveillance cameras installed such that a Londoner is caught on these cameras 300 times a day on average.

With the proliferation of surveillance cameras, it is almost impossible to actively or passively monitor this huge amount of data by trained operators within a sensible period. It has been realized that currently deployed surveillance systems are not fully efficient in any mode of operation. As a result, these systems are mostly used for the purpose of post-incident analysis rather than enhancing preventive measures.

A solution is in developing intelligent systems capable of performing automatic video analysis. Thanks to increasingly matured activity recognition systems that better scene analysis and ability to search and retrieve relevant pieces of data have become possible. Such capabilities allow distinguishing “interesting” information from “uninteresting” and consequently “abnormal” events from “normal” ones.

Human-Computer Interaction and Entertainment Industry

Figure 1.4: Examples of human-computer interaction applications in entertainment industry.

Apart from surveillance, a smart computer-based system commonly exploits video cameras to capture information of users interacting with the system. The inferred information is usually about human pose or gesture as well as the position of hands. Such information can be translated into abstract commands, comprehensible for virtually any digital system. These interactions are so intuitive and natural and the systems based on them are becoming so cheap that human-computer interaction started to be a natural part of our daily lives. Natural user interfaces are currently employed in controlling computer applications, game consoles and smart TVs (Figure 1.4). Gesture recognition and short action interpretation therefore play a key role in interacting with those systems.

Health Care: A Social Issue

There is an aging trend in world population. Almost all countries are experiencing a sharp growth in number and proportion of their older people. It is anticipated that the growth rate will be accelerated in the upcoming decade. Today, almost 10% of the world population is above 60. However, it is predicted that between 2015 and 2030, the growth rate of older people will reach 56%. This way, the population of people having 60+ age will be 1.4 billion in 2030 and 2.1 billion by 2050 which is double the number of this population in 2015 [158].

Aging is meant to be the most remarkable social transformation of the current century with a great impact on nearly all parts of the society. This impact is significant in the health care sector where innovative approaches and reformed social policies targeting this population is of paramount importance. Hence, computer-aided solutions providing automatic monitoring of older people based on activity and gesture recognition methods are becoming more important. These automatic systems are used for long-term monitoring of older people allowing them to continue living autonomously in their own houses. Moreover, such systems can be used for inferring behavioral patterns of people and helping clinicians

Figure 1.5: Shows samples of different monitoring experiments in nursing homes in France and in Greece. It also shows experiments conducted using gesture analysis for diagnostic purposes on older adults in Institute Claude Pompidou in France.

to study and propose a timely diagnosis of cognitive decline such as in Alzheimer's disease.

Accordingly, both patient monitoring and evaluations are in the great interest of our studies in this thesis. By studying a person's behavioral patterns (activities over days and weeks), doctors can have a better understanding of senior difficulties in real life scenarios, and then propose more accurate interventions to improve their living conditions. However, the long-term monitoring of people is still a problem superficially explored in computer vision, since most people tracking algorithms are evaluated in short video sequences that span from seconds to minutes. Nevertheless, as tracking people in unconstrained scenes for long periods of time is already a challenging task, human activity recognition has been a daring topic in computer vision and machine learning for almost two decades. Many methods have been proposed for short-term action recognition, but long-term activities, such as activities of daily living (ADL), are still under investigation. Accordingly, although our proposed frameworks are general-purpose activity recognition methods, we apply them to ADLs. In addition, our gesture recognition method addresses the problem of early diagnosis of cognitive disorders. More importantly, we have also collected a challenging dataset from 60 senior people performing Praxis gestures. By that, we propose a totally new and unaddressed challenge in gesture recognition which is the automatic diagnosis of cognitive impairment by analyzing gestures. The challenge in this task rather than distinguishing different gesture classes is to distinguish which one is the correct gesture among very similar instances of the same class (Figure 1.5).

1.3 Problem Statement

Throughout the years, various taxonomies have been used to describe activity recognition. However, in different studies, terms like action, activity, gesture, pose, and behavior are usually defined and used interchangeably. In order to follow a common terminology and be able to compare different methods, in this thesis, we will use the definitions proposed by Moeslund et al. [150]. Based on the complexity of a motion, we use the following hierarchy:

Gesture
A gesture is defined as an atomic and elementary movement of a person's body part. A gesture also is referred as **action primitive**. "Putting left hand on right ear" is an example of gestures.

Action
An action is a more complex body movement than a gesture. Gestures are building blocks of an action. If several gestures (action primitives) are combined with a specific order in time, an action is built. "Walking" and "Punching" are examples of action category. Within this work, we also use **Primitive Event** and action terms in exchange.

Activity
An activity is a more complex body movement than an action. It consists of several successive actions. "Preparing a Meal" or "Playing Tennis" are examples of activity category. Activity is the most complex category usually occurs on a larger scale and typically depends on the context and environment and also objects and human interactions.

Activity/Gesture Detection and Recognition
In the activities of daily living (ADL), we should deal with long-term videos where a video V contains several activities performed in unconstrained settings. This means that unlike recognition problem in short-term videos, each video contains several complex activities and corresponds with a set of labels V_L. The goal of activity detection is to predict the delineation of the constituent activities (beginning and ending time-stamps of each activity) and activity recognition is the problem of predicting the label of a given activity in that video.

In this thesis, we address the activity and gesture recognition problem with supervised, hybrid, and unsupervised methods. In supervised/hybrid methods, we have a training set with annotated labels (we know which activity they contain and their intervals). Using the training set, we learn the activity models that later are used for recognition of activities in unseen videos. In the unsupervised method, an activity discovery (detection) process is followed by a recognition step. The evaluation of activity recognition approach is performed after mapping the detected activities on ground-truth annotations. In the gesture recognition frameworks, we follow the same protocol that we use for the supervised activity recognition approach. However, we also propose a gesture spotting mechanism.

1.4 Research Challenges

In spite of its pivotal role in human behavior analysis and immense applicability, major challenges in human gesture/action/activity recognition still remain unresolved. In this section, we discuss some of the main research challenges in this regard.

One of the main research challenges is the problem of **intra-class variation**. A gesture or an action can be performed in many different ways. Complexity grows in daily activities while there is no fixed way of doing such activities and it is very likely that a person performs the same activity in different ways or in different temporal orders of its sub-activities. For example, one can perform the "Wave" gesture with right hand, while other one prefers to do it with the left hand or one person "pours water" to the coffee machine *before* "putting coffee" in it or vice versa. The designed algorithms other than being discriminative to distinguish instances of different classes, they need to be generic enough to cope with intra-class variation as well. If the designed recognition algorithm depends heavily on a specific feature modality (e.g. human skeleton), failure in **detection** of that feature will be the weak spot of that algorithm. Any failure in detection algorithm will result in failure of the recognition algorithm. The recognition algorithm should be robust enough to deal with noisy detection (false positives) as well as miss-detection (false negatives). When the targeted recognition task becomes more complex (such as in ADLs), space and time features will be required to capture characteristics of both shape and motion in videos in order to provide an independent representation of the events. Modeling the **Spatio-temporal association** of the complicated activities is very important in designing a robust recognition framework. In addition, variation in viewpoint, scale and appearance of the subjects and also occlusion, noise and handling huge amount of data are among the most important problems which make analysis and recognition of human activities a challenging research topic.

Figure 1.6: Example of ambiguous behavior in traffic junction. The trajectories are not enough to describe the situation semantically. Right: Normal behavior; Left: Abnormal behaviour.

1.4.1 Semantic Gap Problem

To generate proper semantic descriptions of a scene, given a set of the low-level descriptors, one needs to bridge the semantic gap. The semantic gap in scene understanding is the problem of the lack of a correlation between the extracted visual information by a recognition system and the semantic meaning (interpretation) which a user is looking for in a given situation [213]. As an example, figure 1.6 shows trajectory of cars in a traffic junction. The possible interpretation of the extracted trajectory features is either "Normal" or "Abnormal". The trajectories of the car in the left figure 1.6 shows normal behavior of the car turning the junction when the light is green. The trajectory of the car in the right image demonstrates abnormal behavior when a car turns the junction while the light is red. Nevertheless, further information for complete semantic description of both situations in the scene is required. Using more information may help for a better understanding of the situation. Knowing the status of the traffic light (contextual knowledge) in this example helps to describe abnormal scenarios. In activity recognition, the gap exists between low-level video descriptors and high-level semantic descriptions (or labels) of the given video (Figure 1.7). An intelligent vision-based system should be capable of capturing that quantitative information and providing qualitative (semantic) interpretation out of them. Usually, to tackle this problem, two categories of strategies have been proposed in the literature. In the **Top-Down** approaches, various techniques are used to define ontologies that reflect experts' expectation of what information will be extracted from data. Top-down methods try to plan all of the connections of system's sub-parts to produce a complete model of a scene. To reach this goal, extensive contextual knowledge should be provided a priori to the system. Generally, top-down approaches are fully **supervised** which makes them inefficient in handling complex systems such as long-term daily activities. It is difficult to define all possible variations of such activities, therefore, manually modeling those activities requires a lot of time and effort.

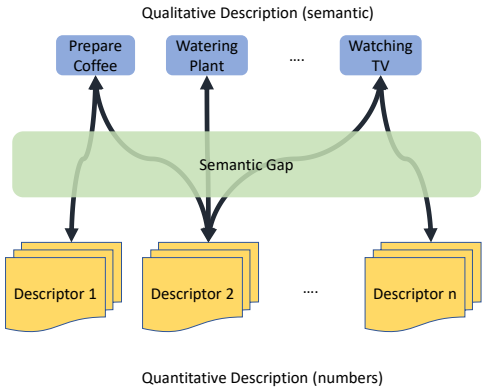


Figure 1.7: Illustrates the semantic gap problem in activity recognition.

Conversely, **Bottom-Up** approaches usually focus on particular application (e.g. surveillance, traffic, etc.). Since their goal and context are well-defined, the main task will be to extract and handle relevant information of the given content. Having prior knowledge of the prospective data, these approaches are usually **unsupervised** with a data-driven learning procedure. These methods generate models from the extracted features that later will be used for detecting events in unseen low-level data by comparing and matching the models. Due to the lack of a proper semantic layer, these methods can associate only the generated models to the similar unseen data, hence, they are unable to retrieve semantic interpretations.

For activity recognition problem, we propose a hybrid method by taking benefit of advantageous aspects of both methods. In a data-driven process, we automatically process low-level feature (bottom-up, unsupervised) until it can be semantically describable. Then, we use a supervised method to obtain semantic information about the discovered activities (labels). Generating activity models is finalized by combining the two sources of information embedded in a single hierarchical activity model. The two levels of information (high and low) are associated by creating a mid-level layer out of action primitives learned directly from data. Interpretation of the activities in unseen videos is achieved through comparing the generated model by newly discovered activities.

1.5 Contributions

To overcome the limitations of gesture and activity recognition, we propose different methods that can be applied to any gesture and activity recognition problem. However, our focus in this work is on daily living activities and its application in health care and early

diagnosis of cognitive impairments. Our main contributions can be summarized by the following items:

- **Unsupervised scene modeling and activity discovery:** In order to characterize scene regions with a higher prior probability of occurring interesting activities, we propose an unsupervised method building model of the scene in multiple resolution layers. The proposed method is capable of analyzing and updating the created models and modifying (merge/split) them if required. To discover higher level activity patterns, we extract primitive events in different resolutions using the learned scene model. Combination of primitive events creates bigger events equivalent to the high-level activities. Moreover, the scene model approach successfully overcomes the problem of temporal localization of activities existing in sliding window approaches. Using this model, beginning and ending of activities can be extracted with acceptable precision (Chapter 4 and chapter 5).

- **Dynamic length unsupervised temporal segmentation:** Rather than using ground-truth intervals in supervised learning or fixed size temporal segmentation of the input videos, we perform video segmentation using the scene region models [162, 163]. Using global motion information, we detect each time a subject enters or exits a scene region. Accordingly, we break down a long-term video into shorter clips of variable lengths (Chapter 4).

- **Hierarchical Activity Models:** Having multiple spatial layers of abstraction, we use these layers to construct a hierarchical model for activities. The process of model construction starts with extracting and collecting descriptors at different layers and ends with combining the collected information in a hierarchical tree structure. Each layer in the tree contains descriptive information of the corresponding resolution in the scene model (Chapter 4).

- **Generating Hybrid Activity Models:** We employ the proposed activity modeling approach to describe the discovered activities. The generated hierarchical models are data-driven that intake global motion information containing no annotations (unsupervised). To inject higher level semantics into the generated models, extracted local motion descriptors of the discovered activities are used to train a classifier (supervised) [163]. The annotations coming from the supervised classifier combined with the unsupervised hierarchical models generate a hybrid model suitable for describing observed activities in the scene (Chapter 5).

- **Generating Unsupervised Activity Models:** We also used the hierarchical models to develop a totally unsupervised activity recognition framework [162]. After creating

hierarchical models using the global motion information, unlike the hybrid framework, we continued with un-annotated local motion descriptors by training a visual vocabulary (dictionary) for every scene region. The trained dictionaries, combined with the hierarchical models, generate the final activity models (Chapter 5).

- **Zone Refinement with Hand-crafted Activity Models:** The created scene regions based on the unsupervised approach may be inaccurate and far from ideal segmentation formation. A region can be smaller or bigger than the expected area for a specific activity. If it is bigger, it may also cover another activity's spatial region and needs to be separated. When smaller regions appear inside the spatial zone of a certain activity, it is better to merge small scene regions to build up one big region. To overcome this challenge, we propose a hybrid model combining unsupervised scene model creation with knowledge-based hand-crafted models. After matching the generated models by the two approach, using the supervised knowledge of the activity regions injected from the hand-crafted models, we designed a split/merge algorithm that rectifies the scene region model. The knowledge interaction between the two methods can be continued in a loop until an optimal segmentation of the regions is acquired (Chapter 4).

- **Online Activity Recognition:** The automatically created scene model enables online activity recognition [163, 162]. During the testing phase, as subject moves through the scene regions, s/he crosses the region boundaries. We perform temporal segmentation of the test videos based on the enter/exit signals. As soon as a subject exits an activity region, activity recognition procedure is triggered allowing for an online recognition routine (Chapter 4).

- **Gesture Recognition:** We have developed four different gesture recognition frameworks [165]. Each framework employs a certain information modality or combination of different modalities. The modalities include: Skeleton information, RGB image and depth map. The first framework extracts spatial skeleton features and associates them temporally using a temporal offset. The videos are represented with the extracted features and are used to train a supervised classifier for gesture recognition. In the second framework, the skeleton information is utilized to detect the hand joints. To detect the precise shape of the hands, depth maps information in the patches surrounding the hand joints are utilized. After hand detection, VGG deep features are extracted and are used for training a SVM classifier. Moreover, the framework is capable of combining image and skeleton features. The third framework is similar to a recognition method based on the local descriptors. Different types of extracted local descriptors are encoded with Fisher vector representation

and can be used for training a classifier. The fourth framework is a deep architecture. A trained CNN network is applied on the cropped patches around the hand joints to extract the deep features. Next, the features are used for training a LSTM network and predicting the gesture labels. Other than gesture classification task, we propose a different challenge owing to the application of our method. The challenge is the evaluation of "Correctness" in performing the gestures. The task of the framework is to decide whether the given gesture is performed correctly or not. In most of the gestures, the difference between a correct and incorrect performance is very subtle which it is even difficult for human observer to tell the difference (We used experts' opinions for annotations). This makes the recognition task extremely difficult (Chapter 6).

- **Assessment of Cognitive Disorders and New PRAXIS Dataset:** The proposed frameworks are used for the assessment of the cognitive status of senior subjects. The evaluations are based on the Praxis cognitive assessment test which is diagnostically an accurate estimation of the Alzheimer's disease. A poor performance can be indicative of cognitive disorder. In order to evaluate performance of the proposed methods, we have collected and released a new and challenging gesture recognition dataset based on the Praxis test. The gestures are performed by 60 senior subjects in the Memory Center at the Institute Claude Pompidou (ICP) in Nice, France. The test sessions are supervised by an animated avatar. There are 29 gestures in the dataset from which 15 are dynamic and 14 static upper-body gestures. In total, the dataset contains almost 830 minutes of video recordings. It includes skeleton and depth map information as well as RBG images. The recordings are continuous and each video contains 100 gestures in average, hence, the dataset can also be used for temporal gesture localization. The gestures are annotated by medical experts in two ways. They are annotated based on gesture class as well as gesture correctness. The two annotation modes allow evaluating the dataset from gesture recognition aspect as well as cognitive assessment. The dataset recorded from the real patients introduces lots of challenging scenarios. Detailed information about the dataset is available in chapter 6. Additionally, we have developed a user-friendly evaluation tool for the clinicians [67, 160]. It provides them with detailed information about the subjects, their performance and also about every individual gesture (Chapter 6).

- **Extensive Evaluation, Comparison and Analysis of Proposed Activity and Gesture Recognition:** On activity recognition side, we provide extensive evaluation of two baseline methods based on local features. The first baseline is based on Bag-of-words approach which is one of the most popular techniques for encoding local features. The second baseline follows Fisher vector encoding scheme which

has shown superior results in different Computer Vision tasks over the bag-of-words scheme. We also provide evaluation results based on the two proposed frameworks of the hybrid and unsupervised. Our baselines and proposed methods are applied on two activities of daily living datasets: GAADRD [102] and CHU Nice Hospital [51]. We also provide deep evaluation of the four gesture recognition methods on our recorded PRAXIS dataset. On the other hand, we perform extensive evaluation of gesture recognition framework on the recorded PRAXIS dataset. We show that the designed deep learning based architecture outperforms other baseline techniques (Chapter 5 and chapter 6).

1.6 Thesis Structure

In this chapter, we introduce the problem of activity and gesture recognition, the problem motivations and their potential applications. We also describe the main research challenges in these fields and summarize our contributions. We have divided this manuscript into seven chapters. A brief description of each chapter is as follows:

- *Chapter 2* – **Related Work:** In this chapter, we explore the available state-of-the-art methods which address our defined challenges in short and long-term activities. We follow a taxonomy dividing the approaches into single-layer (unsupervised) and hierarchical (mostly supervised). We also discuss the similar attempts to solve the problem of long-term activity discovery and their success and failures.

- *Chapter 3* – **Supervised Activity Recognition:** In this chapter, we introduce components of the supervised activity recognition baseline framework and its two variations. The main components are: feature detection, feature extraction, feature encoding and classification. The framework accepts a variety of feature types from hand-crafted geometrical features to representation learning deep features. In the first framework, bag-of-words encoding method is used as the feature representation method. In the second framework, feature encoding is carried out by Fisher vector representation. We present the two daily activity datasets (GAADRD and CHU) that we use in the following evaluations. Finally, we present the extensive evaluation of the two presented techniques, comparisons and analysis.

- Chapter 4 – **Activity Discovery, Modeling and Recognition:** In this chapter, first, we explain the process of activity discovery from the sequence of detected primitive events in different levels of resolution. This is done by extracting the global trajectory of the subjects in the scene and by creating a scene model (Topology) through

clustering of the trajectory points. Detecting the optimal number of clusters is assessed by cluster analysis. Additionally, the optimal shape and number of the scene regions are investigated by proposing a method combining knowledge-based handcrafted scene models with unsupervised data-driven scene topologies. Considering distinctive attributes, the activities are clustered and represented by hierarchical activity models. Given an unseen video, activity recognition takes place by computing its similarity with the generated activity models.

- Chapter 5 – **Hybrid Activity Recognition Framework:** In this chapter, we propose two activity recognition frameworks based on the proposed activity models in the previous chapter. The first one is a hybrid framework combining supervised and unsupervised methods benefiting from the advantages of both approaches. This framework achieved state-of-the-art performance compared to the baselines and other suggested frameworks. The second framework is an unsupervised framework. Instead of using a supervised classifier to obtain labels, the extracted descriptors are clustered and used for training a visual vocabulary embedded in the hierarchical models. An extensive evaluation using different descriptor types is presented. At the end of this chapter, we discuss the challenges we face on the evaluated datasets.

- Chapter 6 – **Gesture Recognition:** In this chapter, we present details of the proposed four gesture recognition frameworks and developed cognitive assessment application. Gesture recognition problem is investigated in the context of cognitive disorders. The frameworks are designed to accept different data modalities.

- Chapter 7 – **Conclusion and Future Work:** Here, we summarize our proposed approaches and their strengths and limitations. We also discuss the future direction and extension of our research by presenting our short-term and long-term perspectives.

Chapter 2

Related Work

"Progress is made by trial and failure; the failures are generally a hundred times more numerous than the successes, yet they are usually left unchronicled."

- William Ramsay

Contents

2.1 Introduction

In this chapter, we review various methods proposed for the problem of activity recognition in recent years. For extensive and in-depth investigation of the topic, readers can refer to the reviews available in [3, 161, 187, 24].

Over the last two decades, many approaches have been proposed for recognizing human activities from videos. Different features have been examined for robust and discriminative representation of activities. In addition, many machine learning approaches

have been applied for modeling activities and obtaining robust classifiers. The objective of activity recognition is to automatically understand and describe ongoing activities in an unseen video based on previously obtained patterns from observations in training phase. In a more simple case of the problem, the goal of a recognition system is to correctly classify activities in a long-term video that is already been divided into chunks containing only one activity instance. In a more difficult case, the delineation of activities is not given. The videos are continuous with an arbitrary order and number of activities. Even activities can take place concurrently. In such scenarios, in prior to the recognition, an additional step concerning the detection of beginning and ending of activities is required.

There is a large body of literature that focuses on activity recognition in short-term actions which the duration of actions is in the range of seconds. The recognition of long-term activities that the duration of activities varies from minutes to hours (and even days), is more challenging hence, it is less explored. The long-term activities have different characteristics and are performed in larger variation than short-term activities. Moreover, collecting such data is challenging and unlike short-term actions, they can not be recorded in a lab environment. Usually, this kind of data is collected in real-world settings where people perform activities naturally (such as in home doing daily chores). Unlike scenarios happening in structured scenes, these activities are common in unstructured environments. For example, in a traffic light scenario, the objects in the scene usually follow a limited number of pathways with a constrained spatiotemporal order. Conversely, in a daily activity scenario taking place in an apartment, activities are not strictly defined based on objects or areas in the environment. Several activities can happen in a specific area (*eg.* kitchen) with an unconstrained temporal order. Furthermore, the algorithms developed to process these activities need to be computationally efficient in order to handle a huge amount of recorded data.

Although, we are interested in various types of long-term activities such as surveillance, abnormal activity detection, etc., in the context of this thesis, we are mainly interested in a special class of long-term activities called Activities of Daily Living (ADLs). Recognition of these activities is the essential part of services developed for health care and assisted living purposes. Originally, Katz et al. [103] proposed a set of ADLs (such as dressing and feeding) to measure the biological and psychological function of people. Over the years, different variations of these activities have become a standard to evaluate the well-being of older people. Later, Lawton et al. [123] extended the set of ADLs to include activities that involve different instruments (Such as Answering phone, preparing coffee etc.) and called them Instrumental Activities of Daily Living (IADLs). Due to the complexity of ADLs and IADLs, modeling and recognition of them are challenging. Our

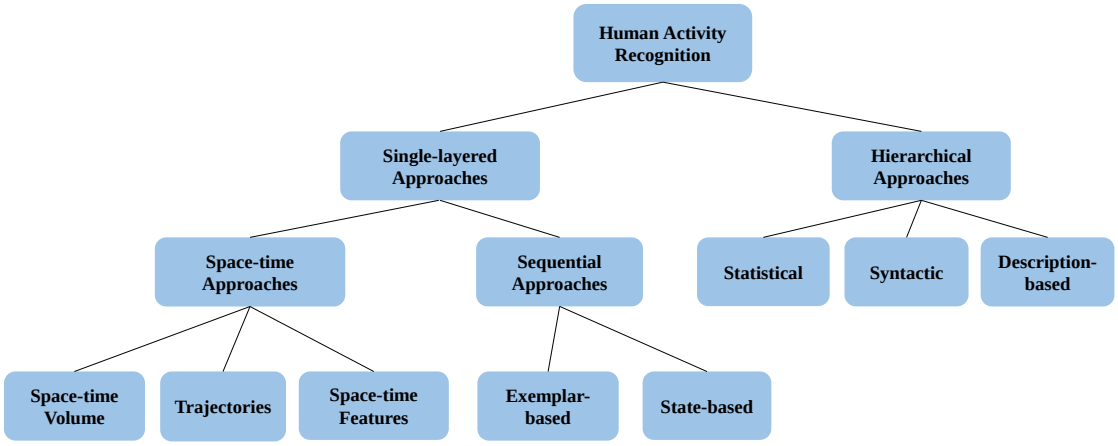


Figure 2.1: The hierarchical taxonomy of human activity recognition problem and its solution domain.

main focus in this thesis is on automatic recognition of these activities.

This chapter provides an overview of the prominent methods in human activity recognition. The goal is to explain advantages and disadvantages of those techniques and their influence on the current work. The field of activity recognition moves at a fast pace. There are many surveys summarizing the proposed methods throughout the years. We use the taxonomy proposed by [3] dividing the activity recognition methods into two main categories (Figure 2.1). **Single-layer approaches** aim to model and recognize activities directly from the given image sequences by learning the models from extracted image descriptors. Usually, these methods are successfully applied to the recognition of short actions or gestures with sequential characteristics. On the other hand, **Hierarchical approaches** represent high-level activities in terms of their simpler constituents which are referred to as sub-activities. These approaches make the construction of multi-layer models possible, hence, allowing to generate a semantic description of complex activities. Since the proposed methods in the manuscript take advantage of both sing-layered and hierarchical approaches, we use this taxonomy to explain the related work in the available literature to ours. Depending on the structure of the modeling, the single-layered approaches are divided into two categories. The *space-time approaches* consider a video as a 3D volume (XYT) where the descriptors are computed (such as short trajectories). These methods are further divided into three categories based on the features they use from 3D space-time volume for activity modeling: volumes themselves [88, 188, 193, 204, 104], trajectories [186, 92, 244, 205, 259] and local interest point descriptors [233, 84, 169, 22, 91, 75]. On the other hand, the *sequential approaches* view a video as a sequence of observations. These approaches are also divided into

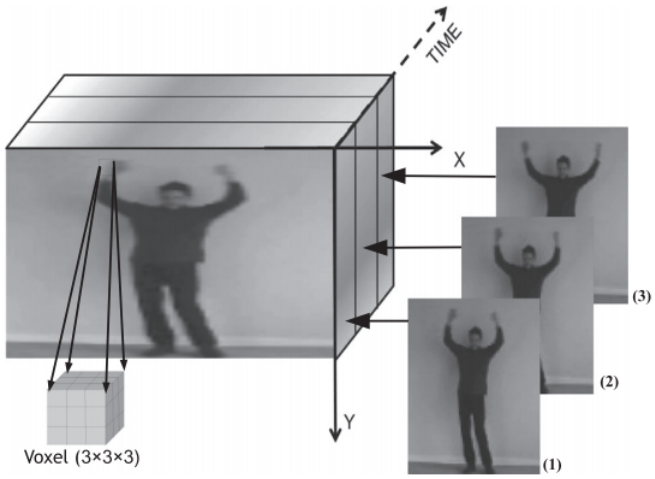


Figure 2.2: Shows the procedure of computing a Local Binary Pattern volume of "Waving" action by concatenation of the image sequences [14].

two sub-categories. Depending on the utilized recognition methodologies they are categorized either as exemplar-based [50, 72, 254, 228, 20] or model-based approaches [255, 202, 268, 214, 175, 240, 20, 231].

Hierarchical approaches are also divided into three categories based on the utilized recognition methodologies: statistical [261, 267, 170, 47, 77], syntactic [94, 241, 110, 100, 149, 151], and description-based [82, 152, 12, 189, 222, 194] approaches. In the statistical approaches, state-based models are concatenated hierarchically in order to describe high-level activities (For example, layered HMMs). In syntactic approaches, a grammar syntax such as stochastic context-free grammar (SCFG) is used to model the activities. These methods model the high-level activities with strings of atomic activities. Finally, description-based methods represent activities by describing the sub-activities and their spatial, temporal and logical relations.

2.2 Single-Layer Approaches

In the single-layered approaches, activities are directly recognized from the image sequences. These approaches interpret a set of specific image sequences as a particular class of activity. Activity recognition in unseen sequences takes place by grouping the sequence into its predefined set of classes. To make a decision that whether an image sequence belongs to an activity class or not, various representation techniques and matching methods have been suggested. In continuous sequences that activity intervals are not given, most of the single-layered approaches adopted a sliding-window approach to classify all possible sub-sequence patterns [104, 204]. To have a more effective recogni-

tion, the sequential pattern of describing activities should be captured from the training data. Intrinsically, these methods are suitable to model and recognize short and relatively simple human actions (such as "waving" and "walking"). These methods become less effective when the activities are long-term, more complex and occurring in unstructured scenes. In these scenarios long-term activities may have many variations which is challenging to characterize. These methods have to adapt a sequential model to describe such activities as a sequence of events. Two types of single-layered approaches are space-time and sequential approaches. The main difference between the two approaches is the way each method handle time. Space-time approaches model the activities using 3D volume in spatiotemporal dimension or extracting features from the volume [188, 186, 92, 244, 205, 233, 84, 22, 91]. The space-time volumes are created by concatenating the image sequences in the time axis and measuring their similarities. Sequential approaches consider an activity as ordered observations (by taking into account their sequential relationships) where the activity recognition is the process of searching for such sequences in unseen videos [254, 228, 202, 268, 214, 175, 240, 231].

2.2.1 Space-Time Approaches

An image is a two-dimensional matrix of data encoding a real 3D scene. It captures spatial relationships of humans and objects. Accordingly, a video can be defined as a sequence of those 2D images in order of time. As a result, an activity can be seen as a three-dimensional volume which is created by concatenation of two-dimensional images along the time axis.

The space-time activity recognition methods typically perform the recognition task by analyzing the detected space-time volumes [88, 188, 193, 204, 104]. A common pipeline of a space-time method follows these steps: Using the videos available in the training set, the system computes the 3D space-time volumes for given activities. This volume is stored as template volume. When a new video appears, first the space-time volume is computed and then, the extracted volume is compared to the template model volumes. Based on the appearance and shape similarities between the test and template volumes, the recognition algorithm (template-matching) determines the semantic of the performed activity obtaining the highest similarity score. Figure 2.2 shows an example of space-time volume of waving action.

Other than methods that only use 3D volumetric representation, some utilize the space-time volume differently. In one variation the recognition system represents the activity as a trajectory rather than a volume [186, 92, 244]. The most salient points in

human activity are body joints positions. If the recognition algorithm can successfully track these joints the activities can be represented more clearly using a set of trajectories. In another variation of space-time methods instead of volumetric trajectory-based representation, the activities are represented through a feature set obtained from the 3D volume or tracked trajectories. In these methods the space-time volumes can be considered as rigid objects that the extracted features can be used for characterizing them.

Apart from representation, also various recognition methods have been developed to use space-time representations. The goal of these recognition algorithms is to match volumes, trajectories or their features with learned models from training data as accurate as possible. Discriminative approaches such as Neighbor-based matching algorithms have been extensively applied to recognition task where the system describes activities by sustaining a set of samples (volumes or trajectories). In the recognition phase, the system matches the input with the stored samples. Moreover, algorithms based on statistical learning have also been developed that utilizes the probability distribution of activities in matching the samples.

2.2.1.1 Space-Time Volumes

The main challenge regarding the recognition task using space-time volumes is how to measure the similarity between the volumes. The simple way to do it will be to take the whole volume as feature or template and perform the matching for classification. Taking the entire volume as feature adds noise to the feature vectors since the features will also include redundant background information. Therefore, some methods use background extraction to obtain the meaningful motion information of a person in the foreground performing activities [19]. Using the idea of extracting foreground movement, some methods performed action recognition [88, 188, 193]. [204] proposed an approach to use patches of the volumes to compare the actions. Similarly, [104] used an over-segmented volume to model human motion. To have more reliable similarity comparisons [191] proposed a method to use filters to characterize the volumes.

Bobick and Davis [19] proposed an efficient framework based on template matching. For each action they create two 2D templates consisting Motion Energy Image (MEI) and Motion History Image (MHI). They created these templates by a weighted projection of sequence of extracted person in the foreground (Figure 2.3). Template matching performed efficiently and in real-time.

Inspired by Bobick's work Qian et al. [188] combined global and local features to achieve better accuracy of recognition. For the global feature, they used MEI images but to avoid the existing hollow part of the foreground because of the presence of human

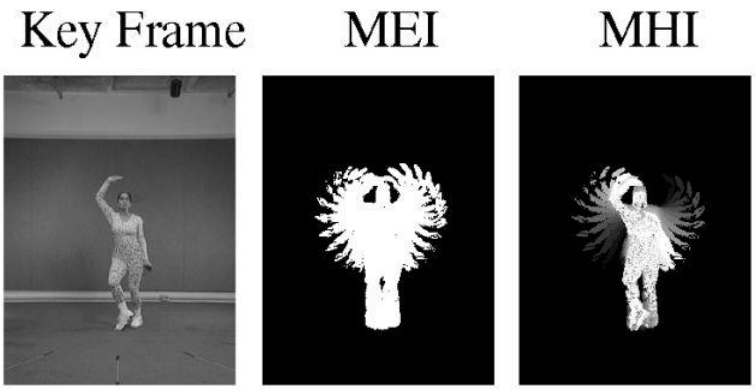


Figure 2.3: Shows the created 2D action templates from a sequence of images [19].

blob contour coding is applied on MEIs. Bounding boxes of the objects used as local features. Roh et al. [193] modified Bobick's 2D MHIs to 3D. Instead of 2D templates, 3D volumetric templates used for view-independent action recognition. Similarly, Hu et al. [88] combined MHI with appearance features to achieve a better description of human activities. They used two different appearance features. The first one is the foreground image obtained from background subtraction and the second one is Histogram of Oriented Gradients (HOG) features identifying direction and magnitude of edges and corners. They use a special SVM classifier called SMILE-SVM (simulated annealing multiple instances learning support vector machines) which looks for the global optimum. Using simulated annealing method helps the SVM classifier to avoid local optimums.

Kim et al. [108] proposed Accumulated Motion Image (AMI) based on gait energy analysis in order to represent the spatiotemporal aspect of the actions. The AMI is composed of image differences averaged over a video sequence. Based on obtained AMIs, they calculate a rank matrix. To recognize a query action, they measure the L1-norm between the temporal and spatial rank matrices and choose the model with minimum distance as recognized action.

Other researchers proposed methods for using body models for recognition of actions. These models are constructed using information such as skeleton joints or silhouettes. For example, Ikizler et al. [90] proposed a pose descriptor using silhouette information. Histogram of Oriented Rectangles (HOR) describes a body pose in an action sequence with oriented rectangular patches created by silhouettes. The histograms calculate the distribution of oriented rectangular patches. The local dynamic of actions is captured by the accumulation of HORs inside a sliding window. For recognition, they used various methods such as SVM and dynamic time warping.

In [243] Wang et al. developed a framework using Semi-latent Topic Models (STM). Inspired by the text mining methods a "word" is equivalent to a frame in an action video

and accordingly, a "document" is equivalent to the whole video. Optical flow is used for describing motion where a visual dictionary is trained by considering them as action descriptors. Developed based on Latent Dirichlet Allocation (LDA) [18] and Correlated Topic Model (CTM)[17], STM infers the number of topics and provides efficient training procedure.

Rather than viewing poses as words in a document, Guo [78] considered them as a sequence of shape deformations of silhouettes. For action representation, a covariance matrix of calculated geometrical features of silhouette tunnels is constructed and a Riemannian metric is used to measure the similarity between matrices of different actions.

Various efforts to adopt different techniques to the action recognition problem have been tried. For example, to compare the spatiotemporal video volumes and measure their similarities, Kim et al. [107] applied Canonical Correlation Analysis (CCA). Liu et al. [131] detect the repetitive parts of actions as Salient Action Units (SAU) and train an AdaBoost classifier using these features. To combine different features Cao et al. suggested Heterogeneous Feature Machines (HFM) [29].

To avoid equal treatment of all spatiotemporal volumes in videos and using irrelevant volumes for training models, Zhu et al. [269] proposed a deep network for mining of discriminative key volumes. Training of the network has been done with the forward-backward stages of Stochastic Gradient Descent (SGD). In the forward step, the key volumes are detected and the parameters of the network get updated in the backward step. They showed that mining key volumes significantly improves the accuracy of the recognition achieving state-of-the-art performances in various action datasets. [10] developed a 3D convolutional neural network that extends the conventional CNN by taking space-time volume as input.

In overall, it can be seen that the space-time volumes are calculated as a set of dense descriptor to model of short term action unit. Such methods face difficulty when applied to long-term activities. Extracting dense descriptors is accurate and feasible for short actions, however, it gets difficult and inefficient when the amount of data getting big which makes the models created by these methods inefficient to modify and difficult to update. Moreover, the recognition of actions involving multiple people is challenging especially when the spatial segmentation is not possible. It is also difficult to handle noise in these methods since the background is also captured and modeled as part of actions.

2.2.1.2 Space-Time Trajectories

Trajectory-based action recognition approaches work based on tracking of interest points such as human skeleton joints. According to Johansson [99] tracking human body joints is enough to recognize the occurring action. Therefore, various body part estimation

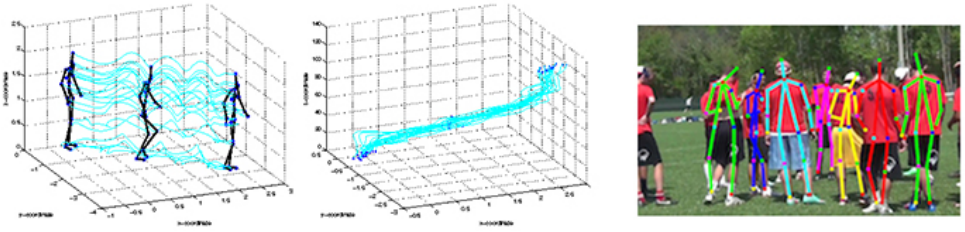


Figure 2.4: Left: shows tracking of body joints while walking in XYZ [205]; Middle: shows tracking of the joints while walking in XYT. Right: using deep learning to estimate body parts from RGB images [186].

methods [209] skeleton representation techniques and recognition methods proposed to recognize human action by matching trajectories (Figure 2.4). Most of the earlier body part estimation were based on RGB-D images. However, recent advancements in deep learning and availability of powerful GPUs made real-time extraction of skeleton information from RGB images possible [186, 92, 244].

Some of the proposed methods [205, 259] directly used the trajectories to represent and recognize the actions. To describe actions in spatiotemporal space with joint trajectories, Sheikh et al. [205] extended the three-dimensional representation of joints into four-dimensional (XYZT) space-time domain. To preserve the action representation from the variation of viewpoint they applied an affine projection to normalize the trajectories. Similarly, Yilmaz et al. [259] proposed an approach to match trajectories captured in four-dimensional XYZT space. In order to analyze trajectories, Messing et al. [148] use dense clouds of KLT [136] features extracted by tracking Harris3D interest points. The trajectories uniformly quantized in log-polar coordinates. They use velocity-history of trajectories as their basic features. To learn the velocity-history a generative model is used. To model the action classes a weighted mixture of bags-of-augmented-trajectory sequences is learned where each action model has a distribution over 100 shared mixture components. The mixture components are considered as velocity-history words. Finally, the velocity-history features are used for training a classifier to recognize the actions.

Meanwhile, many proposed trajectory-based approaches focus on representing skeleton by extracted features from joint positions. These methods usually encode information of relative joint positions or angle between the joints in consecutive frames. Wang et al. [238] calculate relative distance of a skeleton joint with all the other joints and enumerate all the pairwise joints to generate the final feature vector for skeleton representation. Rather than calculating distance in a single frame, Yang et al. [258] calculated relative

distance of joints among video frames as a feature. [168] and [201] joint angles to represent skeleton. For a better representation of angular space, [201] uses quaternion coordinates.

Most of the mentioned methods use bag-of-words (BoW) scheme for encoding actions. Inspired by methods from Natural Language Processing (NLP), this scheme ignores the temporal relations of the visual words. To model temporal relations between the poses, [143] proposed an ensemble tree models. Recently, Agahian et al. [2] proposed an approach for temporal encoding of the pose descriptors. First, a set of key poses are detected by applying K-means clustering on skeleton data in the training set. An SVM classifier is trained on the calculated cluster centers to classify action pose sequences to key pose sequences. Every action in a given dataset is encoded with a key pose histogram before going through an Extreme Learning Machine (ELM) for classification. Some of the methods reported that the contribution of joints in recognition phase differs from action to action. Hence, they provide methods to manually [253] or automatically [159] select the most informative joints. Negin et al. [164] proposed a Random Decision Forest (RDF) based feature selection mechanism. The calculated geometrical features are fed into an RDF in order to mine the most discriminative joints and feature types. The final classifier is trained by the selected salient features. Rather than modeling the temporal relations in feature level, some methods [116, 138] postpone it to the classifier phase where they train classifiers such as Hidden Markov Models (HMMs) suitable to model sequential data.

Joint trajectory representation provides a compact representation of an action which can be suitable for describing long-term activities. The described methods use these features to describe short-term actions. However, we use these methods in the context of long-term activities. Moreover, these methods suffer from occlusion problem and also robust detection and tracking of the joints.

2.2.1.3 Space-Time Features

These approaches treat space-time volume as a rigid 3D object and extract features to represent the characteristics of the volume. Such methods have been previously tried on object and scene recognition. The extracted local features use interest points and their surrounding volume to describe the space-time volume. Regarding the density of the interest points, space-time features can be divided into two categories. The Harris3D [119] and The Dollar [54] detectors are examples of *sparse* detectors. Interest point detectors based on optical flow are considered as *dense* feature detectors.

Dollar detector uses Gabor filtering to detect variation of intensity in the temporal

domain. It detects local 3D patches with complex motion. However, using only local information in a relatively small patch cause this detector to neglect transitional motions. Moreover, it is not effective when there are slow movements corresponding to the objects present in the scene or in case of camera motion or zoom. To overcome these shortcomings Bergonzio et al. [22] proposed a detector with a different design of spatial and temporal filters and their combination to produce the final response. They achieved this by extraction of holistic features from clouds of interest points detected from multiple temporal scales where the final features are automatically selected and accordingly classified by SVM and Nearest Neighbor classifiers.
Thi et al. in [220] use Harris3D to detect interest points. Next, they use a Bayesian classifier to decide whether the interest point belongs to an action of interest or not. They localize actions with weighted Conditional Random Field (CRF) and use the extracted point of interest descriptors to describe them. For classification, they use PCA-SVM classifier.

Although Harris detectors were effective to some extent, they suffer from sparsity. Knowing this, Gilbert et al. [75] used two dimensional dense Harris corner detector [119] in multiple spatial scales instead. They constructed features using the detected interest points and used a Hierarchical clustering to group actions into different classes. Similarly, Sadek et al. [195] by only using Harris corner detector described local features with fuzzy log-polar histograms. Combined with global features, the final feature vectors are classified with SVM.

Several methods [84, 169, 91] used optical-flow information to detect interest points for feature description. Ikizler-Cinbis et al. [91] used a homography-based motion compensation method to detect foreground flow in the scene related to the motion of person or objects. A set of shape features based on detected objects are extracted before giving as input to a Multiple Instance Learning algorithm to localize the interesting spatial regions in a video. Oikonomopoulos [169] also uses optical flow field to detect salient points around a spatiotemporal cylindrical geometry based on B-splines. The final B-spline polynomial descriptor is invariant to scale and transformation in space and time.

After detection of salient interest points, usually, for local representation, various descriptors around the local volume of the detected points are computed. Throughout the years, many descriptors have been proposed [48, 233, 247, 200]. Moreover, the idea of extracting descriptors around feature trajectories became popular. These methods track the detected interest points in time and compute trajectory shape and descriptors around the space-time volume of the trajectory to describe actions in videos. In recent years,

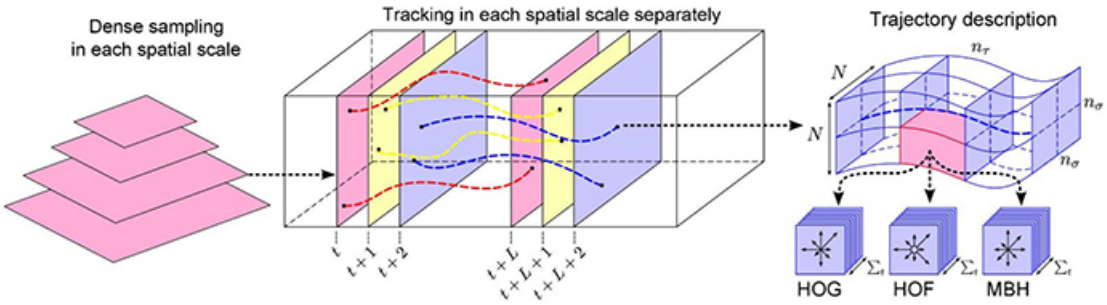


Figure 2.5: Shows the process of extracting dense trajectories proposed by Wang et al. [233]. Left: shows dense sampling to detect interest points in multiple spatial scales. Middle: illustrates tracking of the trajectory points in a given interval (L frames). Right: HOG, HOF and MBH descriptors are extracted around the retrieved trajectories.

dense trajectory-based method proposed by Wang et al. [233] has gained popularity (Figure 2.5). They sample dense interest points in each frame and track their displacement based on the optical flow field. At the end, to represent the actions, they calculate HOG [48], HOF [233] and MBH [48] descriptors in the space-time volume around the trajectories. This method resulted in high recognition accuracy and inspired others to propose various methods [15, 239, 114, 115, 16] based on this representation method and improve discriminative power of descriptors.

Depending on the complexity of actions, complexity of the relationships among local descriptors changes. To model these relationships in real-world situations, probabilistic models seem a favorable option. Inspired by probabilistic methods in NLP, using topic modeling in describing actions and events has been examined in several works [248, 243, 64]. In topic modeling, it is common to divide the videos into short clips of several frames and extract motion information such as optical flow fields to create the documents out of videos. Wang et al. [243] proposed semi-latent Dirichlet allocation (S-LDA) and semi-latent Correlated Topic Models (S-CTM) to model and classify actions. They use motion descriptors obtained from the whole frame to create the "visual words". After learning the visual dictionary, generative topic models are applied where the topics of the document are denoted as latent variables in these models. The main problem of this method is its requirement for manual and predefined configuration of the model by the user. In [248] Wu et al. proposed an unsupervised framework to understand activities and their relations. The proposed topic model method allows to model long-range dependencies between the action-topic in complex activities. By learning the temporal relations between the actions, the generated topic models can be also used to detect the forgotten actions by finding missing sub-actions learned in training phase. The main

challenge in modeling actions and activities based on topic-models is inferring the number of topics which is similar to the problem of cluster analysis in the clustering algorithms. In ideal case, the selection of topic number should be done automatically. Emonet et al. [64] proposed an unsupervised Non-parametric Bayesian methods based on Hierarchical Dirichlet Process (HDP) to discover recurrent temporal patterns of words (Motifs). The method automatically finds the number of topics, number of time they occur and the time they occur. Additionally, it can infer the length of the motifs. The method is applied in traffic scenes where it successfully obtained the interesting motifs. In spite of topic models high capability to model complex activity, due to utilization of Gibbs sampling in learning hidden variable, the learning time can be inefficient. Moreover, the generated models are complex and difficult to modify and hard to interpret. Due to nature of the utilized optical flow features, recognition of individual activity is not possible (motion of all objects in the scene are captured and modeled). Also, they need to tune some parameters such as size of sliding window to achieve the best codebook configuration.

Other than these handcrafted local descriptor computation and modeling actions, many deep learning methods have been used for learning action representations. Although the deep learning based methods provide end-to-end learning of descriptors and classification boundaries, they are prone to overfitting, not hierarchical and require a big amount of data. In addition, in spite of the indisputable success of deep CNN methods in image classification, their merit in video recognition is yet to be confirmed. The CNNs are designed to learn features in static images and various methods proposed to adapt them in order to model motion (time) in videos [97, 210, 242, 219, 141, 165]. To model motion in videos, [97] stacked video frames as the input of the convolutional network. Simonyan [210] proposed a two-stream convolutional networks which learns filters based on a stack of N input frames. They model motion with a CNN trained on optical flow information (temporal). Another CNN is trained on still images (appearance). The fusion of the output of the two streams are performed in the last convolutional layer. Fusion in convolutional layer instead of softmax layer helped them to improve the recognition accuracy. In an extension of this approach, [68] replaced the late fusion scheme with an early fusion of the two networks. To model static spatial information, short and long-term motion in videos [252] proposed a hybrid framework to use Recurrent Neural Networks (RNN) along CNNs. The spatial and the short-term information are captured by two CNNs and combined in a regularized feature fusion network for classification. Finally, to model long-term temporal dependencies, a LSTM networks are applied on top of the two features. Ma et al. [141] used CNN together with LSTM for activity and gesture recognition respectively. In [43], after extracting the CNN features from parts of images, they aggregated them using min-max pooling operation before classification.

Although they claim that the aggregation step takes care of the temporal dependency between the frames, the model misses the spatial dependency between different image part. More recently, Carreira et al. [31] used 3D CNNs to define C3D and I3D features aiming for action classification where they achieved high accuracy on benchmark dataset. The problem with these networks is their big number of parameters which makes their training difficult for small-sized dataset. Moreover, they do not consider long term temporal dependency among the action frames.

In overall, the main challenge for space-time approaches is how to describe variation of speed and motion. The joint trajectory methods are usually view invariant and can describe actions in high detail, however, extracting joints in different occlusion and illumination conditions can be challenging. Local descriptor based methods are robust against issues such as noise and illumination conditions, however, they are suitable for modeling local motion of short and simple actions. In general, space-time approaches have achieved acceptable accuracy in recognition of short-term actions. Since their focus is on local motion, description, and modeling trajectory of skeleton joints, they lack the global description of the scene. Hence, they face difficulties handling long-term complex activities such as when the subjects are "sitting" in a "chair" and "Reading" for a long time. Methods such as RNN LSTM are introduced to cope with these challenges, however, modeling temporal dependencies in videos is still an active research topic.

2.2.2 Sequential Approaches

Sequential approaches are another type of single-layer methods that recognize activities by capturing the temporal relationships of the observations (sequence of features). In a given video set, the observations are associated with global or local features. Then, the action recognition in an unseen video is to look for feature patterns of a previously learned sequence of observations. The system deduces that an action occurred when the likelihood between the sequence and the learned action class is high. In the literature [3, 42], sequential approaches are divided into two categories: exemplar-based and state-based approaches. Exemplar-based approaches learn action classes directly from the observations in the training set. They either preserve one representative sequence per action class or multiple instances of training sequences for each action class. During recognition, they match the stored instances with the given sequences. State-based methods learn a generative model that produces a sequence of feature vectors for a given test instance. The recognition takes place by calculating the likelihood of the generated vector. State-based approaches have been used for recognition of short, mid and long-term activities.

2.2.2.1 Exemplar-Based Approaches

In the sequential-based approaches, activities are a sequence of observation and there is no restriction on how to obtain the observations. In exemplar-based methods, actions are represented by a template sequence or a set of sample sequences such as object-based trajectories. Therefore, the main challenge in these methods is how to compare a new sequence with the previous observations.

Originally proposed for speech recognition, Dynamic Time Warping (DTW) has been widely applied in exemplar-based approaches [50, 72, 228, 20]. In [254] atomic activities are considered as a set of measurements in a temporal window. The modeling and recognition of exemplary atomic activity instances are carried out with an algorithm parametrizing their representations using Principal Component Analysis (PCA) and analytical global transformation. Lublinerman et al. [135] proposed a gait analysis approach to discriminate between different activities. First, the system creates an abstraction layer from dynamics of evolution of features captured from image sequences. Fourier descriptors together with vectors of silhouette widths are the two kinds of representation used for describing a frame. Having the representations, each sequence is modeled as a Linear Time Invariant (LTI) system.

Efros et al. [61] proposed a method for recognition of actions when the subjects are in a distance from the camera and their sizes are around 30 pixels. Optical flow fields are used for capturing subtle motions of subjects that are unclear from a far distance. First, 2D optical flows features are extracted. Blurry motion channels computed from optical flows are used for describing motion. Human actions are described as a sequence of the motion descriptors by splitting optical flow vector into horizontal and vertical components. The final feature vector is extracted by obtaining each component's half-wave rectified non-negative channels. Nearest Neighbor (NN) classifier is utilized to match a test sequence action to the training sequences by calculating frame-to-frame similarities.

Lin et al. [130] describes human actions as a sequence of prototypes. The prototypes are constructed based on a shape-motion feature. To assemble the prototype sequences, a hierarchical tree model created by K-means clustering is applied iteratively. Given an unseen video, its generated sequence is matched with the prototype using a FastDTW algorithm with high computational efficiency.

Recently, Lieghtley et al [128] motivated by low-level and invariant models, proposed

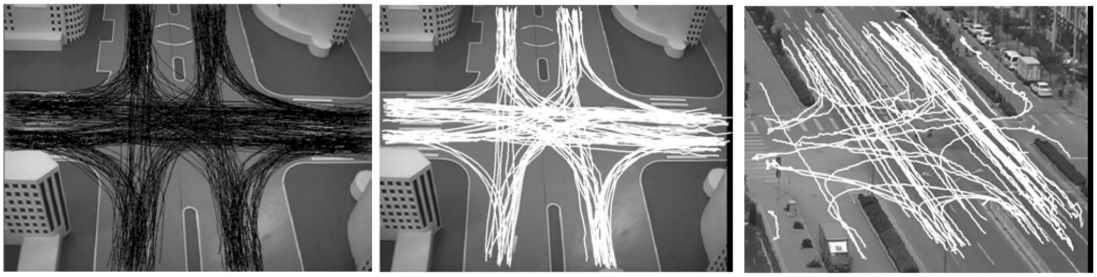


Figure 2.6: Shows the steps of modeling a traffic scene with extracted trajectories proposed by Hu et al. [86]. Left: shows trajectories of a model scene. Middle: shows the result of learned paths in a model scene. Right: is the results of learning traffic model in a real traffic scene.

an exemplar-based method to model an action class with a single sequence. Delegate exemplars are chosen by applying k-means clustering on the poses where their rotations are parameterized using Exponential Maps. They automatically tune the number of exemplars based on the complexity of the given action. Dynamic Time Warping and template matching techniques are applied to find the similarity between an observation and the action models.

More recently [85] addressed human action recognition problem claiming that human actions can be recognized only by looking at a single image if human-object interactions (HOIs) are taken into account. Evaluating the interactions can reveal information about the spatial structure of the scene where the interaction between human and the manipulated object occurs as well as their appearance features. They propose an exemplar-based approach to learn a set of pose-object interaction exemplars. The exemplars are probabilistic density functions modeling spatial interactions of a subject with an object. Rather than a single image, the framework is also extended to recognize actions in videos.

The explained methods have been widely applied on recognition of short-term sequences. For long-term sequences, it is common to reduce the image sequences to object trajectories and learn object behavior by analyzing the trajectory patterns [86, 145, 13, 182]. This analysis is usually done via clustering algorithms to detect the most semantically relevant regions in a given scenario.

For example, in [145] Makris et al. addressed the problem of automatically-learned an activity-based semantic scene model from videos. Their scene model labels each region based on an assigned semantic activity such as enter and exit zones. For the scene models, they use two different representations. First, a topographical representation showing spa-

tial location of scene elements and second, a topological model representing probabilistic nature of the model with a network architecture. They proposed an unsupervised method to learns the scene elements. For learning the trajectory models, a multi-step Expectation Maximization (EM) algorithm is employed. Given a trajectory, comparison and matching procedure with the exemplar paths are carried out. If the given trajectory is similar to the existing exemplars, the likelihood of the existing models are updated, otherwise, a new path is created. The system is evaluated as a supporting tool for tracking moving objects in a surveillance environment. The limitation of this approach is that they only utilize spatial information for clustering of the trajectories and detection of anomalies. Since the temporal information is not taken into account, behavior prediction is not performed.

Hu et al. [86] presented a complete system for automatic learning of motion patterns. The system aims to detect anomalies and predict the behavior of multiple objects in a scene. Their tracking algorithm clusters foreground pixel information using fuzzy K-means ensuring that each cluster is related with a moving object. Later, the trajectories are hierarchically clustered using the extracted spatial and temporal information and each detected pattern is represented with a chain of Gaussian distributions. Statistical pattern analysis is used for detecting the abnormal behaviors in the scene. The abnormal trajectories are the one that deviates from the previously learned exemplars. The system is examined by evaluating data acquired from a traffic scenario. Since abnormal behaviors are less common in real traffic data, they additionally examined the system in an indoor model scene (Figure 2.6). This method is capable of distinguishing between detected events, however, applying such method in an extremely crowded scene is not feasible since the number of events to be modeled is substantial and there is a high variability in abnormal behaviors in such scenarios. Moreover, this method is applied on traffic scenario and are not representative of challenges that surveillance systems should deal with in real-life human activity videos.

2.2.2.2 State-Based Approaches

State-based approaches are another type of sequential methods that learn a state model for activities instead of considering them as sequential observations. In such models, the activities are considered as a set of hidden states of the model where the subject is assumed to be in one of those states at each time instant. Usually, the training is performed using statistical learning of observed sequences and creates generative models which produce sequences with certain probabilities. Generally, a model is learned per activity. For recognition, classification methods based on Maximum a posteriori (MAP) or Maximum Likelihood Estimation (MLE) is employed. Generative approaches such as

Hidden Markov Models (HMM) and Dynamic Bayesian Networks (DBN) and discriminative methods such as Conditional Random Fields (CRF) have been widely applied in state-based recognition tasks [255, 214, 175, 240, 20].

The work of Yamato et al. [255] is the pioneer in this field. HMMs originally applied for speech recognition but they adopted HMMs to recognize activities. First, they convert the input images into feature vectors composed of an array of meshes. The extracted low-level feature vectors are considered as a sequence of observations produced by the activity model. Then, each feature vector is assigned to a symbol which is a word from a codebook created by vector quantization. Next, the time-sequenced feature vectors are converted into symbol sequences. The training phase is optimization of parameters of HMM models to best fit the sequence of symbols describing human action. For recognition, they use Viterbi algorithm [231] by finding the Viterbi path which is the most likely sequence of hidden states for a given action. Based on their achieved reliable performances by HMM models, they encouraged researcher to use these models in their future studies.

Bobick et al. [20] described gestures with 2D trajectories and trained state-based models to recognize them. They converted the trajectories into sequential feature vectors representing state sequences for training. The states are designed to be fuzzy in order to account for variation of speed and motion in gesture performances of the same class. The proposed state-based model is equivalent to a fuzzy version of Markov Models where each transition between the states has its own transition cost. Dynamic programming is used for recognition such that for an input observation the goal of the algorithm is to find a matching model with the best overall transition cost. Yu et al. [262] proposed an action recognition method with extremities that describe human poses with a compact semantic representation named variable star skeleton (VSS). The aim of VSS representation is to accurately capture human extremities by extracting the contour of a human body from input images and encode them with histograms. Each VSS is considered as a state in an HMM model and the recognition is performed by matching the best model. Instead of body contours representation, [105] used textures to describe body motion in the images. The temporal evolution of texture motion histograms is modeled with an HMM. Also, several extensions of HMMs have been tried in action recognition. These extensions by using HMM-based solutions tried to model more complex activities, semantic and temporal relationships among them and to model duration of activities. In [21] an entropy minimization method for HMMs is proposed showing that the video perception problem can be considered as the problem of inferring states in HMMs. Oliver and Pentland [171] described a state-based system working in real-time for modeling human

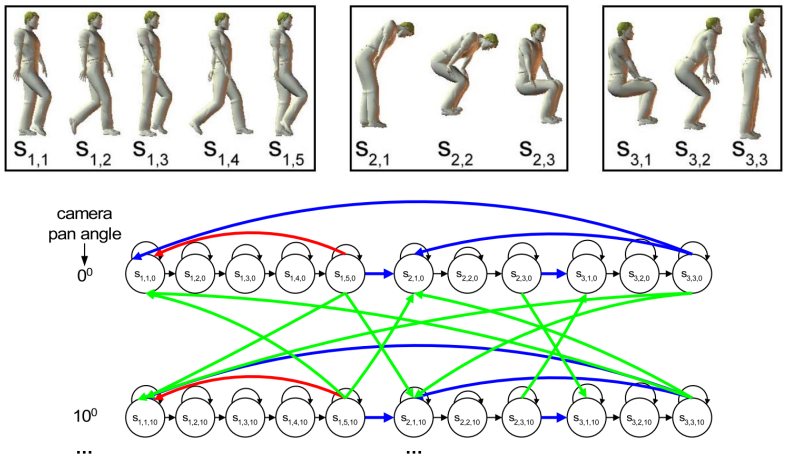


Figure 2.7: Top: The key poses that Automatically extracted. Bottom: A simple Action Net consisting of the three actions [140].

interactions. The model detects and classifies types of interactions. For learning behavior and interaction models, they proposed an architecture that combined bottom-up and top-down methods. For learning the interactive activities they used Coupled HMMs (CHMM) where each HMM is dedicated to infer activities of one subject. The interactions are modeled by coupling the HMMs considering their dependencies. In [157] the authors extend the idea of CHMMs further and developed a Coupled Hidden Semi-Markov Model (CHSMM) that also models the duration of staying in each state. In original HMMs the probability of subject staying in the same state decreases as time passes. In CHSMMs each state has its dedicated duration that fits the model of the described activity. They modeled human interactions more efficiently compared to HMMs and CHMMs. Lv et al. [140] also proposed a semi-supervised architecture similar to CHMMs named Action Net and used 3D human pose for view-invariant action recognition. To overcome the problems of recovering 3D poses from single view, they search from the existing models to find the best action matching the input rather than recovering poses in each frame. The Action Net is a graphical model that learns state transition constraints and is responsible for describing transitions of the automatically recovered poses (Figure 2.7).

Park et al. [175] uses hierarchical Dynamic Bayesian Networks (DBN) to model two-person interactions. DBN is an extension of HMM that consists of multiple hidden nodes that are conditionally independent. At the low-level of the network, the subjects are simultaneously tracked and their body parts are detected. At the higher level, the body poses are estimated using the Bayesian network (BN). Finally, the evolution of the

poses are modeled by DBN.

Recently, Lei et al. [127] proposed a hybrid hierarchical method combining CNNs with HMMs. CNN is used for learning features directly from the input data and the HMM to model action and sub-action dependencies. In the hybrid model, Gaussian Mixture Model (GMM) that has been used for modeling emission distribution of HMM, are fed by high level representations learned CNN from raw data. The Viterbi algorithm is employed for training the models.

Instead of generative methods, some works [214, 240] adopted discriminative approaches to model the activities. [214] proposed an action recognition algorithm based on discriminative Conditional Random Fields (CRF) and Maximum Entropy Markov Models (MEMM). The methods based on HMM usually make conditional independence assumption to simplify the problem which ignores long-term contextual dependencies. The proposed method overcomes these problems using CRFs and convex optimization for training their parameters. Wang et al. [240] proposed a probabilistic framework based on simple silhouette observations. They employed Kernel Principal Component Analysis (KPCA) for extraction of space-time silhouettes features and Factorial Conditional Random Fields (FCRF) for modeling motion information. Modeling long-range temporal dependencies of the sequences is done by information sharing between the nodes of the graphical model. In [207], a discriminative semi-Markov model approach is proposed and in order to simultaneously perform segmentation and recognition. To efficiently resolve this inference problem, a Viterbi based dynamic programming algorithm is proposed.

Nevertheless, deep learning methods improved accuracy of various recognition tasks including action recognition. Other than extracting deep features, these methods attempt to model the temporal evolution of the extracted features. Modeling of temporal progression of actions is usually done using discriminative Recurrent Neural Networks (RNN) and its variations such as Long Short-Term Memory (LSTM). HMMs are simpler models than RNNs, hence, they perform better when working with lower amount of data and problems with less complexity. Also, different from HMMs that make Markovian assumption (that the future state of the process depends only on current state and not sequences of past events), RNNs can find dependency patterns through time. In HMMs there is one-to-one correspondence between input and output of the model (such as in part-of-speech tagging). However, in RNNs the correspondence is not one-to-one and several data points can be mapped in to one and vice versa (such as in translation).

Shahroudy et al. [202] proposed stacked LSTMs (DeepLSTM) and part-aware (P-

LSTM) in order to model long-term temporal correlation among the features of each body part. Instead of using one LSTM cell to learn motion pattern of entire body, the joints are split to different body-part groups representing by different cells. Therefore, each body part is kept independently and the output of P-LSTM is a combination of all part cells. This way, each body part cell has its own input and forget gates and a shared output gate with other body parts. In another variant of LSTM, Liu et al. [132] introduced spatiotemporal LSTM using skeleton information that jointly learn hidden source of spatial and temporal information related to relationship among joints. They propose a trust gate that controls the reliability of the given sequences and handle the inaccuracy of the given 3D joint information. Similarly, in [268] authors propose a distance based geometrical features along with a 3 layered LSTM model to overcome limitations of previous models such as ignoring relationship between non-adjacent geometrical parts.

All in all, the exemplar-based methods are more adjustable since they maintain several instances of the sample sequences hence, they can deal with lack of enough training data. The state-based methods are more capable in terms of inferring probabilities of unseen test instances. However, to achieve an ideal state-based model for describing complex activities, a lot of training data is required. Usually, long-term activities are composed of sub-activities. The single-layer approaches intrinsically ignore modeling of sub-activities. Next, we investigate available methods that model activities hierarchically by putting together atomic sub-activities.

2.3 Hierarchical Approaches

The hierarchical approaches aim at semantically describing high-level activities by recognizing low-level sub-activities or sub-events. For example "Prepare Coffee" activity can be recognized if the sequence of "Entering the coffee region", "Pour water into kettle" and etc. is observed. The sub-activities can be considered as high-level, as long as they cannot be decomposed into semantic ones. The hierarchical approaches have several advantages making them a suitable choice for modeling complex structure of long-term human activities. In addition, hierarchical models intrinsically are more flexible and convenient to incorporate prior knowledge which makes them more comprehensible. Also, they help to better understand the structure of activities. Once the low-level atomic blocks are recognized, they can be used, reused and arranged in various configurations to build different hierarchical models. High-level complex human activities such as daily living activities are easier to describe with hierarchical approaches. The complex structure of single-layer approaches and their features prevent them to be an easy option to create semantically comprehensible and interpretable models. Even if they can model

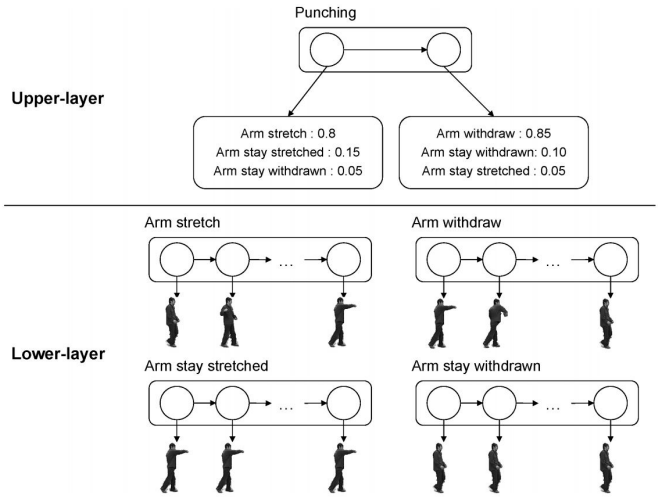


Figure 2.8: Shows a Hierarchical HMM (HHMM) modeling "Punching" action [3]. The lower level recognize atomic sub-actions and the upper level uses them as observations to construct a model based on the probabilities of observations.

a complex activity, they require a big amount of data. A single-layered HMM needs a lot of training instances to learning the state transitions and probabilities of observations knowing that the number of parameters grows as the complexity of the activities increase. The hierarchical models can be trained with less amount of data as they encapsulate redundant sub-activities shared among different activity models. Thereby, they are more efficient in training and recognition phases [3].

In spite of their different structures, hierarchical approaches are strongly connected to the single-layer methods. Prior to constructing a hierarchical model, a single-layer model is required to provide the low-level atomic building blocks. Conversely, single-layer approaches can be further extended into hierarchical methods such as combining several HMM and construct a multi-layer HMM. Most of the proposed methods use single-layer approaches to recognize low-level sub-activities and hierarchical methods to describe higher-level complex activities by linking them hierarchically. We use the taxonomy illustrated in Fig. 2.1 and divide hierarchical approaches into three groups: statistical approaches, syntactic approaches, and description-based approaches.

2.3.1 Statistical Approaches

Statistical approaches usually use state-based models such as HMMs and DBNs to describe activities. Two-layered models are a common hierarchical approach where each layer is a separate state-based model. In the first layer, low-level activities are recognized similar to the single-layer approaches converting feature vectors into atomic semantic

sub-activities. The second layer assumes the recognized sub-activities in the first layer as sequences of observations and creates a model based on their probabilities (Figure 2.8). For recognition, an MLE or MAP classifier can be trained.

To generate such two-layered models, usually Hierarchical HMMs [261, 267, 170] and Dynamic Bayesian Networks (DBNs) [47, 77] are utilized. Yu et al. [261] proposed an action detection method using block-based discrete HMM. The algorithm uses binary blob contour as the feature for each frame which after extraction extended to a star-skeleton representation. The highest point of contour and the center of the blob are considered as two stars and the distance between them is calculated and used for detection of local maxima. To evaluate the obtained time series an HMM is constructed based on predefined classes as state blocks. Each block is dedicated to a subset of basic actions that are trained independently. The recognition is performed by decoding the state sequences through the trained block based HMM. Zhang et al. [267] suggested a method for the semantic description of human interactions. The interactions are modeled as a two-layered structure in a generic hierarchical HMM framework. The first layer's aim is to recognize basic individual activities from low-level audio-visual features (I-HMM). The second layer models the group interactions (G-HMM). G-HMM accepts the output from I-HMM and a set of group features that are directly extracted from the videos. The proposed model is simple to train (Since they are trained on low-dimensional observations), easier to interpret (As each action in the model has a semantic meaning) and easier to extend (each layer can be replaced with various models).

The single-layered HMMs usually have a large parameter space which requires a lot of data for training. Additionally, they need to be retrained when transformed to a new environment. To overcome these difficulties, Oliver et al. [170] proposed a layered probabilistic HMM model for inferring activities in multiple level temporal granularities. The layered representation decomposes the parameter space to different levels and reduces the learning and tuning requirements. The layered HMM (LHMM) representation is robust to variations and can be adapted to a new environment with minimal tuning. LHMM is considered as a cascade of HMMs.

Other than hierarchical HMMs, some methods [47, 77] focused on DBNs as a different type of hierarchical methods. Dai et al. [47] modeled group interactions in a context-aware online framework. Both events and contexts are considered as multi-level, hence, the framework generates interweaved context-event hierarchical models. The event-driven multilevel DBN (EDM-DBN) performs both bottom-up reasoning and top-down context guidance. Gong and Xiang [77] proposed a Dynamic Probabilistic Networks

(DPNs) to model temporal relationships of events to analyze group behaviors. The developed model called Dynamically Multi-Linked HMM (DML-HMM) is constructed based on factorization method resulting in a topology inferring the temporal order of object events. In an extension of this work, [49] used Bayesian Networks to create a hierarchical model of activities based on Reversible Jump Markov Chain Monte Carlo (RJMCMC). The Bayesian Network is used for joint recognition and linking of events which can be extended to multi-layer linking representing compositional event hierarchies.

Depending on the existing scenario, sub-activities can occur concurrently or in a sequential order. For modeling sequential sub-activities, HMM-based methods are adopted. In HMMs, simultaneous activation of multiple state nodes is not possible. In order to describe both concurrent and sequential sub-activities, [208] introduced Propagation Networks (P-net) by allowing activation of multiple states. The activities are presented in partially ordered intervals where each interval is restricted with temporal and logical constraints. The constraint is in terms of duration of sub-activities and their relationships with other sub-activities. Each node in the network is represented by a probability density and is associated with the action intervals. In addition to probabilities, the nodes also accept perceptual information obtained from observations. A particle filter model is adapted to provide real-time analysis of the input sequences.

Rather than using HMMs and DBNs, Yin et al. [260] proposed a Hierarchical Probabilistic Latent (HPL) model consisting multiple layers. In the first level, spatiotemporal perceptual features are extracted. In the second level, the features are utilized to detect atomic action patterns using clustering. Latent Dirichlet Allocation (LDA) is employed in the third abstraction layer to recognize the actions without specifying the number of latent states. This way, they can describe complex human behaviors by clustering low-level features into atomic patterns and finally, to latent topics. However, success of this approach is highly dependent on the correct choice of parameters. Therefore, it requires knowledge from the domain expert or estimation of parameters based on data mining.

By considering the human body as a hierarchical structure, the framework proposed by Han et al. [80] learn a hierarchical manifold space that represents motion patterns of the body. To recognize the represented motion patterns they used Cascade Conditional Random Fields (CCRFs). Similarly [147] is also used a hierarchical representation of motion features encoded via hierarchical K-means trees. Zeng [266] associated domain knowledge in the form of first-order logic, to overcome the problem of insufficient training data. DBN is used for learning the models.

The statistical approaches are capable of successfully recognizing sequential activities in presence of sufficient training data. However, they face difficulties modeling activities with complex temporal structure especially, the ones containing concurrent sub-activities. Edges of HMMs and DBNs models represent the sequential relationship between the states and are unable to describe concurrency of temporal events.

2.3.2 Syntactic Approaches

In the syntactic approaches, activities are considered as a string of symbols. Each symbol in the string represents an atomic-level sub-activities and their integration construct the whole activity. Recognition of atomic sub-activities are required to create the string of symbols (similar to statistical approaches) and can be done with any of the previously explained hierarchical or single-layered approaches. These approaches assume human activities as production rules of a Context-Free Grammar (CFG) that generates a string of atomic actions which are intrinsically hierarchical. The recognition process is performed by parsing of the generated strings. However, string representation is also limited when concurrent activities are described. The models are strict in terms of temporal order of the sub-activities that needs to be sequential.

Various CFG-based methods and their extension have been introduced [94, 100, 149, 151]. Similar to the statistical approaches these methods are generally two-layered where the lower layer detects the atomic sub-activities and the higher layer performs parsing to recognize the activities.

Ivanov and Bobick [94] used a real-time probabilistic syntactic approach for detection and recognition of gestures, activities, and interaction of multiple human subjects in surveillance scenarios. The recognition process is divided into a lower level probabilistic candidate event detection and a higher level stochastic context-free event parsing which is extended to distinguish among uncertain and certain candidate event streams. The framework provides long-range temporal associations and integration of a priori temporal knowledge of events in a given situation. Joo et al. [100] incorporated attribute grammars for recognition of normal events and detection of abnormal incidents. They utilize the power of such grammars to define feature constraints and by that, to describe the syntactic structure of the input strings. Early parser [60] used for event recognition in a parking lot scenario which is extended to handle concurrent events. Strings that do not fit in the syntax of the grammar are considered as abnormal events. Knowing that temporally extended activities can be predicted by a detailed high-level description about the expectation of ordered activities, Minnen et al. [149] introduced a stochastic

grammar for efficient representation of those expectations. They extended stochastic grammars to account for event parameters, sensitivity, and state checks. Moore and Essa [151] proposed a technique to recognize complex multi-tasked activities. Perceptual appearance and motion features, as well as domain-specific contextual information, are used for the description of activities. Atomic sub-actions are represented by unique symbols allowing to construct an interaction with a sequence of strings. Sub-strings are parsed through the Earley-Stolcke algorithm and Stochastic context-free grammar (SCFG) is used for inferring the high-level semantic structure of observations. A new parsing strategy suggested handling the detection and recovery from the errors made.

An important limitation of syntactic approaches is their requirement for production rules. These set of rules should be provided by the user describing all possible events. To avoid this problem, Kitani et al. [110] proposed a method to learn the production rules automatically from the observation in training set. [241] introduced a four-level hierarchical method representing actions by a set of rules. Based on spatiotemporal relations, they divided the rules into three classes of strong, weak and stochastic.

In overall, the syntactic approaches produces activity models that are easy to interpret and can be easily updated and modified. Nevertheless, they can not be applied to concurrent activities and also they require strong supervision where the user needs to provide the rules considering all possible scenarios. In real-world settings none of these limitations are applied, hence, syntactic approaches confront difficulties when utilized in such situations.

2.3.3 Description-Based Approaches

Different from previous hierarchical approaches, description-based approaches are known for their explicit spatiotemporal description of the activities. Therefore, unlike statistical and syntactic methods that only model sequential activities, description-based approaches overcome the problem of modeling concurrent activities and sub-activities. Description-based approaches are intrinsically hierarchical and model activities as a hierarchical structure of sub-activity occurrences. To recognize an activity with such models, the occurrences of sub-activities should meet carefully specified constraints on spatial, temporal and logical relationships which are characteristics of target activities. The constraints allow for a possible combination of different knowledge sources [40, 30]. Such models can be hand-crafted provided by the domain expert [45, 12, 30] or can be learned directly from data (data-driven) or combination of both forms [40]. Allen's temporal interval-based method and context-free grammars are generally applied for description-based activity recognition. In [82, 194] a formal syntax is defined to rep-

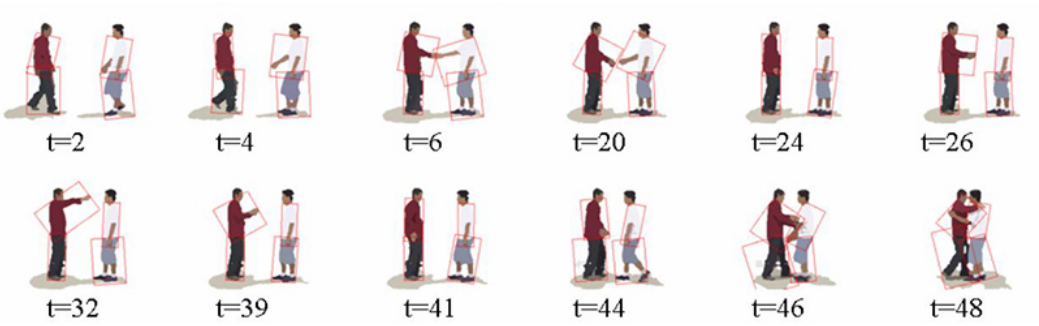


Figure 2.9: shows sequences of processed images converted into body part layers [194].

resent activities. Ryoo and Aggarwal [194] employs a context-free grammar (CFG) for representation of activities and interactions. Dividing the activities into three categories: atomic, composite and interactions, CFG is used for their formal definition. They used pose and gesture retrieved from image sequences to generate action string sequences (Figure 2.9). To match similar temporal features, [184] proposed a method to convert Allen's algebra into a PNF-Network to describe the temporal structure of multimodal information. The obtained values are mapped to a simple network indicated with three states: past, now and future (PNF) allowing fast detection of actions and sub-actions. An occurrence is detected using the imposed constraints by the current status of the sensor and previous state of the network. Motivated by model-based object recognition, Intille and Bobick [93] used Bayesian belief networks to recognize an agent's goal through visual pieces of evidence. For each basic temporal element, they define a visual network and a temporal analysis function validates the temporal relationships of the basic temporal elements such as "before". A large belief network similar to the structure of naive Bayesian classifier automatically generates the action models reflecting the temporal structure of the basic elements. Ghanem et al. [74] developed an interactive system for surveillance that produces descriptions for the queries about a given video. The queries are composed of either predefined queries or directly from the primitive events. Relying on the deterministic and stochastic inference power of Perti Nets, they have been employed for both representation and recognition. The Petri Net representation of users' queries is produced automatically from simpler primitive event nets. The recognition is performed by feeding event tokens through the generated Petri Nets. In [212], for recognition of high-level activities, event logic is used. In [223] symbolic artificial intelligence techniques such as Markov Logic Networks (MLN) is utilized for probabilistic inference of interesting activities. Ijsselmuiden et al. [89] introduced a method based on temporal logic combining various input sources. A framework for analyzing behaviors of basketball players is proposed in [152] using ball trajectory and tracking player's body parts. To generate video descriptions the framework should be provided with semantic

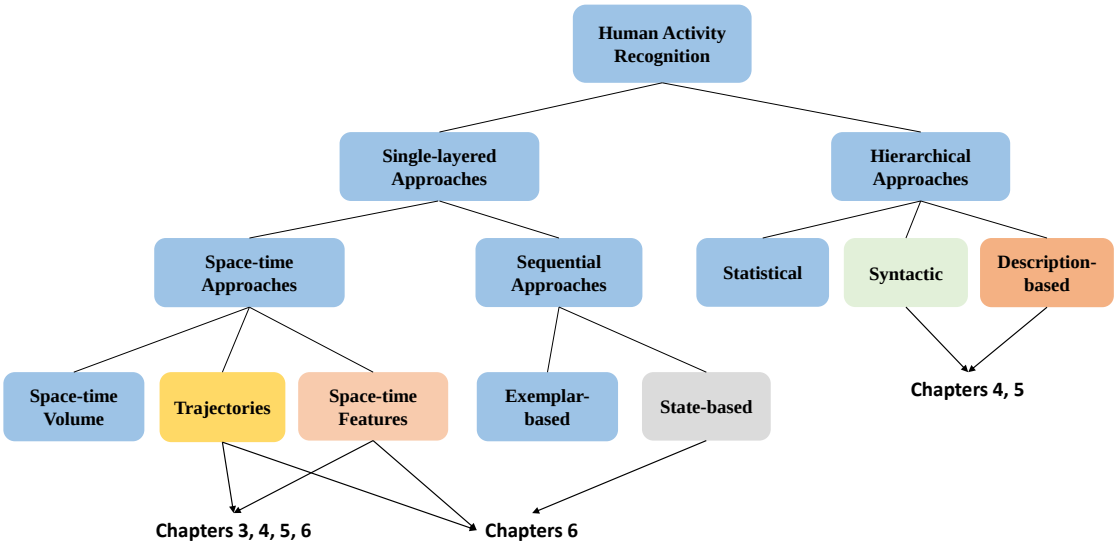


Figure 2.10: Illustrates the activity recognition taxonomy and the related categories to the methods developed in this thesis.

descriptions of various scenarios in the game. Then, it uses first-order probabilistic logic inferences to represent the spatiotemporal knowledge structure. Uncertainty of low-level observations is controlled via MLN network.

It is common to associate description-based solutions with ontology-based formalism in order to define concepts of a specific domain and its relationships [46, 33, 30, 222]. In [222] a description-based surveillance system is developed that uses ontological formalism to manage knowledge and reasoning regarding observations. To describe high-level events, [33] proposed a video analysis system based on Ontological Realism that integrates semantic knowledge in the description process. The system is supervised by human providing the descriptions in the processing loop. Chen et al. [41] designed an ontological framework that uses a data-driven approach to update its model parameters. The deterministic nature of reasoning mechanism in knowledge-based methods forces them to handle the noise in their modules. Some methods [12, 189] handle the noise in observation level, while others [223, 23] adapt to a probabilistic reasoning module to compensate the noisy input at the event level.

In overall, hierarchical approaches are a suitable option for modeling high-level activities that can be decomposed into lower-level sub-activities. They can integrate supervised knowledge easily owing to their composite characteristics. Hence, they can

be trained with efficiency requiring fewer data. Moreover, in presence of noisy data probabilistic hierarchical methods (Statistical and Syntactic) can guarantee a reliable output despite their incapability regarding concurrent activities. Being fully supervised is the main drawback of these methods. The user needs to define every expected situation may the system faced with. Additionally, the deterministic nature of these recognition systems makes updating of the models a challenging task.

Based on the discussions in this chapter, in this thesis, we use a variety of approaches that can be classified in each one of the mentioned categories (figure 2.10). For feature extraction parts of our frameworks we also use different methods. We use geometrical features which are calculated from space-time trajectories of body joints. We extract local space-time features that are calculated from dense trajectories. In addition, we use deep features extracted from CNN networks. To produce the baselines, we have developed a supervised framework 3 that follows a single-layered approach. By taking into account the pros and cons of single-layered and hierarchical methods, in chapter 4 and 5 we introduce a hybrid method that uses single-layered approaches to handle spatiotemporal features and hierarchical method to construct semantical activity models. These methods are capable to work with low amount of data and generate models that are easier to interpret. For gesture recognition 6, in addition to single-layered approaches, we also propose a deep network that uses CNN to extract features and state-based RNN to model the temporal dependencies of the sequences. In next chapter, we start explaining the proposed methods by describing the developed single-layered activity recognition framework.

Chapter 3

Supervised Activity Recognition

"In the end you should only measure and look at the numbers that drive action, meaning that the data tells you what you should do next."

- Alex Peiniger

Contents

3.1 Introduction

In this chapter, we give an introduction to the supervised action and gesture recognition framework which we use throughout the thesis to produce baseline evaluations. This framework belongs to local descriptor based methods which have achieved good performance on different recognition tasks on videos [233, 120]. In the available studies in the literature, these methods have shown robustness toward viewpoint changes, camera movement and scale variations. However, for some of the recognition tasks (*eg.* gesture recognition or data-driven method for activity modeling), we also use deep architecture in our experiments. The supervised part follows bag-of-words (BoW) and Fisher Vector (FV) methods and consists of the following sections:

- Feature detection
- Feature Extraction
- Feature Encoding
- Classification

The pipeline of supervised classification part of the framework is depicted in figure 3.1. Based on this framework, we evaluate two baseline methods for supervised activity recognition. The two methods are: improved Dense Trajectories (iDT) [233] and Trajectory-Pooled Deep-Convolutional Descriptors (TDD) [239]. iDT method belongs to the hand-crafted feature category whereas TDD is a representation learning method using CNN feature maps. However, using two methods from both categories we have got the chance to compare our method with those popular methods. It should also be noticed that in some of the tests we also use the features from other prevalent categories such as geometrical features (*eg.* features based on angles and distances of the different body parts). Moreover, we evaluate two different variants of the framework for feature encoding and recognition. Both variations use the same mechanism for feature detection and classification. The only difference is in feature encoding part. First one is based on conventional bag-of-words method while the second variation uses improved Fisher vector in the encoding step. We evaluate these methods on one public and one private daily living activities datasets: GAADRD [102] and CHU[1]. In the following chapters, we follow a gradual transition from supervised learning toward unsupervised modeling of the activities. For the unsupervised part, we use a hierarchical representation scheme in the framework in order to model the activities.

[1] https://team.inria.fr/stars/demcare-chu-dataset/

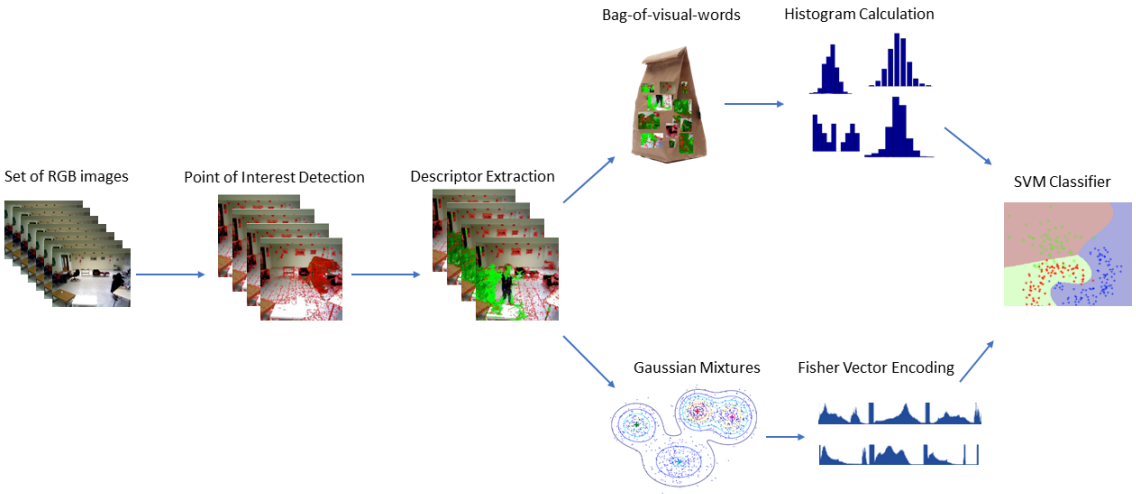


Figure 3.1: Supervised action recognition: (Top row) features are detected and extracted from the input RGB images and the bag-of-words dictionary is trained using these features. Afterwards, the histograms of the extracted features are calculated and used for classification. Rather than the bag-of-words model, the feature distributions are calculated and Fisher vector encoding is performed prior to the classification step (the down row).

Next, we describe building blocks of our supervised activity recognition framework: in section 4.3, the process of feature extraction is described. We explain how the features are detected and then, how different types of features are extracted from input images. In section 3.3, we describe the bag-of-words model, and then in section 3.4 we explain the Fisher Vector model in more details. In section 3.5 the classification method is discussed. In section 3.6, we describe evaluation metrics details and finally in section 3.7, we provide evaluations results. We conclude this chapter in section 3.8.

3.2 Features Extraction

In this section, we describe how the supervised framework detects the interest-points and local features of the input images, the bag-of-words process that encodes the extracted features and the classification task. As mentioned (see figure 3.1 top row) these tasks are carried out in four steps: detection of local features, descriptor extraction, feature encoding and SVM classification.

3.2.1 Local Feature Detection

In order to find salient local points of interest in video and image categorization tasks, different methods based on global and local features have been employed [119, 101]. In this work, we use improved dense trajectories [233] which densely samples point of interests and tracks them in consecutive frames of a video sequence.

In dense trajectories method, the points of interests are sampled using a W pixels sized grid in multiple scales. Each trajectory is track separately at each scale for L frames. We use a sampling size of $W = 5$ which based on experiments in the original paper [233] gives good results. To avoid drifting in the tracking of interest-points, the length of the trajectories are set to be limited and the trajectories exceeding this limit are removed from the process. This limit in the original work was experimentally set to $L = 15$, however, we found a specific value for GAADRD dataset ($L = 5$) in our experiments. It is impossible to track interest-points in homogeneous areas of the images. For such regions, after an interest-point is sampled, eigenvalues of their auto-correlation matrix is calculated and if it is below a threshold [206], the interest-point is excluded from tracking. Since these kinds of descriptors are designed for action recognition and action recognition is mostly interested in dynamic motion information, the static trajectories and trajectories with abrupt movement and large displacements are pruned in the preprocessing step.

Each interest-point at frame t, $P_t = (x_t, y_t)$ is tracked to the next frame in the sequence $(t + 1)$. Optical flow field w_t is calculated using frame t and $t + 1$ and is used to smooth the trajectory via a median filter on it:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega_t)|_{(\bar{x}_t, \bar{y}_t)} \tag{3.1}$$

where M is the median kernel and $(\bar{x}_t, \bar{y}_t)$ is the rounded coordinate of (x_t, y_t).

3.2.2 Extraction of Descriptors

Once the trajectories are extracted, the descriptors in the local neighbourhood of the interest-points are computed. There are three different types of descriptors extracted from the interest-points: Trajectory shape, motion and appearance based descriptors. Next we explain how each one of them is extracted.

3.2.2.1 Trajectory Descriptors

Local motion pattern of a scene region can be encoded by shape of the trajectories. Given a trajectory of length L, its shape can be described by a sequence $(S = (\Delta P_t, \ldots, \Delta P_{t+L-1}))$

of displacement vectors: $\Delta P = (P_{t+1} - P_t)$. If the resulted displacement vector gets normalized by its magnitude, the final displacement vector (trajectory shape descriptor a.k.a TSD) will be obtained:

$$S' = \frac{\Delta(P_{t+1} - P_t)}{\sum_{j=t}^{t+L-1} ||\Delta P_j||} \tag{3.2}$$

Other than spatial scales, the trajectories are also calculated in multiple temporal scales in order to represent actions done with speed. However, this doesn't help improve the recognition results as claimed in the original paper and is recommended to keep the length of the trajectories fixed. In the original work the value of L is set to 15, therefore the length of the TSD descriptor is 30. In our experiments, we use $L = 5$ for the GAADRD dataset due to its low frame rate, to capture the local motion of the subjects in videos.

3.2.2.2 Motion Descriptors

We also compute two motion descriptors (*i.e.* HOF and MBH) along the extracted trajectories. Like other popular descriptor computation methods [54, 111], this is performed by calculating these descriptors in a volume around the detected interest-points and throughout their trajectories (spatiotemporal volume). Size of the constructed volume is $N \times N$ pixels around the interest-point and L frames long. For all of the grids in the spatiotemporal volume, the descriptors are calculated and concatenated to represent the final descriptor. In our experiments, we set $N = 32$ and parameter L to 5 or 15 depending on the dataset. HOF descriptor [48] is based on optical flow and captures local motion information. Thereby, the orientation of the optical flow is calculated in each volume, normalized (L_2 norm) and quantized into 8 bins with an additional 0 at the end (final size of bin is 9). Since the spatiotemporal volume is subdivided into 4 spatial and 3 temporal grids, the final size of the computed descriptor is 108 ($4 \times 3 \times 9$).

For MBH descriptors, derivatives of optical flow vector are separately calculated for the horizontal and vertical axis. This descriptor encodes relative pixel motion between consecutive frames. Similar to HOF the computed derivations in each axis is quantized into 8 bins and L_2 normalized. Consequently, there is a vector of size 96 ($4 \times 3 \times 8$) for each component (x and y components). Finally, by concatenating MBHx and MBHY descriptors, the final MBH vector's dimension is 192. In the calculation of MBH descriptor, constant motion in the optical flow field is ignored and only information about the change of this filed is considered. This helpes a lot in reducing the noise pertinent to the constant background motion and achieves better results in various experiments compared to HOF descriptor [233].

3.2.2.3 Appearance Descriptors

Unlike motion-based descriptors that focus on the representation of the local motion, appearance descriptor's focus is on representing static appearance information. Having shown good results on a variety of datasets [237], HOG descriptor is selected as appearance descriptor in our experiments. Similar to MBHX and MBHY descriptors, the final size of concatenated and normalized HOG descriptor is 96.

3.2.3 Geometrical Descriptors

These descriptors represent the spatial configuration of the skeleton joint information and therefore model human body pose in each frame. They use calculated angles between the skeletal vectors or a computed distance between the joints using different metrics. In order to formulate a pose descriptor, similar to [257], first, pairwise joint distances and angles at each frame are calculated and then, to augment the characteristics of the final descriptor spatial and temporal relations between consecutive poses are described (similar to [216] and [129]).

For a video, the feature extraction algorithm accepts a sequence of high dimensional vectors of skeleton joints for each action with T frames and J joints for each skeleton:

$$S = \{P_t \mid \forall\, t \in (1, \cdots, T)\} \tag{3.3}$$

where $P_t = \left\{p_t^i \;\middle|\; \forall\, i \in (1, \cdots, J)\right\}$ is the set of skeleton joints at t^{th} frame and $p_t^i = (x_t^i, y_t^i, z_t^i)$ is the i^{th} joint of the skeleton p^{th} in t^{th} frame.

We represent human pose as a tree structure where the chin node is considered as the root node. The joint coordinates are transformed according to the root coordinate in order to eliminate the influence of joint positions with respect to the sensor coordinates. Before representation, to reduce jitter in estimated joints trajectories we smooth joints position over temporal dimension by applying polynomial regression using weighted linear least squares and second-degree polynomial model. Each subject performs similar gestures with variable speed resulting in variable frame sizes and joint trajectories. To achieve uniform performance speed along the temporal dimension and to remove outliers in joints trajectories, once the smoothed joint positions are obtained, cubic interpolation of the values at neighboring joints is applied in the respective dimensions. Furthermore, to remove abrupt movements of the hand and elbow joints that are neither part of the gesture nor a jitter, a threshold is set which results in more stable joint values.

To compensate variations in body size, shape and proportions, we follow the method

in [264]. Starting from the root node (chin), we iteratively normalize body segments between the joints to average bone size in the training data.

To represent the skeleton, both joints' Euclidean distances and angles in polar coordinate are calculated using normalized joint positions. In order to preserve temporal information in pose representation, a feature extraction scheme based on temporal sliding window is adopted. At each time instance, Euclidean distances between all the joints are calculated. Besides, for each joint, distances from other instances' joints included in the sliding window is calculated and stored as well. If J_i^t represents features of joint i at time t and w shows the sliding window size: $J_i^t = [x_i^t, y_i^t]$ defines raw skeleton features at time t, where $i = 1, ..., 8$. Then, F^d calculates the distance descriptor:

$$F^d = \sqrt{(x_i^t - x_j^{t'})^2 + (y_i^t - y_j^{t'})^2} \tag{3.4}$$

Similarly, to calculate angular feature in polar coordinate we use:

$$F^a = \arctan\left(x_i^t - x_j^{t'}, y_i^t - y_j^{t'}\right) \tag{3.5}$$

where $t' \in \{t, t-1, ..., t-w\}, t' > 0$ and $i, j = 1, 2, ..., 8$ for both Eqs. 3.4 and 3.5.

Combining these features together, produces the final descriptor vector $F = [F^d, F^a]$ of dimension $N_f = 2 * w * {N_j}^2 = 1280$. To eliminate redundant information, PCA is applied on the position of torso joints and 512 dominant values preserving 99% of the descriptor information are kept. The final vector is normalized to zero mean and unit variance.

3.2.4 Deep Features

In this section, we give a brief explanation of the deep feature we use in our experiments called: Trajectory-Pooled Deep-Convolutional Descriptors (TDD) [239]. These video features try to combine both benefits from hand-crafted and deep learned features. To achieve this, multi-scale convolutional feature maps pool deep features around the interest-points of the detected trajectories by improved trajectory method. The process of extracting these types of deep features are very similar to the one in hand-crafted we explain in the previous section (3.2.1). The main difference here is that rather than computing the hand-crafted features around the spatiotemporal volume of the trajectories, deep features are extracted using convolutional neural network (CNN) maps. Specifically, a trained two-stream ConvNet on a large dataset is used as a generic feature extractor in multiple scales to extract features from RGB frames. Meanwhile, trajectories by improved dense trajectory method are computed. Afterwards, the ConvNet responses over the spatiotem-

poral tubes located at the trajectories are pooled. Finally, Fisher vector representation is used to aggregate the local features into a global super vector for the whole video. The two-stream ConvNet architecture proposed by Simonyan [210] is adopted for TDD feature extraction. The two-stream CNN consists of two separate CNN: spatial and temporal networks. The motion features are trained on optical flow and extracted using the conv3 and conv4 layer of CNN. Additionally, for the training of the appearance features on RGB frames, the conv4 and conv5 layer of CNN is used. In the next subsections, we explain how these two streams are trained and performed feature extraction for our experiments.

3.2.4.1 Spatial Stream

After the CNN is trained (we use VGG-16 net [211] pre-trained on ImageNet [53]), it is applied frame-by-frame and volume-by-volume to extract generic features. Spatial features are designed for capturing static appearance. The expected size of the input images for this net is $(224 \times 224 \times 3)$. In total there are 16 layers in the VGG-16 network. 13 layers of them are convolutional layers and the remaining 3 are fully connected layers. The network uses $3by3$ filter sizes which helps it to learn a deeper neural net. The sampled frames in the training phase are cropped before going through the net. The cropping is performed by taking a random patch of size 224×224 from the center or any other corner of the sample. A random flip operation can be applied to the selected patches.

3.2.4.2 Temporal Stream

The purpose of the temporal network is to describe the dynamic motion information. In order to do that, this method utilizes optical flow fields to capture the changes (displacements) between the consecutive frames. The input for this network is the volume of stacking optical flow fields. Similar to the spatial network, this net also uses the VGG-16 architecture with 16 layers. The training process is also similar: sampling M frames from the video, computing the optical flow fields and random crop/flip of the selected patches.

It should be noticed that the action recognition framework is designed in a way that it takes advantage of any number of features whenever they are available. In case that one feature is absent (for example when a dataset does not have skeleton), it becomes excluded from the computed feature vector. The framework is flexible and designed to be adaptive to such occasions.

3.3 Bag-of-visual-features Encoding

Inspired from bag-of-words frequency-based models (frequencies of words in a dictionary) designed for order-less document representation in natural language processing (NLP) [197] and texture recognition (where characteristics and repetition of textons matters more than their spatial arrangement) [124, 153], bag-of-visual-features model involves with the representation of a training set (words of a dictionary) and training of a classifier over those representations. For encoding of a probe image, the trained dictionary can be used and the classifier can assign a label to the given encoded image or a set of encoded images (Figure 3.1 top row).

After interest-point detection and descriptor extraction steps described in the previous sections, the feature encoding (vector quantization) starts with the learning of a visual vocabulary. This process is done with clustering of the feature word vectors. There are many clustering methods that can be used for this purpose. The nearest-neighbor algorithm is used for clustering of features and also for similarity ranking and classification. However, K-means algorithm is the common algorithm that has been used for this task. The goal of this algorithm is to minimize the distance between points $(x_1, x_2, \ldots, x_n)$ in a set of clusters $S = \{S_1, S_2, \ldots, S_k\}$ and their cluster center μ_k:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} ||(x - \mu_i)||^2 \tag{3.6}$$

where μ_i is the mean of observation points in cluster S_i. This algorithm starts with a random initialization of K clusters. Then each one of features is assigned to the nearest cluster center. Afterwards, the new cluster centers are obtained by calculating the mean of all features assigned to the same cluster. This process is repeated until convergence. The final arrangement of the clusters defines the visual vocabulary. Each cluster center obtained by the algorithm is a code vector which is used in the encoding process of the features.

No matter how the clustering is done, there will be some features that lie in the cluster boundaries and have an equal distance to more than one cluster centroid. Centroid initialization affects the outcome of the clustering and the obtained vocabulary. For relatively small vocabularies this problem can be avoided with multiple iterations of K-means in a validation process and select the best vocabulary. However, this becomes impractical for larger datasets. Since the datasets in our experiments are not very large, we use iteration approach to obtain the near optimal vocabulary. Moreover, to

calculate the distance between feature points in the clustering process and assignments in the encoding stage, the common choices are Manhattan (L_1), Euclidean (L_2), and Mahalanobis distances.

In terms of calculating the distances in the vector space for similarity ranking during classification methods such as Euclidean, Manhattan and Quadratic distances are the popular ones. We use Euclidean distance to measure similarity. If h_1 and h_2 are two calculated histograms of the input frames' feature vectors, the Euclidean distance L_1 between the two is calculated as below:

$$D(h_1, h_2) = \sum_{i=1}^{N} |h_1(i) - h_2(i)| \tag{3.7}$$

where N is the number of bins in the calculated histogram. However, during the distance calculation, some words show relatively higher importance to the others. There are different methods that weight words during distance computation which Term Frequency-Inverse Document Frequency (TF-IDF) [196] is one of the most popular schemes.

In our experiments we use K-means algorithm for clustering and Euclidean distance for distance measurement in cluster centroid computation, since they result in high classification accuracy with our data. We use the same metric for word term assignment using the visual vocabulary. To find the optimal vocabulary with proper cluster centroids the iteration number is set to 200.

3.4 Fisher Vector Encoding

The calculated descriptors are employed to create action representations based on Fisher vectors [180, 181] (3.1 down row). Accordingly, first and second order statistics of a distribution of the feature set $\mathbb{X}$ are used for encoding a video sequence. Generative Fisher vector model is formed to model the features and the gradient of their likelihood are computed according to the model parameters (λ), *i.e.* $\Delta_\lambda \log p(\mathbb{X}|\lambda)$. The way the set of features deviates from their average distribution is depicted through a parametric generative model.
To improve the learned distribution to further fit the observed data, a soft visual vocabulary is obtained by fitting a M-centroid Gaussian Mixture Model (GMM) into the training

features within the preliminary learning stage:

$$p(x_i|\lambda) = \sum_{j=1}^{M} w_j g(x_i|\mu_j, \Sigma_j), \tag{3.8}$$

$$\mathbf{s.t.} \quad \forall_j : w_j \geq 0, \quad \sum_{j=1}^{M} w_j = 1, \tag{3.9}$$

$$g(x_i|\mu_j, \Sigma_j) = \frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}} e^{-\frac{1}{2}(x_i-\mu_j)^T \Sigma_j^{-1}(x_i-\mu_j)}, \tag{3.10}$$

where $x_i \in \mathbb{X}$ represents a D-dimensional feature vector, $\{g(x_i|\mu_j, \Sigma_j)\}_{j=1}^{M}$ are the component of Gaussian densities and $\lambda = \{w_j, \mu_j, \Sigma_j\}_{j=1}^{M}$ are the parameters of the model: Respectively, $w_j \in \mathbb{R}_+$ is the mixture weights, $\mu_j \in \mathbb{R}^D$ is the mean vector, and $\Sigma_j \in \mathbb{R}^{D\times D}$ is the positive definite covariance matrices of each Gaussian component. The parameters λ are found using the Expectation Maximization restricting the covariance of the distribution to be diagonal.

The GMM parameters are assessed through random sampling of a subset of $100,000$ features from the training set where the number of Gaussians is considered to be $M = 128$. Initialization of the GMM is performed ten times to obtain high precision and accordingly to provide the lowest error pertinent to the codebook. We define the soft assignment of descriptor x_i to the Gaussian j as a posteriori probability $\gamma(j|x_i, \lambda)$ for component j:

$$\gamma(j|x_i, \lambda) = \frac{w_j g(x_i|\mu_j, \Sigma_j)}{\sum_{l=1}^{M} w_l g(x_i|\mu_l, \Sigma_l)}, \tag{3.11}$$

Thereafter, the gradients of the j-th component can be calculated with respect to μ and σ using the following derivations:

$$\begin{aligned} G_{\mu,j}^{\mathbb{X}} &= \frac{1}{N_x\sqrt{w_j}} \sum_{l=1}^{N_x} \gamma(j|x_l, \lambda) \left(\frac{x_l - \mu_j}{\sigma_j}\right), \\ G_{\sigma,j}^{\mathbb{X}} &= \frac{1}{N_x\sqrt{2w_j}} \sum_{l=1}^{N_x} \gamma(j|x_l, \lambda) \left(\frac{(x_l - \mu_j)^2}{\sigma_j^2} - 1\right), \end{aligned} \tag{3.12}$$

where N_x is the cardinality of the set $\mathbb{X}$. Finally, a set of local descriptors $\mathbb{X}$ as a concatenation of partial derivatives is encoded as a function of the mean $G_{\mu,j}^{\mathbb{X}}$ and standard deviation $G_{\sigma,j}^{\mathbb{X}}$ parameters for all M components:

$$V = [G_{\mu,1}^{\mathbb{X}}, G_{\sigma,1}^{\mathbb{X}}, ..., G_{\mu,M}^{\mathbb{X}}, G_{\sigma,M}^{\mathbb{X}}]^T. \tag{3.13}$$

As a final step, we apply the power normalization and L2-normalization. The dimension of the Fisher vector representation is $2DM$. Fisher descriptors have shown to outperform bag-of-words approach for image classification and retrieval.

3.5 Classification

3.5.1 SVM

In this section we introduce the Support Vector Machines (SVM) [226] and its formulation which we use in our experiment to assign categorical labels for the encoded feature vector of the videos. SVM is a supervised classifier which means it requires ground-truth labels at the training step. Among the other classification tasks, this classifier became popular in computer vision and specially in object and action recognition tasks [180, 181, 233]. The main objective of the SVM classifier is to find a hyper-plane that can separate two classes with a maxim margin. This prevents overfitting and promote the generalization power of the classifier.

Let x be a vector representing feature vector of a video. The goal is to have a classifier that is capable of associating a label to every vector x based on the desired criterion. The classification takes place by checking the sign of a linear scoring function. The learning aims to estimate parameter w in such a way that the result of the scoring function is positive if the vector belongs to the correct class. A negative score means the vector does not belong to the class in question. If a set of n examples is given: (x_i, y_i) where $i = 1, 2, \ldots, n$ and y_i is the ground-truth, the training process is fitting a scoring function to the given set. Exact fitting of a function to a training data reduces generalization capability of the learned function. To avoid overfitting, a regularization parameter (C) which balance fitting accuracy with the regularity of the trained scoring function is used. This yields to a regularized loss function. The trained function will be obtained by solving the following optimization problem:

$$\arg\min_{w} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} l_i \tag{3.14}$$

where l is the loss and C scales the loss cost in the training set. This objective function is quadratic and convex, therefore a global optimum exists for this function. There are different methods to minimize the objective function. In the implementation we used for our experiments Stochastic Gradient Descent method is implemented which is suitable for the linear classifier.

SVM inherently is a binary classifier. The most common way to do multi-class classification with SVM (which is required for action classification in our experiments) is to use C one-versus-all binary classifiers where C is the number of classes. The class with the greatest margin is chosen as the predicted label. Another strategy is to train a set of one-versus-one classifiers and at the end to choose a class which most of the classifiers voted for it. While this strategy decreases the training time, it does not outperform the classifiers trained with one-versus-all strategy [176].

In our experiments we use SVM implementations from both LibSVM [37] and Scikit-Learn [176] libraries for classification.

3.6 Evaluation Metric

3.6.1 Precision and Recall

The performance of tested baselines and proposed methods are evaluated with Precision, Recall, and F_1 score metrics. Precision metric is a measure that indicates the relevancy of the results. However, the Recall signifies how many returned instances are truly relevant. F_1 score metric is harmonic mean of the precision and recall metrics. The metrics are defined as follows:

The Precision metric is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p):

$$P = \frac{T_p}{T_p + F_p} \tag{3.15}$$

Recall metric is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n).

$$R = \frac{T_p}{T_p + F_n} \tag{3.16}$$

the (F_1) score is defined as:

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} \tag{3.17}$$

In the evaluation of the obtained results, it should be noticed that when the system

achieves high recall rate but it has a low precision means that it return many results which many of them are incorrect predictions of the target labels. On the other hand, when the system has a high precision but low recall rates, it is less sensitive but most of its detected results are correct. An ideal framework has a high rate of both precision and recall metrics.

3.6.2 Data Split

To split the datasets, we follow the split suggested by the dataset providers. In the datasets for daily living activities, the evaluated datasets are divided into training and testing sets where three-fifths of the dataset is used for training and the rest for testing. We use the same protocol in order to be compatible with the literature and be capable of comparing our framework with previously suggested methods.

3.7 Experiments

In this section, we present our evaluations of the baseline methods on two datasets. As explained in the previous section, there are two types of evaluation methods: the first one is based on feature extraction and encoding by Bag-of-Words model. In this method, first, 3 types of descriptors are extracted: dense trajectories (HOG, HOF, MBHX, MBHY), geometrical features (Distance and angle) and deep features (trajectory pooled spatial and temporal deep features). After, the extracted features are encoded with BoW model with 4 different codebook sizes: 250, 500, 1000, 2000. The process is finished with a classification step where we use SVM classifier for this purpose.

In the second method, we use a different encoding scheme. Rather than BoW model we use Fisher vector encoding. The extracted features stay the same as the previous method (8 feature types). For training of the Fisher vector codebooks, we use 6 different codebook sizes: 16, 32, 64, 128, 256, and 512.

3.7.1 GAADRD Dataset

The GAADRD [102] action dataset consists of 25 people with dementia and mild cognitive impairment that perform ADLs in an environment similar to a nursing home. The GAADRD dataset is public and was recorded under EU FP7 Dem@Care Project[2] in a clinic in Thessaloniki, Greece. The camera monitors a whole room where a person performs directed ADLs. The observed ADLs include: Answer the Phone (AP), Establish Account

[2]http://www.demcare.eu/results/datasets

Balance (EAB), Prepare Drink (PD), Prepare Drug Box (PDB), Water Plant (WP), Read Article (RA), Turn On Radio (TOR). A sample of images for each activity is presented in Figure 3.3. Each person is recorded using a RGBD camera of 640×480 pixels of resolution. Each video lasts approximately 10-15 minutes. We have randomly selected 2/3 of the videos for training and the remaining ones for testing.

Table 3.1 shows the obtained results by the BoW framework. Based on the obtained results we can conclude that:

- Medium size codebook works best for this dataset. This might be because of the medium size of the dataset. Accuracy is lower with small codebook size (250). As codebook size grows, the accuracy grows as well until the point that a drop in accuracy is observed. This drop in accuracy coincided with bigger codebook size (2000) might be due to an overfitting situation.

- Deep TDD features obtain better performance among others. However, improved dense trajectories achieve competitive performance. Especially, HOG feature achieves same level of accuracy when the optimized codebook size is set (1000 words).

- Geometrical features achieve the worst results. Angle feature achieves better accuracy since angular posture features are more informative in daily activities (some activities such as "Watering Plant" is performed while the subject has a bending posture). Distance feature performs the worst since the distance features in daily activities is not a key feature. Most of the activities are performed in a way that the relative distances between the joints stay the same during the activity (*eg.* in "Prepare Drug Box" and "Prepare Drink").

Table 3.2 shows the obtained results by the Fisher Vector framework. Based on the reported results we can draw the following conclusions:

- There is no constant trend of achieving better performance by applying bigger size of the codebook in all of the feature types. However, usually bigger codebook size results in a better performance, especially with deep features.

- Deep TDD features along with HOG feature achieve the best results. This high performance might be because of capturing contextual information by the appearance features.

- As it is expected, the geometrical features perform poorly. The distance feature performance is the worst even when it is compared with the bag-of-words model.

- Fisher vector encoding achieves much better results than the bag-of-words model owing to better feature encoding power. This performance boost is obtained because in the Fisher Vector encoding, rather than only keeping visual word occurrences, statistics such as the difference between extracted features and dictionary elements are also stored as part of the representation.

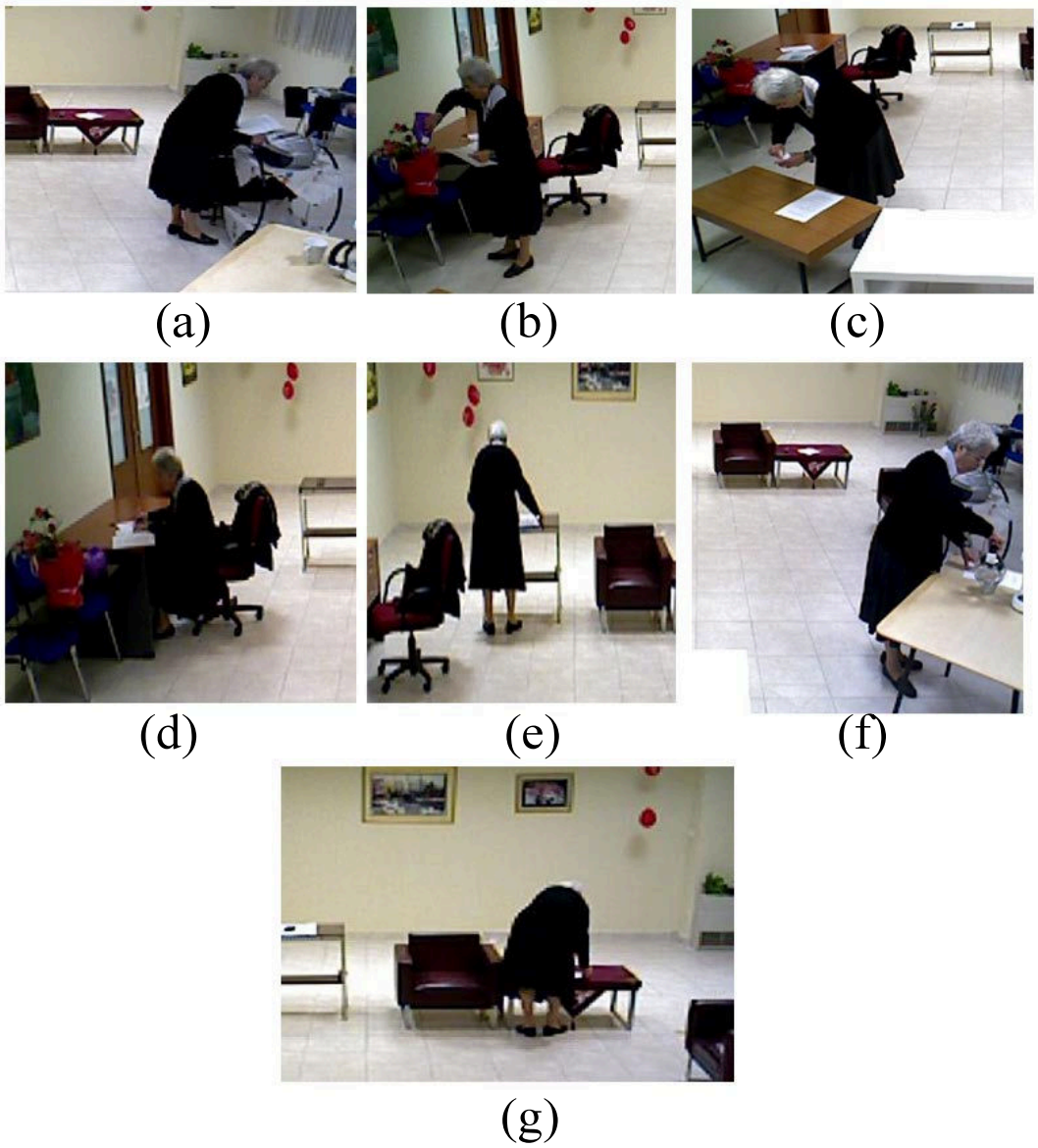
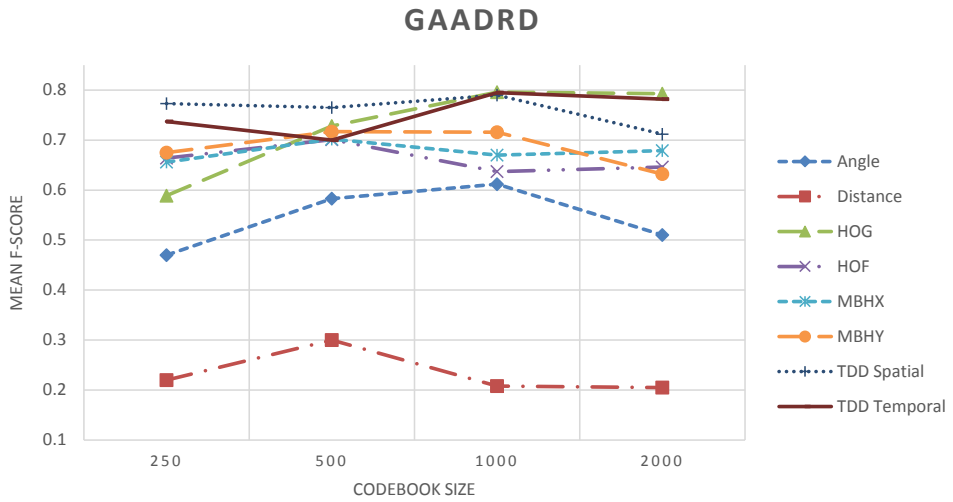


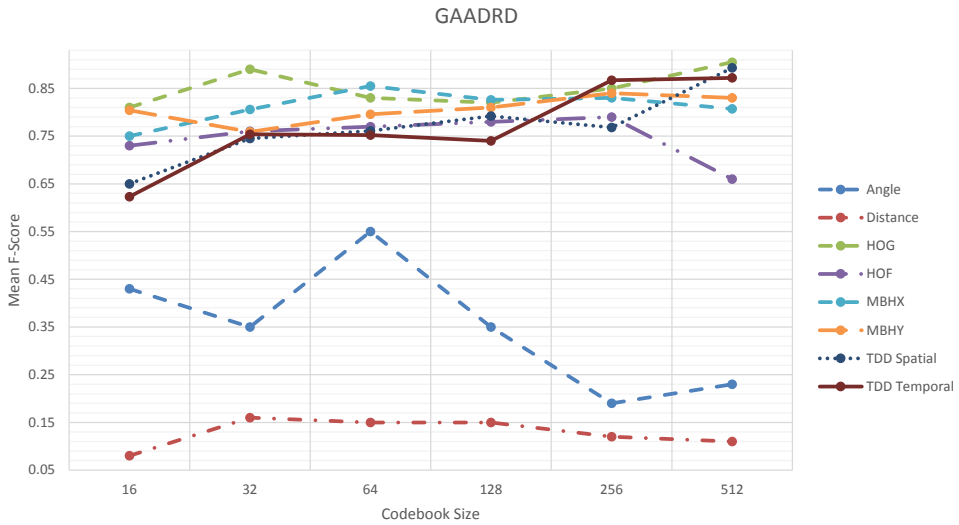
Figure 3.2: A sample of activities in datasets: (a) Turn On Radio, (b) Water Plant, (c) Prepare Drug Box, (d) Read Article, (e) Establish Account Balance, (f) Prepare Drink, (g) Answer the Phone.

Table 3.1: Results of using different feature types applying bag-of-words method on GAADRD dataset. The plot shows F-Score values w.r.t. codebook size.



	250			500			1000			2000		
	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score
Angle	48.8	48.2	0.47	60.4	58.4	0.58	61.5	62.7	0.61	62.3	48.4	0.51
Distance	23.5	22.8	0.22	24.6	25.1	0.24	22.4	22.5	0.20	22.2	22.7	0.20
HOG	66.5	62.4	0.58	80.5	72.0	0.72	83.4	79.4	0.79	83.5	79.1	0.79
HOF	68.8	67.0	0.66	75.7	70.1	0.70	68.5	65.1	0.63	69.0	65.8	0.64
MBHX	71.1	65.8	0.65	75.4	70.4	0.70	73.0	67.1	0.67	71.8	68.8	0.67
MBHY	71.5	66.5	0.67	78.0	71.8	0.71	77.8	72.0	0.72	69.8	65.1	0.63
TDD Spatial	84.1	77.1	0.77	83.7	76.2	0.76	78.8	80.8	0.79	74.8	72.1	0.71
TDD Temporal	75.8	75.2	0.73	73.4	71.1	0.70	86.1	79.0	0.79	79.7	80.0	0.78

Table 3.2: Results of using different feature types by applying Fisher Vector method on GAADRD dataset. The plot shows F-Score values w.r.t. codebook size.



	16			32			64			128			256			512		
	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score
Angle	61.2	42.4	0.43	47.7	34.4	0.35	63.5	53.5	0.55	42.5	35.0	0.35	41.4	20.4	0.19	33.5	27.2	0.23
Distance	10.7	9.1	0.08	18.2	18.7	0.16	17.7	18.2	0.16	19.8	17.8	0.15	13.8	14.4	0.12	12.5	12.5	0.11
HOG	83.7	80.5	0.81	89.8	90.8	0.89	82.1	85.5	0.83	86.0	80.2	0.82	84.4	88.0	0.85	88.5	93.4	0.90
HOF	72.0	74.5	0.73	75.5	78.8	0.76	76.6	80.5	0.77	75.5	81.8	0.78	79.1	79.8	0.79	73.4	68.2	0.66
MBHX	75.5	75.8	0.75	81.1	81.1	0.80	86.0	86.0	0.85	83.2	82.8	0.82	85.0	84.4	0.83	81.7	80.8	0.80
MBHY	80.7	80.5	0.80	77.2	76.0	0.75	80.8	79.5	0.79	82.0	81.0	0.81	85.7	84.8	0.84	84.4	83.2	0.83
TDD Spatial	70.7	61.0	0.65	80.5	74.8	0.74	80.1	76.1	0.76	81.8	79.7	0.79	80.8	76.5	0.76	90.2	89.3	0.89
TDD Temporal	65.1	61.0	0.62	80.8	74.5	0.75	79.8	75.2	0.75	75.6	75.1	0.74	88.1	86.4	0.86	88.0	87.1	0.87

3.7.2 CHU Nice Dataset

This dataset is recorded in the Centre Hospitalier Universitaire de Nice (CHU) in Nice, France. It contains videos recorded from patients performing everyday activities in a hospital observation room. Patients and their families are voluntarily participating in the experiments with the confirmation of the ethical board to use the recording in the research projects. The activities recorded for this dataset are, "preparing drink (P. Drink)", "talking on the phone (T. Phone)", "reading newspaper/magazine (Read)", "watering plant (W. Plant)", "preparing pill box (P. Pill box)", and "searching bus line (S. Bus line)". A sample of images for each activity is presented in Figure 3.3. Each person is recorded using an RGBD Kinect camera, of 640×480 pixels of resolution, mounted on the top corner of the room. The hospital dataset is recorded under EU FP7 DemCare project[3] and it contains 27 videos. The datasets are recorded at different times. For each person, the video recording lasts approximately 15 minutes. Domain experts have annotated each video regarding the ADLs that people perform. For this dataset, we have randomly selected 2/3 of the videos for training and the rest for testing.

Table 3.3 shows the obtained results by the BoW framework on this dataset. Based on the reported results we can conclude that:

- Although the trend of performance with respect to the codebook size is not constant, small codebook sizes perform relatively better. In most of the feature types (such as in Angle, HOG, TDD Spatial etc.), as the codebook size grows a performance drop is observed. This might be because of overfitting issue. Higher codebook size also increases the computational complexity of the classification because of nearest neighbour matching required for projection of feature vector to the codebook.

- HOG feature along with TDD Temporal achieve the best results. HOG feature is based on shape and usually does not achieve superior results for short actions. Each one of the activities in this dataset happens in a different location in the room. Using this feature together with trajectory point position information help to achieve better results since it helps the classifier to discriminate between different locations of activities.

- Similar to the GAADRD dataset, using geometrical features results in a poor recognition of activities. The reason for the failure of these features lies in being less representative of daily activities where subtle movements are important and the activities do not rely only on postural features.

[3] https://team.inria.fr/stars/demcare-chu-dataset/

Table 3.4 shows the results achieved by utilizing the Fisher Vector framework on this dataset. Based on the reported results we can conclude that:

- For some of the feature types (HOG, TDD Temporal, MBHY) bigger codebook size results in a boost in their classification performance whilst for some others (TDD Spatial, Angle) bigger codebook size results in a drop of classification accuracy. This shows that optimal codebook size is not only problem specific but also feature specific.
- Again, the HOG feature achieves the best results.
- TDD deep features performed comparably with dense trajectories when BoW encoding method is used. However, with FV encoding these features perform poorly compared to the dense trajectories. Deep features require big amount of data to achieve optimal performance. The lower performance might be due to lack of enough data needed to learn complex representation and large feature vector of FVs.
- Similar to the GAADRD dataset, using geometrical features results in poor recognition rates. However, Angle feature gains a performance boost (from 0.55 to 0.69) when they are encoded with this method.
- In general, Fisher vector encoding improves the quality of the classification except in TDD deep features where a drop of accuracy is observed while this encoding method is employed. This might be because of lack of data required for learning complex representation of deep features and FV encoding.

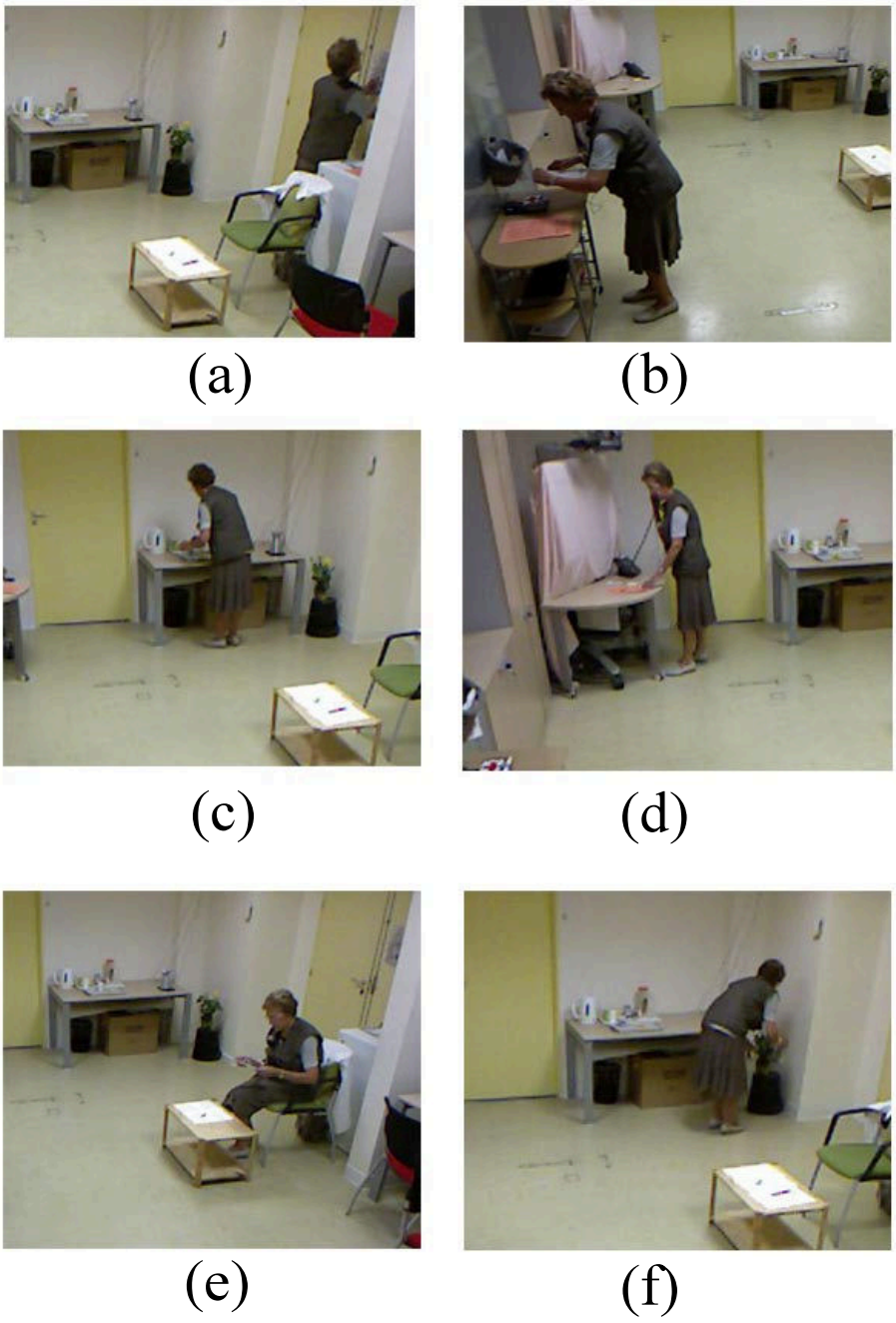
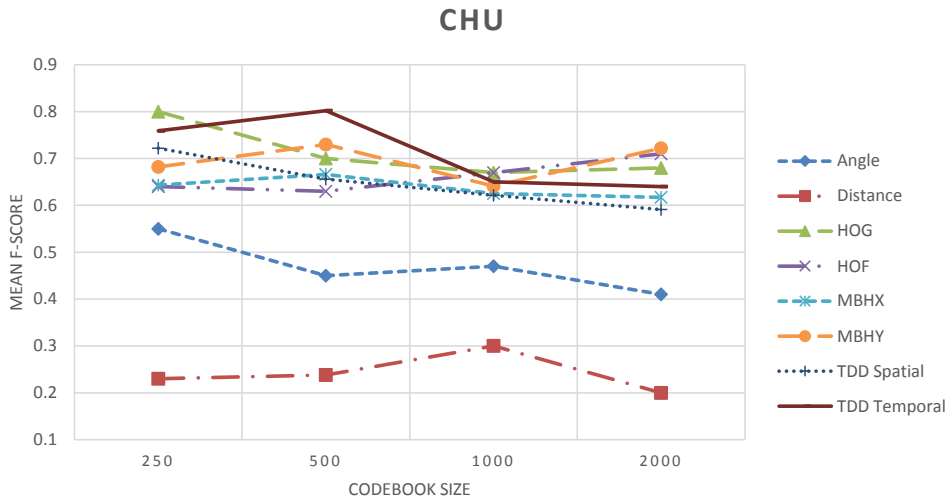


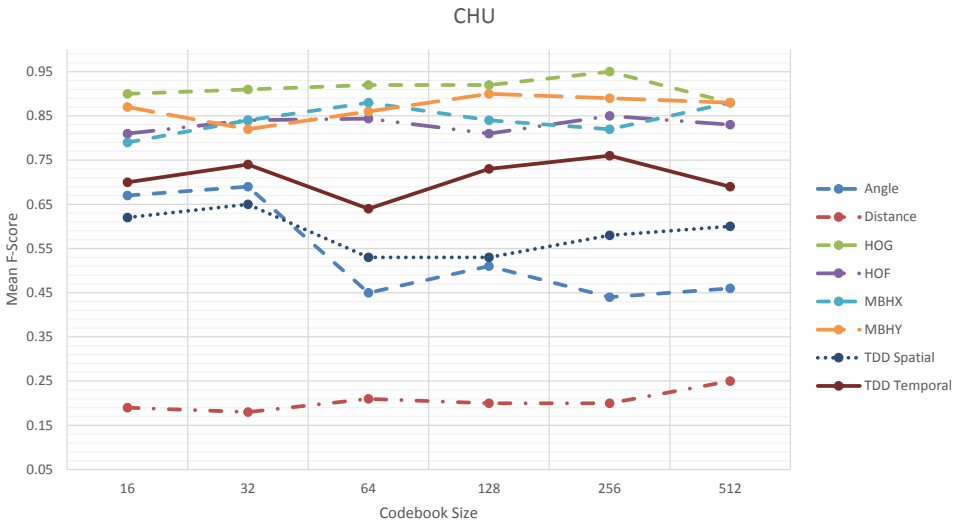
Figure 3.3: A sample of activities in the datasets: (a) Checking Bus Map, (b) Prepare Drug Box, (c) Prepare Drink, (d) Talking on the Phone, (e) Reading Article, (f) Watering Plant.

Table 3.3: Results of using different feature types by applying bag-of-words method on CHU Nice dataset. The plot shows F-Score values w.r.t. codebook size.



	250			500			1000			2000		
	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score
Angle	69.6	55.1	0.55	61.0	45.3	0.45	63.8	47.0	0.47	66.8	42.5	0.51
Distance	32.6	25.1	0.23	34.6	26.0	0.23	23.5	22.1	0.19	29.0	21.3	0.20
HOG	83.6	80.0	0.80	77.6	69.5	0.70	73.0	67.6	0.67	77.8	67.3	0.68
HOF	76.6	63.8	0.64	81.8	63.5	0.63	74.5	66.6	0.67	82.8	69.5	0.71
MBHX	70.3	62.6	0.64	72.8	64.6	0.66	74.0	58.3	0.62	76.3	59.1	0.61
MBHY	72.1	67.0	0.68	77.5	71.3	0.72	75.6	63.8	0.64	78.1	70.3	0.72
TDD Spatial	67.5	65.8	0.65	63.0	62.8	0.62	61.8	59.6	0.59	73.6	71.6	0.72
TDD Temporal	76.1	77.3	0.75	84.8	79.1	0.80	67.0	65.3	0.65	66.0	64.5	0.64

Table 3.4: Results of using different feature types by applying Fisher Vector method on CHU Nice datasets. The plot shows F-Score values w.r.t. codebook size.



	16			32			64			128			256			512		
	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score
Angle	76.6	66.6	0.67	77.6	67.5	0.69	66.1	45.5	0.45	66.3	51.0	0.51	61.8	44.8	0.44	57.1	44.0	0.46
Distance	20.5	21.8	0.19	19.0	21.3	0.18	22.3	24.3	0.21	20.5	22.8	0.20	21.3	22.6	0.20	27.1	27.8	0.25
HOG	94.6	88.3	0.90	95.3	90.0	0.91	96.0	90.1	0.92	94.8	90.5	0.92	97.0	94.6	0.95	90.5	86.6	0.88
HOF	91.6	79.6	0.81	91.8	81.5	0.84	91.5	81.8	0.84	91.0	79.6	0.81	92.8	83.6	0.85	90.6	81.8	0.83
MBHX	88.1	79.1	0.79	90.8	82.8	0.84	93.5	87.5	0.88	92.5	81.5	0.84	88.3	80.8	82.5	93.3	86.6	0.88
MBHY	92.3	86.3	0.87	89.1	80.6	0.82	92.5	86.6	0.87	93.8	89.5	0.90	93.3	87.3	0.89	90.6	86.8	0.88
TDD Spatial	64.3	62.8	0.62	69.6	65.0	0.65	65.5	52.0	0.53	60.6	52.3	0.53	75.1	57.6	0.58	65.0	60.0	0.60
TDD Temporal	84.8	68.5	0.70	84.6	71.8	0.74	78.3	64.0	0.64	78.0	73.8	0.73	86.3	75.5	0.76	80.6	69.3	0.69

Table 3.5: Results of the best performances using different feature types by applying bag-of-words and Fisher Vector method on GAADRD and CHU Nice dataset. The plot shows F-Score values.

Descriptor Method		GAADRD		CHU	
		BoW	FV	BoW	FV
Geometrical	Angle	0.61	0.55	0.55	0.69
	Distance	0.24	0.16	0.23	0.25
Dense Trajectories	HOG	0.79	0.90	0.80	0.95
	HOF	0.70	0.79	0.71	0.85
	MBHX	0.70	0.85	0.66	0.88
	MBHY	0.72	0.84	0.72	0.90
TDD Deep	Spatial	0.79	0.89	0.72	0.65
	Temporal	0.79	0.87	0.80	0.76

3.8 Conclusions

In this chapter, we discuss the two frameworks we use to produce our baseline evaluation of the two daily living activity datasets. The framework is based on local feature detection and extraction. For the encoding, in the first method we used the bag-of-words method and in the second method, we followed Fisher Vector encoding paradigm. Finally, for the classification, we use linear SVM. For the evaluation of the two supervised frameworks, we utilize three different feature types: Geometrical Features, improved Dense Trajectories, and Trajectory-pooled Deep-Convolutional Descriptors. The deep TDD features use the same detection method as of the dense trajectories, however, these features uses convolutional maps based on neural network for feature extraction. Based on the conducted experiments we can conclude that usually bigger codebook size results in a better accuracy. However, the results are not ideal when bigger codebook size makes overfitting issues. HOG and temporal TDD outperform the other descriptors (except that the temporal TDD achieves poor results compared to other descriptors when it is encoded with FV on CHU dataset). Among the temporal descriptors, MBHY descriptors achieve competitive performance with appearance descriptors like HOG.

In table 3.5 we provide comparison of Geometrical, Dense Trajectories and TDD methods. The results show that hand-crafted and CNN features achieve competitive results. In Dense Trajectories, FV encoding always improves the performance. In general, one main reason for miss-classification of the activities is that the feature detection process fails for the activities including less motion as it relies on optical flow values. Accordingly, miss-classification is unavoidable in absence of reliably detected features.

Chapter 4

Activity Discovery, Modeling and Recognition

"You never change things by fighting the existing reality. To change something, build a new model that makes the existing model obsolete."

- Buckminster Fuller

Contents

4.1 Introduction

In order to describe an activity semantically, we can adapt a notion of resolution dividing an activity into different granularity levels. For an ADL we can always name a sub-activity (*eg.* "Preparing Drink" activity consists of sub-activities such as "Pouring Water to Boiler" and "Adding Coffee" etc.). An ideal activity recognition system should be capable of adapting to different activity resolutions and producing meaningful information when it is demanded. Such system should model the activities in a way that the models describe multi-resolution layers of activities by capturing their hierarchical structure and their sub-activities. Hence, the system can move between different layers in the model to retrieve relevant information about the activity. An inherent problem for such a method is the automatic delineation of the activities indicating the time-span where an interesting activity is happening (e.i. beginning and ending of an activity). The problem comes from the interpretation of information coming from different resolutions. For example, imagine a subject sitting on a chair and reading a book. Doing that, the subject moves his/her arm to turn the page. If only one resolution is considered, turning the page changes the label of activity from "Sitting on chair" to "Turning a page" despite that the person still "Sitting on chair". While a system with flat representation focuses on one activity at a time, an hierarchical modeling enables us to model activities considering their constituents in different resolutions. Such a system is capable of detecting the beginning and ending of an activity and its sub-activities at different levels (activity discovery). Then, the modeling is the grouping of these patterns using extracted semantically-rich information. An activity model is a package abstraction containing all the relative information of that activity and its sub-activities.

In this chapter, we describe different parts of our proposed frameworks to address these challenges in activity recognition. The main components of the framework can be decomposed into the following components:

- Input data.
- Feature extraction.

- Automatic activity discovery to detect interesting activities in the scene.
- Creating the models to uniquely characterize the activities by deriving relative information and constructing the hierarchical structure.
- Providing semantic cues for the models by supervision.
- A matching mechanism to compare and recognize newly detected activities.
- Scene region refinement and updating previous models .

Next, we describe the inputs of the framework in section 4.2. The extracted features from the input stream is explained in section 4.3. In the section 4.4 the main parts of the framework for automatic activity discovery, modeling and recognition is described. We conclude this chapter in section 4.6.

4.2 Input Data

4.2.1 RGB

The framework receives RGB images from the capturing device and utilizes it for feature extraction. RGB images are used to extract the features. The framework can extract global features using the whole information in the pixels and the selected image patches or it can obtain local descriptors. In addition to the handcrafted features, the representations can be learned using deep convolutional nets. Our framework uses this input as its source to obtain image features.

4.2.2 Depth-map

With the emergence of low-cost RGBD sensors such as Microsoft Kinect and Asus Xtion, capturing both RGB and depth-map simultaneously and in real-time has become possible. The provided depth map made feasible the extraction of spatial 3D information in the scene by measuring the distance of each pixel relative to the location of the sensor. The measured distance by these sensors is reported in millimeters. The effective working range of the sensor is between 0.5 meters to 7 meters and the accuracy of the measurements drops as an object moves further from the sensor. However, the obtained depth-maps from the RGBD sensors are noisy and error-prone and contains missing values due to the problems such as occlusion or infra-red beam absorption of the scattered beam from the sensor by the surface material of the objects. Despite imperfect measurements, depth-maps help a lot to improve the accuracy of the tasks such as scene segmentation and people detection. Throughout this work, different parts of our framework take advantage of the retrieved depth-maps from detection and tracking of the people in the scene for activity modeling, to segmentation of the body parts such as hands for gesture recognition.

4.2.3 Skeleton

RGB-D sensors are also capable of extracting human skeleton information from depth-maps by the assistance of a skeleton detection algorithm [209]. Inspired by early work of Johansson et al. [99] these algorithms detect 20 body joints of a human skeleton in real-time. Nowadays, retrieving skeleton information from RGB image is prevalent by advancements achieved in Deep Neural Network [186, 244]. By taking benefit from powerful GPUs, these methods detect body parts with reliable accuracy and in near real-time. The skeleton information is utilized by the framework as long as it is provided and reliable.

4.3 Feature Extraction

Here, we explain the low-level and high-level features extracted from different modalities for our system. These features are extracted directly from input sources which are explained in the previous section. Whenever possible, different low and high-level features can be added or removed from the input without affecting the framework. For example, the quality of the skeleton detection is not reliable or not available for some datasets and using skeleton information does not improve the quality of the activity descriptions. However, the type and quality of the selected features can affect the performance of video characterization and is dependent on the environment and the intended application. As it is observed in the previous chapter, geometrical features are not so useful for recognizing long-term daily activities, whilst, they showed quality performance in recognition of short-term actions. Therefore, depending on the task, each one of those features can be added or discarded in the modeling process without affecting the other parts.

4.3.1 RGB Features

As described in chapter 3, given a set of RGB images, first, a feature detector is used to detect a salient point of interest. Second, the features are extracted around the detected interest-points. In our framework, we use RGB images to extract different feature types (either handcrafted or deep features) and also to detect and to globally track a subject throughout the scene. The extracted features are explained in the previous chapter (3) when we describe our supervised framework.

4.3.2 Depth-map

We do not extract features using depth-map information in this work, however, depth-map is used in the algorithm of our global tracker (section 4.4.1). Moreover, depth information

is utilized to segment hand contour in gesture recognition task. This module (Chapter 6) is integrated into the activity recognition framework helping to have a more fine-grained analysis of body part motion for the created models. Later, these features are fused with skeleton information making a multi-modal handcrafted feature created for recognition of upper-limb gestures. In this case, the skeleton joints are used as a point of interest detector on RGB images as depth-maps do not provide sufficient textural information.

4.3.3 Skeleton

Many action recognition methods based on skeleton information use representations based on skeleton joint positions. The actions are described with the constituent poses that are modeled relatively by pairwise joint positions or angles between joints. Inspired by those methods we created geometrical features based on relative positions of the skeleton joints and the calculated angles between them. The details about these features is described in section 3.2.3 of chapter 3.

4.4 Automatic Activity Discovery and Modeling

4.4.1 Global Tracker

In order to achieve an understanding of long-term activities in an environment, information regarding the global position of subjects is essential. The global position is represented as 3D points which help to understand the relative position of the subject with other subjects and the surrounding objects. A sequence of obtained global positions creates a trajectory. Detection and tracking of subjects are challenging tasks and are still open research fields. Trajectory information plays a crucial role in our framework hence, precise calculation of subject trajectories is important and essential for recognition task.

Understanding of smaller parts of a long-term video can help to understand the longer video. By clipping videos to smaller chunks we can explain the long videos. Most of the methods in the literature use fixed size video chunks to clip the videos without any plan to have semantic content in each chunk. However, based on obtained trajectories, we propose a method to achieve variable size video chunks where each chunk contains a meaningful part of an activity.

The tracking of a subject using a fixed video camera still carries various vision-related challenges. In many of the proposed approaches, an object which is classified as a person in consequent frames is tracked spatiotemporally by establishing a link using re-identification techniques. The challenges vary from one scene to another, however, most

Figure 4.1: Examples of the people detection and tracking in the CHU (Left) and GAADRD (Right) datasets.

of the time the problem occurs in the frame-to-frame tracking of the detected subjects. On the other hand, the challenge for detection algorithms usually is related to complete or partially occluded subjects, change in illumination condition of the environment etc.

For detection and tracking algorithm, knowledge of the targeted environment plays an essential role. The conditions of the environment, should be considered in the design of the algorithm. In indoor environment illumination condition is a key subject. Especially for long-term recordings such as in daily activities which go on for hours or even days, the illumination can change drastically during the recordings. In such scenarios to improve the performance, other than the type of utilized method, type and location of the sensor should also be taken into consideration by the designer of the system. For example, in our scenario which happens in a nursing or smart-home, placing a camera at a fixed height and position can help not only to prevent occlusion of the subject but also make the activity discovery much easier in the later steps.

For person detection, we use the algorithm in [166] that detects head and shoulders from RGBD images. Trajectories of people in the scene are obtained using the multi-feature algorithm in [38] that uses features such as 2D size, 3D displacement, color histogram, dominant color and covariance descriptors and adapts online tracking parameters. Figure 4.1 shows samples of detection and tracking method on the two datasets that we have evaluated. Due to above-mentioned challenges such as artificial light, problem such as miss-detection is observed. However, the overall performance of the tracker is sufficiently reliable for our framework to generate meaningful activity models.

4.4.2 Contextual Information and the Scene Models

In most of the trajectory-based activity recognition methods, apriori contextual information is ignored while modeling the activities. These methods [86, 145, 13, 182] generally first, cluster the trajectory points to detect important locations in the scene such as roads or regions and second, extract some features (such as speed and orientation of the trajectories) from the detected interesting cluster regions in order to semantically classify them into the regions corresponding to different activities and then they define a recognition mechanism to compare new information with the constructed model for each region. While these kinds of models suit some of the recognition tasks in highly structured scene settings such as traffic junction scenarios and detecting abnormal/infrequent behaviors and incidents, they face serious challenges. The main problem is related to how these methods interpret the notion of clustering. In these methods, motion vectors (such as optical flow) are clustered without analyzing the source or underlying meaning of the motion (E.g. an irrelevant object passes by the target object and its motion information affects the final model of activity). This relatively simple way of abstracting the information in a single layer results in a different problem in activity modeling.

First, usually, these methods are not successful in modeling complex activities which take place in unstructured environments such as in an apartment (activities consist of multiple global and local motions). The modeling relies on the repetitive time-constrained trajectory patterns assuming that these repetitions identify certain activities. Therefore, these frequent motion patterns that together make a cluster, characterize a unique activity which can be semantically described. These assumptions make sense in structured scenes where static objects in the scene make a structure that restricts the motion pattern of the tracked subjects. For example in the case of traffic scenario, roads and traffic rules define the possible routes and allowed speeds for an object (car or pedestrian) to take. In a junction, a car can “Turn Left” or “Turn Right” and there is no other way around. And if it does something out of these two possibilities, it can be considered as an “abnormal behavior”. So, the activities are less complex and the detected clusters can clearly explain the possible activities. The situation is different in an unstructured scene where making strong assumptions about the motion of the tracked object are not possible. Take an apartment as an example. There is weak rule dictating how to go from one place to another in an apartment and at what speed (Different subjects can take different pathes to move from a location to another). In such scenarios, the complexity level of activities elevates even higher, since rather than cars, humans are involved in those activities. In these activities, multiple motions typify a single activity. Most of the time, a single motion cluster is not sufficient to model complex activities that include concurrent movements

(e.g. walking and talking on the phone), hence, this results in failure in the recognition stage. Additionally, the detected clusters of frequent motions in the scene are difficult to understand and hard to explain semantically due to limited information related to the generated clusters.

One of the main reasons for the above-mentioned limitations in modeling the activities is the absence of the prior knowledge of the scene. Such knowledge carries information that can be very important in characterizing the activities. Some works [151, 79] have proposed to include prior contextual information in the models of complex activities. To characterize the video instants, these prior works suggest to use the knowledge which is usually based on detection of semantic events. The events are detected by inferring relations between different regions in the scene and the extracted perceptual features. The perceptual features could be an object type or a color descriptor. In the end, the activities are modeled given the detected events as their building blocks. These methods propose different ways of using information from scene regions either in detecting the building-block events or in the construction of the models. For example, some methods try to learn paths among different regions in the scene and some others use Context Free Grammars (CFG), n-grams or Finite State Automatons (FSA) to find relations between the events [145, 229, 79, 232].

Even though these event-based methods sufficiently comply with challenges in modeling activities semantically, most of them are accompanied by a couple of shortcomings. In most of these methods [46, 45], the scene regions are determined manually which is a tedious task and requires a great amount of supervision. If the setting of the environment changes, a supervisor needs to modify the scene models and redefine the regions. In addition, subjective opinion of the system's user is included and affects the definitions. Moreover, many of the methods use single resolution definition of the regions which limits them to have a multi-resolution characterization of the activity models by taking advantages of activity and sub-activity relations.

To cope with these challenges and limitations, our framework performs an automatic learning of the meaningful scene regions by taking into account the subject trajectories (section 4.4.3). Learning of the zones are performed at multiple resolutions. By tailoring zones at a different level of resolution, a hierarchical scene topology is created. The learned regions can get modified using a data-driven approach where the handcrafted and automatic models are combined (chapter 5). The multi-resolution model of the scene regions enables characterizing interesting regions and sub-regions in the scene. In later stages of the framework's pipeline, the constructed scene models are used to discover

activities taking place in the scene regions (section 4.4.6).

A topology is defined as a single-resolution scene representation composed of scene regions. The aim is to find interesting regions in the scene that can be semantically explained (e.g. location where the tracked subject interacts with specific fixed object). Each region contains properties that can reveal spatial and temporal characteristics of a scene. For example, "coffee machine" has a fixed location and it takes about 5 minutes to "prepare coffee". A topology is computed automatically using perceptual information from the subjects in the training set. The goal is to detect regions that are commonly used by the training subjects and gives a prior knowledge about relations of a specific activity with a certain location. Learning of the topologies normalizes the scene space, makes the regions semantically explanatory and filters out the noise coming from the acquisition process. If a location (*eg.* "coffee table") is commonly used for an activity (*eg.* "preparing coffee"), we can make an assumption that it is very likely that the subjects will use that region again for the same purpose in the future.

4.4.3 Topology Representation

A topology at level l is defined as a set of scene regions (SR):

$$T_{level_l} = \{SR_0, ..., SR_{k-1}\} \tag{4.1}$$

where k indicates the number of the scene regions. This parameter defines the resolution of the topology. Lower the level means coarser level resolution (capturing bigger areas such as the kitchen) whilst, the higher level results in finer resolution (capturing smaller areas in the kitchen such as "fridge area"). The extracted trajectories of the subjects in the training set are used as input for training the topologies. There is no restriction on the temporal occurrence of the training samples. Moreover, different subjects can be tracked at the same time, producing independent training data. Therefore, input data is defined as:

$$Input = \{Seq_1, ..., Seq_n\} \tag{4.2}$$

Where $Seq_i = Traj_1, \ldots, Traj_T$. i is the label of the tracked subject and T is the number of trajectories in each sequence. Each scene region characterizes a spatial part of the scene and will be represented as a Gaussian distribution: $SR_i \sim (\mu^i, \sigma^i)$.

The clustering takes place in two stages. These two stages clustering helps to reduce the effect of outlier trajectory points in the overall structure of the topologies.

In the **first stage** the interesting regions for each subject in the training set are found by clustering their trajectory points. For a more precise characterization of the regions, instead of trajectory points, other features such as the vector of variation of motion or skeleton can be utilized. For each Seq the clustering algorithm produces k clusters:

$$Cluster(Seq_i) = \{Cl_1, ..., Cl_k\} \tag{4.3}$$

Each resulted cluster characterizes the scene based on the motion information of subject i. μ and ω parameters of the distribution of the SR_i are obtained from the clustering. C^th cluster center (Cl_c) corresponds to scene region k (SR_i). For SR_i, μ is the spatial coordinate of the cluster centroid:

$$SR_i(\mu) = centroid(Cl_c) \tag{4.4}$$

and the standard deviation σ is computed from the point coordinate sequence of the trajectory set.

The **second stage** of the clustering merges the individual scene regions into a single comprehensive set of regions. Each region is a new cluster (Cl) in the second stage that consists of the obtained cluster centroid in the first stage. These new clusters explain the spatial regions that are shared between all of the training subjects. We use K-means algorithm for the clustering. The value of K is set equal to the number of daily activities in the ground-truth.

As explained in the previous chapter, K-means [144] is an unsupervised learning algorithm to solve clustering problem. The algorithm tries to classify a set of data into a certain number of clusters (K clusters). In the first step, K centroids, one for each cluster, are defined. Then each data point is associated with the nearest centroid. In the next step, a new set of cluster centroid is detected from a newly generated clusters. The new centroid is the center of the resulted cluster. After detecting K new centroids, the new association between data points and those centroids is created. This procedure continues iteratively until the position of the centroids does not change and the loop converges. To find similarity between a data point and a centroid, K-means needs a distance measurement criteria. Different distance measurements are defined for K-means. Some of them are based on Euclidean distance or a variation of it (such as City-block distance). Some other measures are based on correlation coefficients such as Pearson correlation, Spearman's rank correlation and Kendall's τ. In our experiments, we use

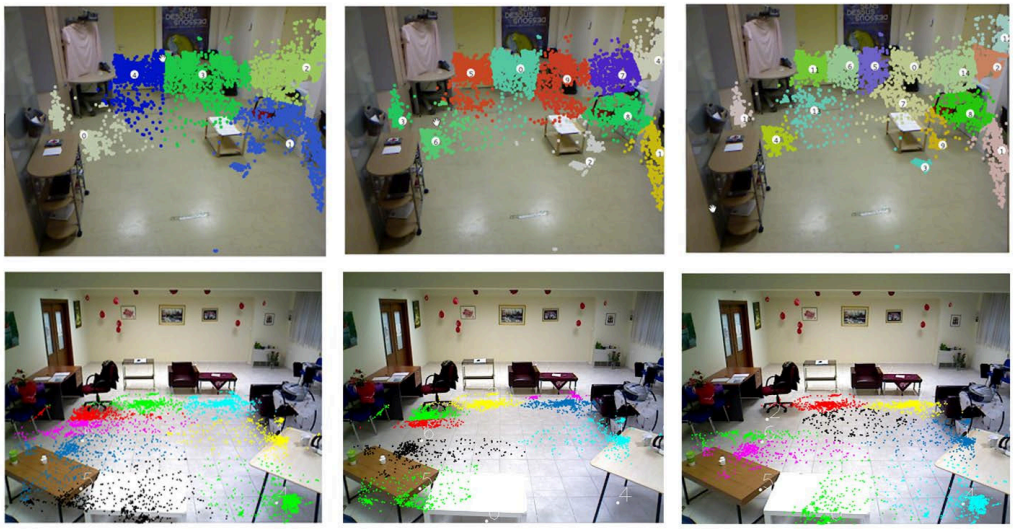


Figure 4.2: Example of k-means clustering using city-block and Euclidean distance measurements. The top row shows samples of CHU Nice dataset clustered using City block distance measure. The number of clusters is set to 5, 10 and 15. Most of the discovered regions can be associated with a semantic activity concept. There are also scene objects that the activity can be associated with. In the below is the GAADRD dataset clustered using Euclidean distance measure.

Euclidean-based measures (Especially City-block distance) to detect the scene regions since they produce acceptable clusters compared to the other measures. Figure 4.3 depicts examples of the calculated scene regions in two hospital rooms in CHU and GAADRD datasets. The scene regions are created with the information extracted from videos of 15 subject performing activities during 20 minutes. The clustering performed at three different resolutions and each detected region can be explained semantically.

A topology provides an abstraction of the scene in a single resolution. However, semantic explanation of spatial regions in the scene has different levels of abstraction. At each level, a region can be decomposed into sub-regions. Accordingly, we define a scene model as a set of scene regions (topologies) at different resolutions:

$$SceneModel =< Topology_{highlevel}, Topology_{midlevel}, Topology_{lowlevel} > \quad (4.5)$$

We create a model with topologies at three levels, each one aiming to describe the scene at the high, medium and low degree of abstraction. The scene model with three layers of spatial resolution constructs a structure of the scene that explains the whole scene spatially. At the higher level of spatial resolution it represents main activities taking place

in those regions, while in lower spatial level it can indicate finer motion of the subjects and sub-activities that build up the main high-level activities. At first, the value of resolutions at each level is set by the user, later, the optimal values are found by cluster analysis or by combining provided supervised information coming from manually drawn zones of description based component of the framework. Based on the experiments, the value of spatial resolution in high level is usually close to the number of activities in the ground-truth. It should be noticed that the clustering process at each level is performed independently for each subject in the training set and then the final cluster centers are determined. Then, the links between the layers (activities and sub-activities) are established with notion of neighborhood introduced in section 4.4.9.

4.4.4 Bayesian Information for Cluster Analysis

Despite its effective solution for general clustering problems, K-means algorithm suffers from two major flows. When the number of data points grows, the K-means scales poorly and the number of clusters k should be provided by the user to the algorithm. There are several proposed solutions to make K-means computationally effective, however, in our study, finding the optimal value of k interests us the most. We follow the method suggested by Pelleg et al. [178] to find the optimized value of k. In this method, we use Bayesian Information Criterion (BIC) to analyze the space of cluster locations and cluster numbers. After each run of the K-means algorithm, the method decides which subset of the centroids should get split, in order to better fit the data. This decision is made by computing BIC. The method starts with 2 clusters and evaluates different k values until an upper bound is reached. It is possible to set any reasonable boundary related to the imposed problem. After the evaluations, the clustering with the highest BIC score selects the best k value. As in the original work, to find a new cluster, two strategies are followed: In the first strategy, for each detected centroid, a new centroid nearby is picked and after, the new score is calculated. If the score gets higher, the centroid is accepted as the new centroid. However, this strategy is expensive since it runs K-means algorithm one more time for each cluster centroid. On the other hand, the second strategy, picks half of the current centroid based on a criterion (*eg.* region size of their cluster) and then splits those clusters and recalculates the score. If the score reaches a higher value, the splits get accepted. In this strategy, choosing the best criteria to pick the centroid and unable to pick only the clusters requiring a split are the main drawbacks.

To calculate the BIC score, assume the data D and a set of alternative models (clustering with different k values) are given. As mentioned before, the data points in each one of the cluster are defined as a Gaussian distribution where μ_i is the coordinate of the i^{th} centroid. Here we use i to refer to the index of a centroid as the closest centroid to data

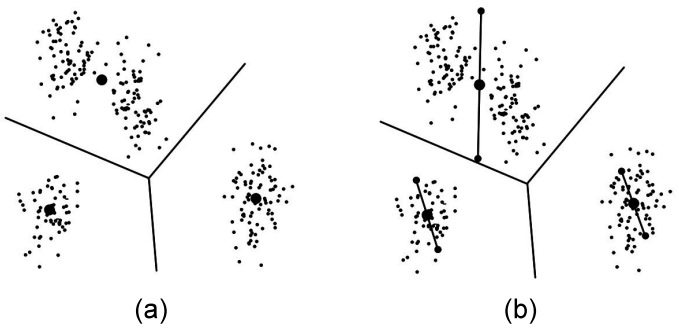


Figure 4.3: (a) shows the result of clustering after applying K-means on the data points. (b) shows the original centroids after splitting into two children.

point x_i. Therefore, D is the input set of points in an iteration where $D_i \subseteq D$ is set of points having μ_i as its closest centroid. To chose the best model the posterior probabilities are calculated. To approximate these probabilities for one centroid M_j the following BIC formula is used:

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2}.logR \tag{4.6}$$

where $\hat{l}_j(D)$ is the log-likelihood of the jth model. p_j is the number of parameters in M_j. R is the total number of data points belonging to the centroids under consideration. In this approach, the maximum-likelihood estimation (MLE) of the variance with different values of K is calculated as:

$$\hat{\sigma}^2 = \frac{1}{R-K}\sum_i (x_i - \mu_{(i)})^2 \tag{4.7}$$

x_i is the data points belonging to centroid i. Accordingly, the point probabilities of the model are calculated as:

$$\hat{P}(x_i) = \frac{R_{(i)}}{R}.\frac{1}{\sqrt{2\pi}\hat{\sigma}^M}\exp\left(-\frac{1}{2\hat{\sigma}^2}||x_i - \mu_{(i)}||^2\right) \tag{4.8}$$

Where $R_{(i)}$ is the number of data points belonging to model M. The log-likelihood of the data is:

$$l(D) = \log\prod_i P(x_i) = \sum\left(\log\frac{1}{\sqrt{2\pi}\hat{\sigma}^M} - \frac{1}{2\sigma^2}||x_i - \mu_{(i)}||^2 + \log\frac{R_{(i)}}{R}\right) \tag{4.9}$$

By fixing $1 \leq n \leq K$ and focusing only on data points in the set D_n with centroid n and replacing maximum likelihood estimate in the equation yields:

$$\hat{l}(D) = -\frac{R_(n)}{2}\log(2\pi) - \frac{R_n.M}{2}\log(\hat{\sigma}^2) - \frac{R_n - K}{2} + R_n \log R_n - R_n \log R \qquad (4.10)$$

In order to extend the calculation to all centroids, we sum log-likelihood of individual centroids, assuming that probabilities of points belonging to all centroids are the sum of probabilities of individual centroids. The BIC is used globally to choose the best model and also locally for cluster split tests.

4.4.5 Primitive Events

To fill the gap between low-level image features and high-level semantic description of the scene, an intermediate block capable of linking the two is required. Here we describe a method that defines a construction block for learning the activity models.

Most of the proposed unsupervised methods [155, 248] aim to make the link in a single step by creating activity models directly from the image features. Although these methods succeed to detect the occurrence and recognize the activities, the produced models suffer from several problems. If required, it is difficult to update and modify the produced models manually. Furthermore, the models usually have a unique design and are only compatible with specific training data. For such models, if the input of the system changes moderately, new models need to be generated for the same activities. Moreover, the generated models are very complicated for human interpretation. It makes difficult to explain system failures and assess the robustness of the models.

With a deeper look at the activity generation process, it can be inferred that the abstraction of low-level features into high-level descriptions does not happen in a single step. This transition is gradual. To remedy, we use an intermediate representation and we call it Primitive Event (PE). Having two consecutive trajectory data points ($Traj_i$ and $Traj_j$), by using their distances to the cluster centroids, we can find the scene regions that these points belong to (StartRegion and EndRegion). A primitive event is represented as a pair of directed scene regions of these trajectory points:

$$PrimitiveEvent = (StartRegion \rightarrow EndRegion) \qquad (4.11)$$

where *StartRegion* and *EndRegion* variables take values of SR indexes. For example, if

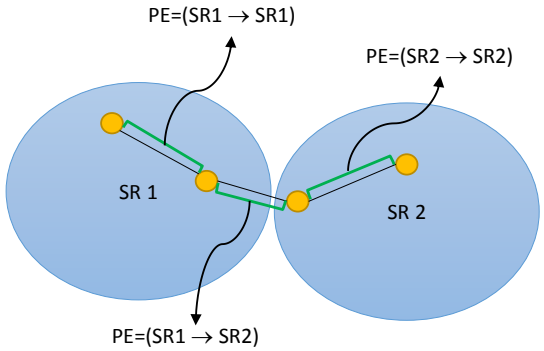


Figure 4.4: Example of calculating primitive events in two adjacent scene regions. The two PEs at the beginning and end of the trajectory shows that the subject stays in the same scene regions, whilst the PE in the middle shows a transition.

StartRegion of $Traj_i$: SR_2 and *EndRegion* of $Traj_j$: SR_4 then, we will have $(2 \rightarrow 4)$ as a primitive event. PE defines an atomic motion block and is used for characterizing the motion of a person in the scene. This way, a whole sequence of trajectory can be translated to PEs. Representing the extracted features with PE brings several benefits by linking the low-level features to a comprehensible semantic information and helps to fill the semantic gap:

- This makes the representation meaningful for humans by providing them with the information to describe a complex activity.
- It also helps to interpret failures of the recognition framework through an understandable representation.
- Having independent and modular units, the PE model enables the method to generate flexible models. The generated models are easier to modify and simple to update when a new PE is learned.

Primitive events are the first level of abstraction and can describe interesting events during a period. The definition of an event and a complete discussion about it can be found in [122]. The primitive events are created from the extracted trajectory features described in 4.4.1. They represent the current status of the subject in the scene and composed of automatically calculated attributes. The defined attributes of primitive events indicate their beginning/ending scene regions which determine the type of the events (Whether the status of the subject stays as previous or changes to a new one) and the time duration that a PE takes. A primitive event represents the overall status of the subject from the previous to the current extracted tracking features.

A *Primitive Event*'s type is *Stay* when the region labels (Such as SR1) stay constant between two time intervals. It is equivalent to a sequence of *stay*s in the scene region P:

$$Primitive\ Event = Stay_{P\_P} \tag{4.12}$$

When a *Primitive Event*'s type is *Change*, a change of region (from region P to regions Q) occurs between two successive time instants (i.e. two successive trajectory points). It is equivalent to a region transition:

$$Primitive\ Event = Change_{P\_Q} \tag{4.13}$$

The duration of the current status (stay/change) can be calculated simply by:

$$Duration = \frac{EndEventFrame - BeginEventFrame}{fps} \tag{4.14}$$

where fps is the frame rate that the video is recorded (calculated in frames per second).

The vector representation of Primitive Events (PEs) using Scene Region (SR) pairs creates a compact building block to represent an event of interest in a video sequence. Using the topology representation, it describes semantic information about an event learned through an unsupervised way. The perceptual features (extracted features from video) characterize a video in low-level and the topological representation of events achieves a semantic high level characterization of the events. The proposed abstraction fuses the two complementary information and describes the global motion of an object in a semantic way (e.g. a person moves from zone A to zone B).

Using a learned topology T, for every video sequence a corresponding primitive event sequence is calculated. This process produces a sequence of primitive events PE_{seq} created using topology T:

$$PE_{seq} = (< PE_1, \ldots, PE_n >, T) \tag{4.15}$$

A primitive Event sequence provides revealing information regarding the underlying structure of long-term activities. This structure contains particular patterns pertinent to different activities in various scene regions. Patterns of PE sequences help us to automatically discover long-term activities appearing in a video.

4.4.6 Activity Discovery

Automatic detection of activity delineations is a difficult challenge inherent to recognition frameworks. The goal is to find the time intervals where interesting activities are happening (beginning and ending of the activities). We refer to the detection of boundaries of activities as *Activity Discovery*. Annotating the beginning and end of activities is a challenging task even for humans when they are asked to do that. The start/end time of the annotated activities varies from one human annotator to another [187]. The problem is that humans tend to pay attention to one resolution at a time. For example, when a person is sitting on a chair the annotated label is "sitting". Later when the subject "moves an arm", s/he is still sitting. Discovering activities using a different resolution of the trained typologies helps to automatically detect these activity parts and sub-parts at different levels of activity hierarchy using previously created semantic blocks (Primitive Events).

Input for activity discovery process is a spatiotemporal sequence of activities described by primitive events. Activity discovery is the process of detecting related sub-sequences of events that match the previously learned patterns of events. The discovered events characterize all activities in a given video using a scene model composed of learned topologies. The activity discovery process is an automatic process and the process takes place at the same time in three coarse to fine resolutions. After the activity discovery process: 1) The beginning and end of all activities in a video are estimated and the video is automatically clipped. This can be seen as splitting a video sequence description into different groups of primitive events resulting from the interesting activities. This segmentation of video into small clips makes online activity detection and recognition possible. 2) The video is classified naively into discovered activities showing similar activities in the timeline.

In the context of daily activities a relationship between the transitions of subjects through different scene regions and the performed activities is clearly observable. For example, in "Preparing Coffee" activity in a kitchen, a subject goes back and forth through sub-regions of coffee machine and sink. This sequential loop involving two sub-regions of the kitchen characterizes the activity "Preparing Coffee". Therefore, as input, activity discovery takes three sequences of primitive events at three resolutions (created using topology T) for each trajectory sequence as in equation 4.15. A discovered activity (DA) is considered either as 1) staying in current state ("*Stay*") or 2) change of its current state ("*Change*"). Basically, a *Stay* pattern is an activity that occurs inside a single scene region

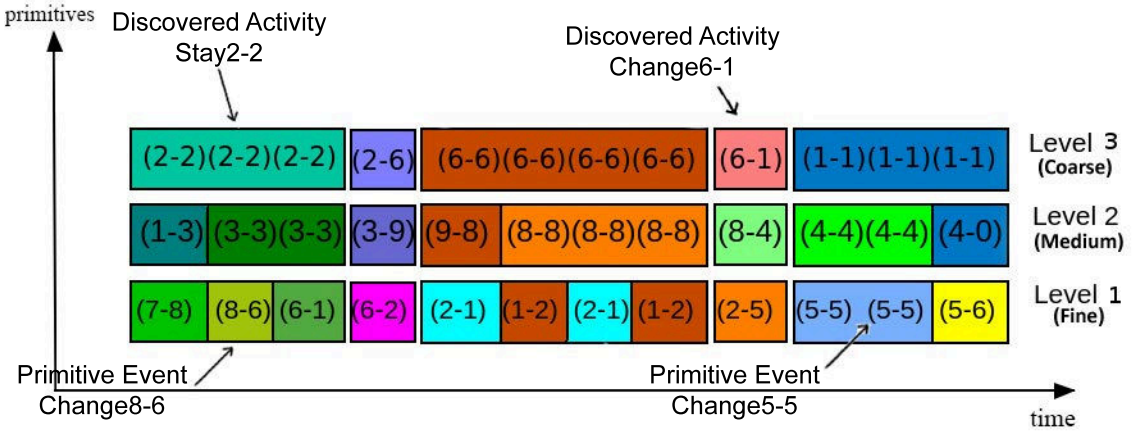


Figure 4.5: A sample video encoded with primitive events and discovered activities in three resolution levels.

and is composed of primitive events with the same type:

$$Discovered\ Activity = Stay_{P \to P} = \{Stay\ PEs\} \tag{4.16}$$

and a "*Change*" pattern is an activity that happens between two topology regions. A "*Change*" activity consists of a single primitive event of the same type:

$$Discovered\ Activity = Change_{P \to Q} = Change\ PE \tag{4.17}$$

Although the detection of primitive events takes place at three different resolutions, the activity discovery process only considers the coarse resolution. Therefore, after discovery process, the output of the algorithm for the input sequence is a data structure containing information about the segmented input sequence in the coarse level and its primitive events in two other lower levels. This data structure holds information similar to the structure in Figure 4.5. The algorithm for this process simply check for primitives' boundaries and construct the data structure for each discovered activity. The framework takes input features and creates three sequences of primitive events in three different resolutions and then, the activity discovery process takes the primitive events in the coarse level and detects the activities corresponding to the events. Each discovered activity at this level is defined by three attributes: its type (*Stay* or *change*), its *start frame* and *end frame*. The DAs appear sequentially covering the whole input video meaning that the *start frame* attribute of current discovered activity is next to the previous DA's *end*

frame attribute. Figure 4.5 illustrates an example of activity discovery process in three resolution levels. With DAs and PEs, it shows the hierarchical structure of an activity and its sub-activities. The DAs are not the activity models. Later, we will show how the activity models are created using the discovered activities and the primitive events combined with local descriptors information (Section 4.4.10).

This notion of representation brings a couple of benefits: The representation is simple and easy to understand by humans. The basic patterns can represent complex activity patterns. Although *Stay* and *Change* blocks seem very simple, by using multiple resolutions these basic blocks can be used to model and distinguish complex interactions. As illustrated in figure 4.6, the simple discovered activity pattern at coarser level is composed of complex interactions at finer resolution level.

In the case that sub-activities are generated using allowed interactions among more than 2 scene regions, even more complex representation of activities is required. Imagine events detected from the sequence of feature patterns between two scene regions. If we represent those events as words consisting two letters, we will have extracted words such as “abbba” or “baab”. Despite having the same letters, these sequences represent different activities due to different frequency of letters. The pattern of such words can be recognized using a uni-gram. Now, if we suppose that the order of letters also become important a more complex representation model such as n-gram would be required. Therefore, more letters (primitives) allowed, more complex/rich the structure of the word (activity) could become.

The discovered activities and their underlying primitive events are used for learning the activity model which is the subject of the next section of this chapter. Moreover, the activity discovery process performs temporal activity detection by finding starting/ending times for activities. This localization of activities in the videos is usually performed using a temporal sliding window. For example, in the proposed method of Hamed et al. [79] which is based on n-grams, the value of n is important and is required as the length of the sliding window to detect a pattern in tokens. Similarly, in [63] non-parametric Bayesian network is used to find the patterns in a fixed-length sequence of letters. In the sequence discovery stage, several parameters need to be set. Therefore, most of the sliding window based methods are time-consuming and require tuning of several parameters. The advantage of our representation over these methods is that our discovery process clusters letters into interesting words where the length of the words is adapted by the perceived features and learned scene regions. Additionally, most of the methods in the literature perform activity discovery with a single layer of resolution. Our method provides insights about the activities automatically and at different layers.

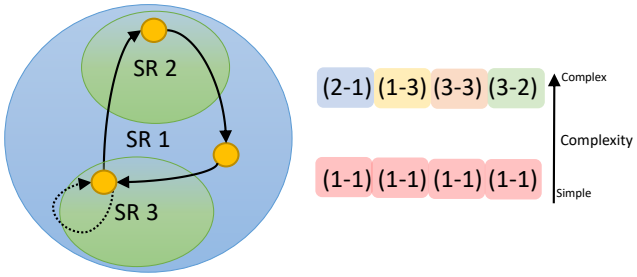


Figure 4.6: An example of a composite activity. Simple building block (stay and change) describes the activity with simple and complex relations of PEs at coarse and fine resolution respectively.

4.4.7 Extracting Action Descriptors

Although *Discovered Activities* present global information about the movement of people, it is not sufficient to distinguish activities occurring in the same region. Thus, we incorporate body motion information by extracting motion descriptors. We employ the approach in [233] which extracts motion descriptors around dense trajectory points as explained in section 4.3. Dense trajectories are sampled at each frame and tracked through consequent frames using optical flow. To avoid drifting, the trajectories are discarded after passing *L* frames. Because motion is an important feature to characterize the activities, we use the following descriptors in spatiotemporal volume around each trajectory point: HoG (histogram of oriented gradient) [48], HoF (histogram of oriented flow) [120] and MBH (motion boundary histogram) [233]. We extract these descriptors in a volume of *NxN* pixels and *L* frames. Then, we follow Fisher Vector (FV) approach to obtain discriminative representation. In supervised approaches, action descriptors are extracted from manually clipped videos and labeled. Instead, in our approach, we extract the descriptors for all *Discovered Activities* that are automatically computed. Usually, the daily activities take a long time to perform, and in most of such activities short-term actions such as "moving arm" do not explain much about the whole activity, the local descriptor information is extracted only for *Discovered Activities* at the first level of topology (coarse level of resolution). The descriptors are extracted knowing the first and last frames of each split segment detected in the activity discovery process. During experiments, we have selected $N = 32$

and $L = 15$. The extracted descriptors contribute in the modeling of activities either by training a supervised classifier based on a learned dictionary (Hybrid framework) or in an unsupervised way by clustering the calculated histogram of the encoded descriptors (Unsupervised framework).

4.4.8 Activity Modeling

To model an activity a formal language is used for describing its characteristics. The model is like a package consisting of different features describing the activity in a unique way. The description is made possible by obtaining different descriptors from the observed activities. A scenario is described using a set of possible features that vary from spatial and temporal to local and global features. The selection of the features mainly depends on the application objective. This means, for example, if measuring the time span of an activity is important for a user, including the *duration* of activity in the models should be considered. **Discriminative abilities of the features** which is the capability of the features to uniquely represent the activity. For example, a feature capable of detecting body parts can characterize an activity by posture feature, whilst, a less accurate feature only indicates the center of mass of a subject. **Reliability of features** makes them contribute in constructing reliable and less erroneous models. In modeling, the trade-off among these criteria should be considered. The goal is to make models with high discriminative strength and less susceptible to noise. We use attributes of an activity and its sub-activities (such as local descriptors) for modeling and the learning is performed automatically using the DAs and PEs in different resolutions. Learning such models enables measuring the similarity between them. With a similarity measure, other than automatic activity recognition, also enables understanding of variance in performing an activity and detection of infrequent activities. However, modeling the frequent activities can produce a generative model for different scenarios in the scene. Such models can be used to describe expected activities and also higher level tasks such as grouping subject in different profile categories and retrieve their daily activity patterns. For example, a person can be categorized as “inactive” if s/he spends long duration in “TV” scene region performing “Watching TV” activity. The models that we propose in this section have all these potentials by capturing the hierarchical structure of activities.

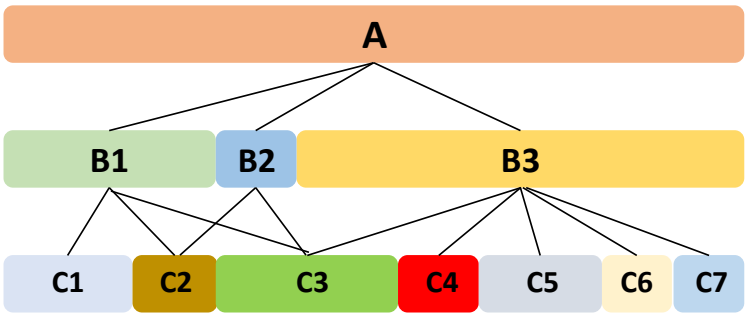


Figure 4.7: Illustrates the neighborhood of discovered activity A

4.4.9 Hierarchical Neighborhood

The hierarchical finer representation of an activity A at resolution level l is the recursive representation of the links between A and its primitive events B_i at the finer resolutions:

$$A_{neighborhood} = ((B1, B1_{neighborhood}), \ldots, (Bn, Bn_{neighborhood})) \tag{4.18}$$

where, $B1, \ldots, Bn$ are the primitive events of A in the next finer resolution. The link between the different levels are established using temporal overlap information. For example primitive event B is sub-activity of activity A in a higher level if their temporal interval overlaps in the activity timeline. Formally, B is sub-activity of A if the following statement holds:

$$\begin{aligned} &((startFrame_A \leq startframeB) \wedge (endFrame_A \geq startFrame_B)) \\ &\| \ ((startFrame_A \leq endFrame_B) \wedge (endFrame_A \geq endFrame_B)) \\ &\| \ ((startFrame_A \leq startFrame_B) \wedge (endFrame_A \geq endFrame_B)) \\ &\| \ ((startFrame_A \geq startFrame_B) \wedge (endFrame_A \leq endFrame_B)) \end{aligned} \tag{4.19}$$

Figure 4.7 shows an example of a discovered activity A and its associated sub-activities layers forming hierarchical neighborhood of the discovered activity. Applying the formula 4.18 on this discovered activity, we can find the primitives in its neighborhood:

$$A_{neighborhood} = ((B1, (C1, C2, C3)), (B2, (C2, C3)), (B3, (C3, C4, C5, C6, C7))) \tag{4.20}$$

This automatic retrieval and representation of the neighborhood of a DA help creating the hierarchical activity models.

The model construction is based on learning characteristic of activities with 2D or 3D global and local information. The models are like an empty prototype of activities that we fill them with the descriptions of the target activities. There are different methods in modeling Natural Language employed to model the activities. The first group of these methods is based on frequency information such as repetition, concurrency, order and relevance of activities and their sub-activities. The *Frequency Models* such as *tf-idf*, *Successor-Predecessor*, *n-gram*, and etc. can only handle models with a single resolution and become unable to handle the modeling task when the hierarchical information can be used. Another category of methods based on natural language is the *Topic Models*. Topic models are probabilistic generative models that characterize a topic based on histograms of the vector calculated from words in the documents. Its most well-known variations are Latent Dirichlet Allocation (LDA) and Hierarchical Dirichlet Process (HDP). In LDA given a set of document, the problem is to classify the words into the different topics. HDP is a derivation of LDA that unlike LDA, the number of topics is not given a priori. Although the challenges in modeling with natural language methods look similar to the one in the video activity modeling, using these methods off-the-shelf for activity modeling confronts with two main problems. *First*, the language models assume a specific word ordering that comes from a language structure imposed by the grammatical rules. The long-term daily activities are unstructured and loosely constrained. There is no guarantee that in "Preparing Coffee" activity the subject will "Use the sink" or "Use the drawer" first. *Second*, these models are not ready for multi-resolution inputs and work with simple words, however, in activities more quantitative complex feature representations are required. Instead, we model the activities in multiple resolutions and with hierarchical order.

4.4.10 Hierarchical Activity Model (HAM)

Inspired by Hierarchical Dirichlet Processes, we introduce hierarchical activity models to capture hierarchical structure of daily activities by taking advantage of the hierarchical neighborhoods to associate different levels. The structure of activity models is represented by a tree that is similar to the structure of hierarchical neighborhood where each level represents a resolution. In the learning process, given a set of neighborhood of a DA as input ($A_{neighbourhood}$), the the goal is to group similar PEs to create nodes (N) of the activity tree. We use clustering for grouping PEs. For clustering, we use *Type* attribute of the PEs by grouping PEs of the same type in one cluster (illustrated with same color

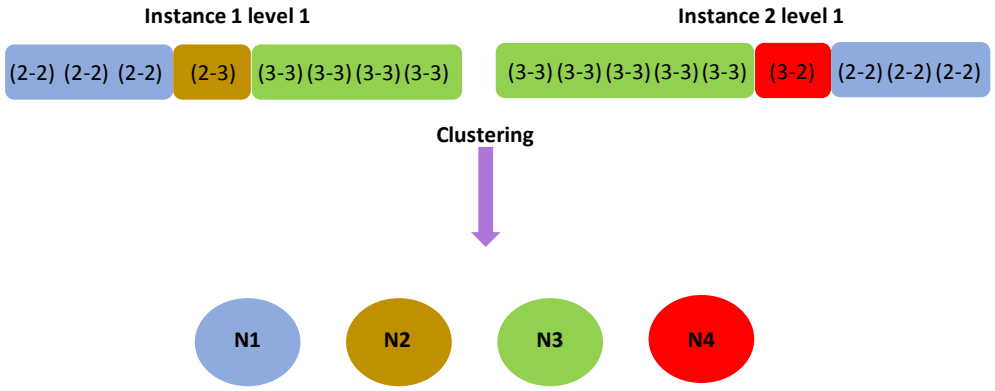


Figure 4.8: Clustering of the primitive events into nodes

in Figure 4.8). This process is repeated in all levels. For example let's assume that we have two instances of a semantically identical DA as input ($A1$ and $A2$). The hierarchical neighbourhood of these DAs are calculated as explained before and is used as input for this step: $A1_{neighborhood} = ((A1, (B1, C1, D1)))$ and $A2_{neighborhood} = ((A2, (D2, E1, B2)))$. The clustering of primitives of this DA (B_i, C_j, D_k, E_l) is performed at each level. The output of clustering is tree nodes (N_1, N_2, N_3, N_4) created based on $Type$ of the constituent PEs. After the clustering, the nodes of the tree model are determined. Next, the nodes are linked together to construct the hierarchical model of the trees. The links between the nodes are obtained from the activity neighborhood of each node (Figure 4.9 shows the complete procedure of creating an activity tree from neighborhood set instances of a DA). After linking, we have a complete tree structure of the given DA and now we can complete the model by adding attribute information for nodes of the tree. Each node in the activity tree contains information about the similar detected primitive events that share similar properties such as duration and type of the primitive and also similar sub-activities in the lower level. So, a node is the representative of all the similar primitives in that level. Each node has two properties. First, is the node's attributes storing information such as average duration of the constituent primitive events and second, is the information about parent node and the associated nodes in the lower level of the hierarchy. The nodes can keep different spatial and temporal attributes about the activity and its sub-activities. We define these attributes to keep information about global and local spatiotemporal properties of the discovered activities and primitive events.
The attributes consist of:

- *Type*: The type attribute of each node adapted from the underlying primitive

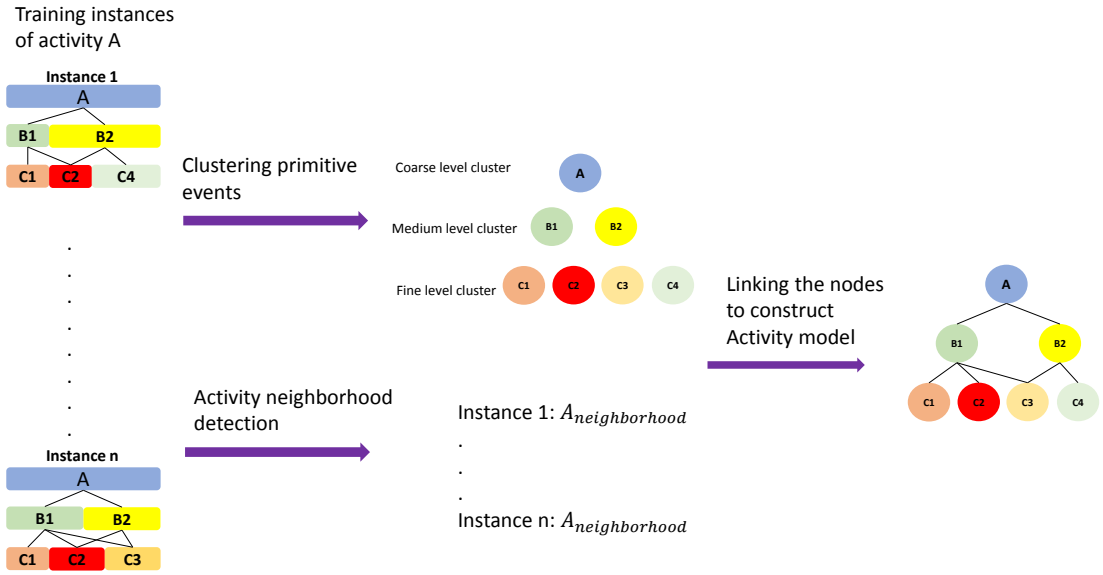


Figure 4.9: The process of creating activity tree. The PEs from the training instances are clustered into nodes and at the same time, the neighborhood set is detected. The final structure is constructed with those building blocks.

or discovered activity (in case of the root node). For node N, $Type_N = Type_{PE}$ *or* $Type_{DA}$ (Where $Type$ of PEs and DAs are either *Stay* or *Change* states).

- *Instances*: Is the list of PEs of training instances indicating the frequency of each PE included in the node.

- *Duration*: It is a Gaussian distribution $Duration(\mu_d, \sigma_d^2)$ describing the temporal duration of the PEs $(\{PE_1, PE_2, \ldots, PE_n, \})$ or discovered activities $(\{DA_1, DA_2, \ldots, DA_n, \})$ of the node. It is the frame length of primitives or discovered activity composing a node in the tree model. It is calculated as:

$$\mu_d = \sum_{i,j=1}^{n} \frac{(endframe_{PE_i or DA_j} - startframe_{PE_i or DA_j})}{n} \tag{4.21}$$

$$\sigma_d^2 = E[((endframe_{PE_i or DA_j} - startframe_{PE_i or DA_j}) - \mu_d)^2] \tag{4.22}$$

where n is the number of PEs or DAs.

- *Image Features*: It stores different feature extracted from the discovered activity.

There is no limitation on the type of feature. It can be the extracted hand-crafted features, geometrical or deep features (section 4.3). The different embedded features can contain information about pose (geometrical features), motion and appearance (dense trajectories and deep features). The image features are only available for the root node (DA level, coarse resolution). In the first variation of the framework (Hybrid), this attribute is the label of the activity classified by a trained supervised classifier from the instances in the training set. In the second (Unsupervised), it is the calculated histogram of the features of the instance in the training set.

- *Node association*: Indicates the parent node of the current node (If it is not the root node) and the list of neighborhood node in the lower levels.

The above-mentioned attributes, do not describe the relationship between the nodes. This relationship is important in overall description of the activities. In order to model the relationships between the node, for each node, we define two attributes regarding their sub-nodes: *Mixture* and *Timelapse*. *Mixture* shows the contribution of the type of sub-activities (*eg.* $Stay_{2-2}$) in the total composition of sub-nodes (in the created HAM of training instances). This number is modeled with a Gaussian mixture $\Theta_{type}^{mixture}$. *Timelapse* of the nodes (with same type and same level in different training instances) represents the distribution of temporal duration of the sub-nodes. This attribute is also represented as a Gaussian distribution $\Theta_{type}^{timelapse}$. The created HAM structure is a hierarchical tree that provides recursive capabilities. The calculation of the attributes and the score in the recognition task are performed recursively using the constructed HAMs.

4.4.11 Model Matching for Recognition:

To measure the similarity between the trained HAM models, different criteria can be considered. This criterion can vary from one application to another. While one application can emphasize more on the duration of activities, in the others, local motion can be more important. Although these criteria can be set depending on the application, we use a method to learn the weights and hence, the importance of each feature.

The recognition process takes place in five steps as follows:

1. The perceptual information such as trajectories of a new subject are retrieved.
2. Using the previously learned scene model, the primitive events for the new video are calculated.
3. By means of retrieved primitive events the discovered activities are calculated.
4. Using the calculated information a test instance HAM (ω^*) is created.

5. The similarity score of the created HAM and the trained HAM models are calculated and the activity with the highest score is selected as the target activity.

All these steps work in an on-line fashion. Once the activity models are trained, to find the activity model that matches with an activity in a test video, we follow Naïve Bayes classification. We decide the final label using the MAP decision rule. The set of generated activity models $\Omega = \{\omega_1, ..., \omega_S\}$ where $S = |\Omega|$. Given the data for an observed test video, $\omega^\star$, we select the activity model, ω_i, that maximizes the likelihood function [Eq. 4.23]:

$$p(\omega^\star|\omega_i) = \frac{p\left(\omega^\star\right) p\left(\omega_i|\omega^\star\right)}{p\left(\omega_i\right)} \tag{4.23}$$

where $p\left(\omega_i|\omega^\star\right)$ denotes the likelihood function defined for activity models $\omega_1, ..., \omega_s$ in model set Ω. We assume that the activity models are independent. Since *a priori* probability of trained models $p\left(\omega_1, ..., \omega_s\right)$ are considered equal (For simplicity, we consider activities independent where the probability of occurring "Prepare coffee" and "Prepare drugbox" as the next activity are equal. However, the frequency of activities are slightly different and in some scenarios there might be dependencies between the activities. Considering these associations will require more complex models and recognition procedure.), we can eliminate $p\left(\omega_i\right)$ and use the following formula [Eq. 4.24]

$$\tilde{p}(\omega^\star|\omega_i) = p\left(\omega^\star\right) \prod_{i=1}^{S} p\left(\omega_i|\omega^\star\right) \tag{4.24}$$

where $p\left(\omega^\star\right)$ is the relative frequency of $\omega^\star$ in the training set. Since the generated models are constructed following a tree structure, the likelihood value should be calculated recursively to cover all nodes of the tree. Therefore, for each model, the recursive probability value is calculated as Eq. 4.25

$$p(\omega_i|\omega^\star) = p(\omega_i^{[l]}|\omega^{\star[l]}) + recur([l]-1) \tag{4.25}$$

$recur$ recursively calculates the probabilities of the nodes in lower levels and stops when there is no more leaf to be compared. Superscripts indexes the levels of the tree ([l]=1,2,3). $p(\omega_i^{[l]}|\omega^{\star[l]})$ calculates probability in the current node given $\omega^\star$ and $p(\omega_i^{[l]}|\omega^{\star[l-1]}$ returns the probability values of this node's child nodes (sub-activities). Given the data for node n of the activity in the test video, $\omega^\star(n) = \{type^\star(n),\ duration^\star(n),\ l^\star(n)\}$, and the activity model i, $\omega_i(n) = \{type^i(n),\ \Delta^i_{duration}(n),\ label^i(n)\}$, where $\Delta^i_{duration} = \{\mu^i, \sigma^i\}$ the likelihood function for node n of the model is defined as Eq.

4.26.

$$\tilde{p}\left(\omega_i(n)^l|\omega^\star(n)\right) = \begin{aligned} & p\left(\omega^\star(n)|type^\star = type^i(n)\right) * \\ & p\left(duration^\star(n)|\Delta^i_{duration}(n)\right) * \\ & p\left(\omega_\star(n)|l^\star = label^i(n)\right) \end{aligned}$$

$p\left(\omega^\star(n)|type^\star = type^i(n)\right)$ checks whether the type of nodes in test tree and trained model are same or not:

$$p\left(\omega^\star(n)|type = type^i(n)\right) = \begin{cases} 1 & if\ type^\star = type^i(n) \\ 0 & otherwise \end{cases} \tag{4.26}$$

$p\left(duration^\star(n)|\Delta^i_{duration}(n)\right)$ measures the difference between activity instance $\omega^\star$'s duration and activity model i bounded between 0 and 1.

$$\begin{aligned} p\left(\omega_\star(n)|\mu = \mu^i_{duration}(n)\right) & \propto \exp^{-Dist_{duration}(n)} \\ \text{where} & \\ Dist_{duration}(n) & = \frac{|duration^\star(n) - \mu^\star_{duration}(n)|}{\sigma^i} \end{aligned} \tag{4.27}$$

$p\left(\omega^\star(n)|l = label^i(n)\right)$ compares the training node and the test node predicted by the *Bayesian Network* classifier.

$$p\left(\omega^\star(n)|l = label^i(n)\right) \begin{cases} 0 & if\ label^\star(n) = label^i(n) \\ 1 & otherwise \end{cases}$$

It should be noted that the $label$ information is only available at root level ($l = 0$) and the recursion stops when it traverses all the leaves (exact inference). Once we have computed $p(\omega^\star|\Omega)$ for all model assignments, using MAP estimation, the activity model i that maximizes the likelihood function $p(\omega_i|\omega_\star)$ votes for the final recognized activity label [Eq.4.28] .

$$\hat{i} = \arg\max_i \tilde{p}\left(\omega^\star|\omega_i\right) \tag{4.28}$$

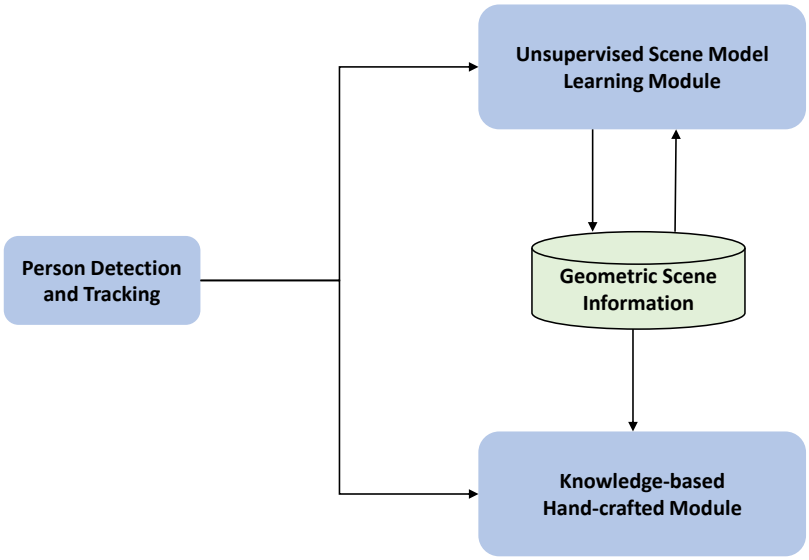


Figure 4.10: The flow diagram of unified method combining trained unsupervised scene models with drawn hand-crafted zones.

4.5 Scene Region Refinement

In this section, we propose a data-driven knowledge-based method for scene region model refinement. The propose model unifies a constraint-based ontology language of event modeling with the unsupervised scene model learning that we explained in the previous section. As Figure 4.10 shows, we detect people in the scene and their calculated trajectories are feed into knowledge-based and unsupervised scene model module. The obtained information are exchanged between the two modules in a loopy way. Here by "knowledge" we refer to the geometric information and scene semantics of the knowledge-based system and also the scene model that unsupervised module learned from the data.

The knowledge-based system that our unsupervised scene model learning is interacting with is a constraint-based ontology system [45] for activity recognition. This system works based on inferring satisfactory values for its defined constraints. An activity model in the knowledge-based method is defined using the prior knowledge about different information in the scene such as objects and scene regions (which are hand drawn in this method) and attributes of mobile objects (such as people) in the scene. The information is dynamically collected by underlying algorithms and a hand-crafted activity model can be created based on them (which is not the topic of this section).

In general all the constraints in this model are defined using its ontology language. A scene model (i) can be defined in this language as follows:

$$\omega_i =< PhysicalObjects, Constraints, Components > \tag{4.29}$$

PhysicalObjects represents a set of objects available in the scene. *Constraints* are a set of conditions or rules that intervene in the modeling of the activities. The *Components* are the set of sub-activities of the main activity. Each one of these model elements has different types. For example the *PhysicalObjects* can be a person or a scene region. The scene regions (or contextual zones in this method) refer to demarcated spatial zones labeled with apriori knowledge (TV zone, coffee zone etc.). It is defined by a human supervisor in the form of a polygon. A constraint is a condition that needs to be satisfy by the physical objects in the scene. A constraint belongs to one of the three possible types: a **Logical** constraint is a Boolean operator such as "Equal" or "Not" (*eg*. "Person is in TV zone" *AND* "Person is sitting"). A **Spatial** constraint defines the spatial relations between the physical objects such as **Person–>Position IN TV Zone**. Finally, a **Temporal** constraint indicates the temporal order between different components of a given activity such as *BEFORE, MEET*. For example "Moving from TV zone to Coffee zone" can be described as "Person IN TV zone" BEFORE "Person in Coffee zone".

In this formalisation, activities according to their complexity are divided into four categories: *PrimitiveState* describes a constant attribute of a physical object for a time interval. For example (e.g. *Person–>Posture* = sitting, *Person–>Position IN TVZone*). *CompositeState* is the combination of two or more primitive states (e.g. "Person stopped" *AND* "Person is standing"). *PrimitiveEvent* shows change of value in an attribute of a physical object (e.g. when person changes posture it can be described as "Person is standing" *BEFORE* "Person is sitting"). *CompositeEvent* shows the combination of two or more activities having a temporal relationship (e.g. "Person is inside TV zone" *MEET* "Person changes posture from standing to sitting"). Figures 4.11 a) and b) show an example of *PrimitiveEvent* and *CompositeEvent*. Figure 4.11(a) presents the Primitive State model of "P_sitting" and "P_insideTVZone". *P_sitting* model checks whether the state of the *Posture* attribute of a *Person* satisfies a target posture value (i.e. sitting). *P_insideTVZone* model is triggered if and only if a *Person* position lies inside a zone with label TV.

Figure 4.11(b) presents the *CompositeEvent* model of "Person watching TV". By using the temporal operator *AND*, this model expresses that its two components ("P inside TV-Zone" and "P_sitting") must be detected at the same time to be a valid targeted activity. The other constraint (i.e. duration) specifies that the first component ("P_insideTVZone")

```
PrimitiveState(P_sitting,
  PhysicalObjects((p1 : Person))
  Constraints ( (p1->Posture = sitting))
)
```

```
PrimitiveState(P_insideTVZone,
  PhysicalObjects((p1 : Person), (z1 : Zone))
  Constraints ((p1->Position in z1->Vertices)
      (z1->Name = zone_TV))
)
```

(a)

```
CompositeEvent(Person_watchingTV,
  PhysicalObjects((p1 : Person), (z1 : Zone))
  Components (
    (c1: PrimitiveState P_insideTVZone(p1,z1) )
    (c2: PrimitiveState P_sitting (p1))
    )
  Constraints ( (duration(c1) > 5)
                  (c1 and c2)
    )
)
```

(b)

Figure 4.11: (a) Primitive State of P_sitting and Person insideTVZone. The variable sitting refers to the desired posture value that defines *sitting* action. (b) Description of the *Composite Event* model Person_watching_TV.

must have been detected for at least 5 seconds.

In the Knowledge-based method created activity models are hand-crafted and tuning the parameters for the attributes are challenging. On the other hand, unsupervised method is efficient but as explained before optimal values for parameter of clustering algorithm should be found such as number of clusters (K) and distance measures. Our method for scene region refinement allows the two methods to interact with each other to find the optimal number of regions. Interaction of knowledge between the two models is conducted in *Zone Matching* step. The learned scene regions are matched with the hand-crafted zones. If the unsupervised module discovers a region that is not defined before in the knowledge-based module, a new contextual zone is generated and it is marked for further labeling by the user.

4.5.1 Zone Matching

We match the scene regions by associating the learned scene model in unsupervised model with the hand-crafted zones in the knowledge-based model. As explained, in the unsupervised model the regions are detected by direct observation of the data. This may cause ambiguities between the learned regions (ψ_i^u) and hand-crafted zones (ψ_j^h). There might be partial overlap between the regions of the unsupervised model and zones of the hand-crafted model. Another case happens where two detected regions lie inside a hand-crafted

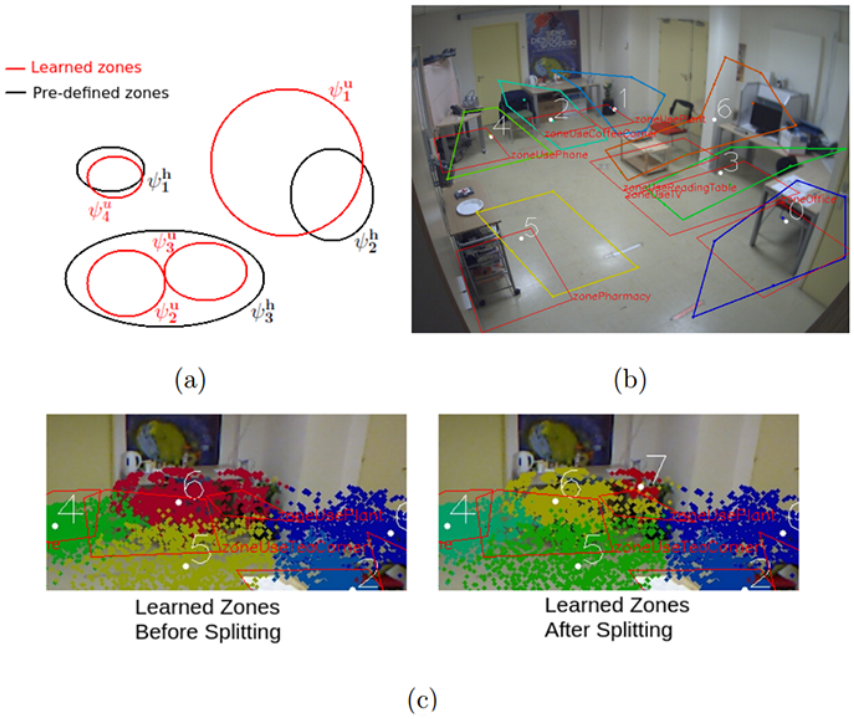


Figure 4.12: (a) Zone matching associates the hand-crafted (pre-defined) zones with learned ones. (b) The hand-crafted zones in knowledge-based module are drawn in red. Learned zones obtained by the unsupervised module for RGB dataset are drawn in green, yellow, orange, blue, and turquoise. (c) The learned zone (#6) that occupies two hand-crafted zones is split into two zones (#6 and #7) in RGBD dataset.

zone (Figure 4.12a). There are also other possibilities. We propose a method that covers all the possible scenarios. If two discovered zones are inside one hand-crafted zone (ψ_2^u, ψ_3^u inside ψ_3^h in Figure 4.12a), they are merged. If there is a partial overlap between zones (ψ_1^u overlaps ψ_2^h in Figure 4.12a), they are split. A new zone is defined for the non-overlapping part of the learned zone (if large enough). As new zones are generated by merging/splitting learned zones, the Primitive Actions and Discovered Activities are modified, thereby new activity models are created accordingly. To measure the overlap between regions, we employ the number of trajectory points that are already clustered during learning of scene regions. For each learned region, we count the number of points that are inside a hand-crafted zone. Then, we associate the hand-crafted zone with the learned region that corresponds to the maximum number of interior points. Algorithm 1 presents the pseudo-code for the zone matching procedure.

An example of learned and hand-crafted zones are given in Figure 4.12b, where we can see that all learned regions are matched with hand-crafted zones. In Figure 4.12c, a learned region that occupies two hand-crafted zones is split into two zones. Figure 4.13a

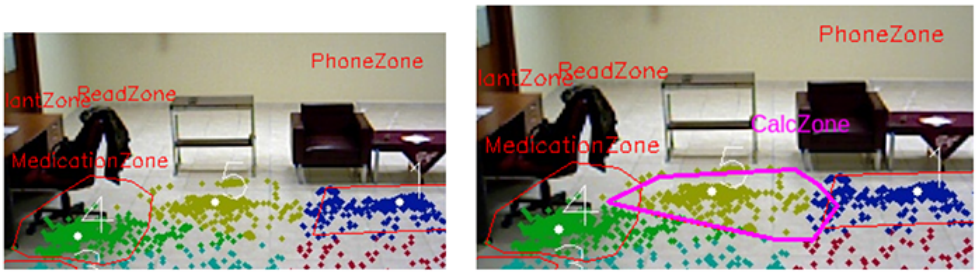
Algorithm 1 Pseudo-code for zone matching procedure.

INPUT: Learned zones ($\{\psi_i^u\}_{i=1:n}$), Trajectory points clustered in z_i ($\{T_i\}_{i=1:N}$), polygons of hand-crafted zones ($\{\psi_j^h\}_{j=1:M}$)
DefinedZAssignedTo$_{1\leq j\leq M} = 0$
LearnedZMatchedWith$_{1\leq i\leq N} = 0$
for *j=1 to M do* **do**
  for *i=1 to N do* **do**
    Count the number of $T_i's$ inside ψ_j^h and assign to *NumInteriorPts*$_j(i)$ *NumInteriorPtsj(i)* $=\| T_i \in \psi_j^h \|_0$
  end
  Find the learned zone that overlaps with ψ_j^h the most: $k = \arg\max_i$ *NumInteriorPtsj(i)*
  if *DefinedZAssignedTo*$_j = 0$ **then**
    if *LearnedZMatchedWith*$_k = 0$ **then**
      ψ_k^u.label $\leftarrow \psi_j^h$.label
      DefinedZAssignedTo$_j = k$
      LearnedZMatchedWith$_k = j$
    else
      Split ψ_k^u into two zones: $\psi_{k1}^u \subseteq \psi^h_{\textit{LearnedZMatchedWith}_k}$ and $\psi_{k2}^u \subseteq \psi_j^h$
      ψ_{k1}^u.label $\leftarrow \psi^h_{\textit{LearnedZMatchedWith}_k}$.label
      ψ_{k2}^u.label $\leftarrow \psi_j^h$.label
      Create two new activity models
    end
  else
    Merge $\psi^u_{\textit{DefinedZAssignedTo}_j}$ and ψ_k^u:ψ^u_{merged}=$\psi^u_{\textit{DefinedZAssignedTo}_j} \cup \psi_k^u$
    ψ^u_{merged}.label $\leftarrow \psi_j^h$.label
    Create a new activity model
  end
end

shows an example where unsupervised module discovers a new zone that is not defined by the user. The system asks the user for the semantic of the new zones, then, new contextual zones and new activity models are automatically generated (Figure 4.13b).

4.6 Conclusion

In this chapter, we propose a hierarchical method to model long-term daily activities. The different steps of the proposed algorithm are explained. First, we describe how the input data are acquired from the scene and then, the methods which we use for feature extraction. The activity models are constructed in a way that any new scene feature can



(a) (b)

Figure 4.13: When a new zone is discovered by the unsupervised module, a new contextual zone and a new activity model are automatically generated. (a) The learned zone (#5) is discovered in GAADRD dataset. (b) Contextual zone and activity models for calculation zone ("CalcZone") is generated.

be appended or removed to the model without interfering with the recognition process. Next, we describe how the important scene regions are detected and then how we used the scene models to detect the primitive events and discover interesting activities in the scene. Once the PEs and DAs are discovered, the collected information is employed to create the hierarchical activity models (HAMs) describing the targeted activities in different scene regions. Finally, the created models are used for recognition when a new video instance needs to be matched with the learned activity models. Additionally, we propose a scene region refinement method that is achieved by combining unsupervised method of generating scene model with an ontology-based hand-crafted method. The discovered scene model by unsupervised method that is learned from the training data compared to the hand-crafted zones marked by user and based on their overlap a split/merge decision is made. In the next chapter, we explain the architecture of both frameworks which are constructed based on the explained method in this chapter. Then, we evaluate the frameworks on the two studied datasets.

Chapter 5

Hybrid and Unsupervised Activity Recognition Frameworks (experimentation and comparisons)

"We can't think fast enough to logically analyze situations, so we rely on our power of pattern recognition."

- Ray Kurzweil

Contents

5.1 Introduction

As discussed in chapter 3, supervised approaches report state-of-the-art results for recognizing short-term actions in manually clipped videos by utilizing fine body motion information. The main downside of these approaches is that they are not applicable in real-world settings. The challenge is different when it comes to unstructured scenes and long-term videos. Unsupervised approaches have been used to model the long-term activities but the main pitfall is their limitation to handle subtle differences between similar activities since they mostly use global motion information [86, 145, 13, 182]. In this chapter, using the model we have described in the previous chapter (chapter 4), we present a hybrid approach (The hybrid framework) for long-term human activity recognition with more precise recognition of activities compared to unsupervised approaches. It enables processing of long-term videos by automatically clipping and performing online recognition. In another variation of this approach (The unsupervised framework), no supervised interference is imposed while learning activity models. Therefore, the framework models long-term human activities without requiring any user interaction. Our goal is to model Activities of Daily Living (ADL) in smart home/hospital settings.

Traditionally, there are two variants of approaches to cope with the challenges in recognizing human activities: *supervised* and *unsupervised* methods. Supervised approaches are suitable for recognizing short-term actions. For training, these approaches require a huge amount of user interaction to obtain well-clipped videos that only include a single action. However, ADLs consist of many simple actions which form a complex activity. Therefore, the representation in supervised approaches are insufficient to model these activities and a training set of clipped videos for ADL cannot cover all the variations. In addition, since these methods require manually clipped videos, they can mostly follow an offline recognition scheme. Analyzing long-term activities has many application areas in surveillance, smart environments, etc. Especially monitoring activities of daily living (ADL) is one of the application areas that has been investigated by researchers in recent years. ADL, such as cooking, consist of long-term complex activities that maybe composed of many short-term actions. As people perform daily activities in different ways, there is a big variation for the same type of activities and it is a very challenging problem to

recognize ADL.

On the other hand, unsupervised approaches are strong in finding spatiotemporal patterns of motion. However, the global motion patterns are not enough to obtain a precise classification of ADL. For long-term activities, there are many unsupervised approaches that model global motion patterns and detect abnormal events by finding the trajectories that do not fit in the pattern [154, 69]. Many methods have been applied to traffic surveillance videos to learn the regular traffic dynamics (e.g. cars passing a crossroad) and detect abnormal patterns (e.g. a pedestrian crossing the road) [87]. However, modeling the global motion pattern cannot capture the complex structure of long-term human activities.

In this chapter, we describe two methods to exploit the benefits of supervised/unsupervised approaches to model and evaluate daily living activities. These methods provide a comprehensive representation of activities by modeling both global and body motion of people. With limited user interaction in hybrid and without interaction in the unsupervised framework (*eg.* , clipping long-term videos into short-term actions, labeling huge amount of short-term actions as in supervised approaches) our framework recognizes more precise activities compared to available approaches. We use the term *precise* to indicate that unlike most of the trajectory-based approaches which cannot distinguish between activities under the same region, our approach can be more sensitive in the detection of activities thanks to local motion patterns. Since the videos are automatically clipped, our frameworks perform online recognition of activities.

5.2 Activity Recognition Frameworks

Having explained the modeling steps in the previous chapter (4), next, we explain different parts of the two frameworks and the data-flow of training and testing phases in more details.

5.3 The hybrid framework

Figure 5.1 illustrates the flow of the training and testing phases in the proposed weakly supervised framework. For the training phase, the algorithm learns relevant zones in the scene and generates activity models for each zone by complementing the models with information such as duration distribution and Fisher Vector (FV) representations of discovered activities. At testing, the algorithm compares the test instances with the generated activity models and infers the most similar model.

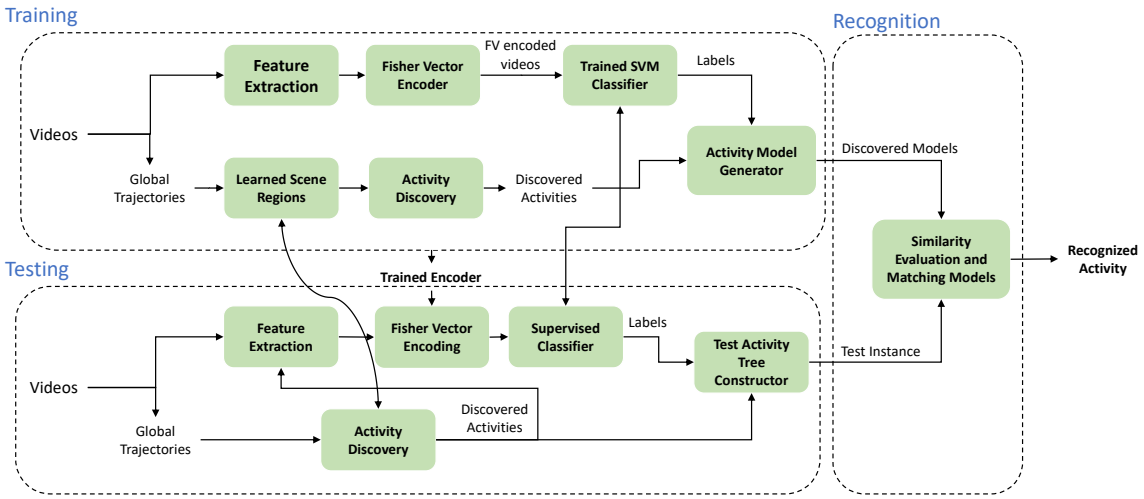


Figure 5.1: Architecture of the hybrid framework: Training and Testing phases

5.3.1 Learning Scene Regions

The regions of interest in an activity recognition scenario are those parts of the scene where there is a higher probability of the recurrence of certain motion patterns (e.g. around phone desk, coffee machine area etc.). Thus, finding these regions helps to discover and localize activities occurring in the scene. In [62], in order to find important scene regions, the authors assume stopping points of trajectories as important regions where activities are happening. They put a threshold on instant speed of the people and based on that, they find out when the people stop moving. By clustering stopping points they learn important regions. However, they use 2D trajectory points that are extracted from averaging optical flow values. Noisy optical flow results in inaccurate stopping point calculation and hence, inaccurate scene regions. To avoid such problems, we use a 3D tracker to extract 3D trajectories which less prone to noise.

As explained in chapter 4, we track people throughout the scene to extract their 3D trajectories. We find dense scene regions by clustering trajectory points corresponding to people's locations on the ground using the *K-means* clustering algorithm. The number of clusters determines the granularity of the regions. A lower number for clustering creates wider regions. Generally, activities occur inside each one of these regions; however, one activity could occur in two consecutive regions and two distinct activities could happen in the same region. An example of scene regions is illustrated in Figure 5.2. We define a scene model with three levels of scene regions: coarse, medium and fine granularity clusters. The trajectory points are calculated from 15 subjects in the training

Figure 5.2: A sample of scene regions clustered using trajectory information (image from the CHU dataset)

set performing the assigned activities. The detected scene regions (in coarse level) shows that using scene topology, semantic explanation of most of the computed regions is possible.

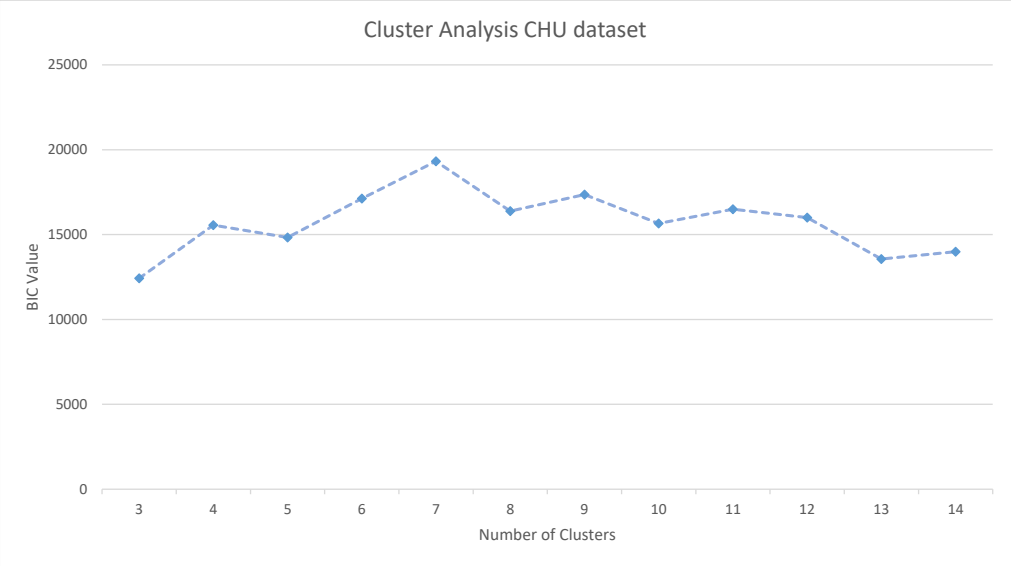
Therefore, a scene region in a given level includes several regions at a finer level. This helps to locate sub-activities that are limited to sub-regions of a bigger scene region. Labels of the regions are randomly assigned as an integer. We use arithmetic mean to calculate cluster centers and City-block distance as distance measure in K-means algorithm. Clustering runs for n = 300 passes each time starting from different random center assignments and solution with the lowest within-cluster sum of distances is chosen.

5.3.2 Cluster Analysis

When we use K-means algorithm we need to supply the algorithm with parameter K which is the number of resulted clusters. As it is explained in chapter 4, we use Bayesian Information Criteria (BIC) to investigate the best range of k which is used in K-means clustering. We use BIC to decide if a cluster is in its ideal form or it is better to split it to two or more clusters. The results of applying BIC formula on the resulted cluster by different values of K in $K - means$ is shown in Tables 5.1 and 5.2. The table 5.1 illustrates the BIC values for the CHU dataset. There is a visible increasing trend in the values of BIC starting from $K = 3$. The BIC value reaches to its maximum at $K = 7$ in the investigated interval. There are 6 activity labels in the CHU dataset, showing the best

BIC value almost coincides with this value. There are some regions in this dataset where the subjects stand in those regions and read the instruction of activities. Such regions add redundant trajectory points into the clustered trajectory set. Table 5.2 shows the BIC values for the GAADRD dataset. The number of activities labeled in the ground-truth for this dataset is 7. The best BIC value is obtained when the K is set to 7 for the $K-means$ algorithm. In this dataset, there is a desk where subjects usually stand by it to take paper in fill it as part of the "Establish Account Balance" activity. Looking at the trajectory data, this region looks similar to an important scene region. That may be the reason we get near optimal BIC value for the K value set to 8.

Table 5.1: Results of cluster analysis by applying Bayesian criteria on cluster points on CHU Nice dataset. The plot shows BIC values w.r.t. the number of clusters.



	Number of Clusters											
	3	4	5	6	7	8	9	10	11	12	13	14
BIC Value	12425	15554	14835	17124	19317	16378	17364	15662	16492	15997	13559	13996

Table 5.2: Results of cluster analysis by applying Bayesian criteria on cluster points on GAADRD dataset. The plot shows BIC values w.r.t. the number of clusters.

	Number of Clusters											
	3	4	5	6	7	8	9	10	11	12	13	14
BIC Value	4450	4945	4985	5104	5451	5321	4122	3893	3862	3520	3221	3114

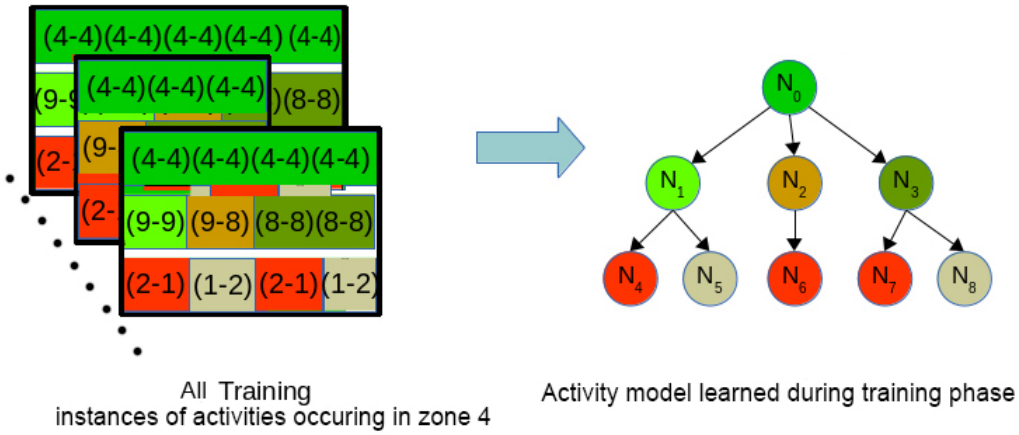


Figure 5.3: Illustrates conversion of a video into sequence of Primitive events and Discovered activities and construction of tree structure of activity models.

The cluster analysis results show that a K value close to the number of available activity labels in the ground-truth is optimal for BIC criteria. We use this detected optimal value as a hint to detect scene regions. However, performing the clustering at three levels of granularity, ensures that any interesting region and sub-region is detected and considered in the modeling of activities. In the experiments, we choose K for each one of the datasets based on the performed clustering analysis.

5.3.3 Primitive Events

As explained in chapter 4, we use trajectory points and learned scene regions to decompose activities into sub-parts. We use the notion of primitives to characterize the movement of people inside the scene. Given the set of scene regions, we assign a region code to each trajectory point. This mapping transforms 3D points into a sequence of scene region labels. If the labels of two adjacent trajectory points are the same, it means that both points stay at the same region (*Primitive Event* type *Stay*); otherwise, there is a transition from one region to the other (*Primitive Event* type *Change*). This mechanism helps to divide the whole video sequence into a sequence of primitives events (Figure 5.3). Semantic labeling is performed independently for the l=*3* region levels.

5.3.4 Discovered Activities

We defined a *Discovered Activity* as a combination of primitive events at a coarser level. It describes the body motion, pose or appearance of a person through motion, geometrical and appearance descriptors and contains its spatial (region information) and temporal (time interval and duration) information.

5.3.5 Feature Extraction

All of the feature types extracted for the supervised framework (Chapter 3 section 4.3) are extracted for all instances of Discovered Activities given their detected intervals. This includes two different types of image descriptors (hand-crafted and deep) and skeleton descriptors: Geometrical pose features (Angle and distance features), Local descriptors using improved dense trajectories (HOG, HOF, MBHx and MBHy descriptors) and TDD deep features (Spatial and temporal).

5.3.6 Activity Models

Using all of the pieces of information stored in the *Discovered Activities* of the same region in the training set, we construct an *activity model*. For the temporal aspect of the model, we compute the probability distribution function (PDF) of the time duration of the activity. For each type of $Stay/Change$ primitive, we record the duration values and learn the underlying distribution functions of the time duration of the *Discovered Activity*. In addition to the extracted descriptors of DAs, we keep region and sub-region information as the spatial component of the model. We define a model of activities as a tree structure where each node has collective information of primitives and *Discovered Activities*. Figure 5.3 illustrates the construction of an activity model based on training instances of discovered activities occurring in scene region 4. As explained in chapter 4, for every node in the model there are attributes characterizing the *Discovered Activities* and their primitive information, namely: *Type*, *Duration*, *Label* and *Sub-activity*.

5.3.7 Generating and Recognition of Activity Models

During training, using scene region information, we calculate primitive events and discovered activities; their descriptors are then extracted and Fisher Vector representations are built. Then, for each activity, we construct an activity model as explained in 5.3.6. To obtain the label of the model we train a supervised classifier. For representation of the extracted descriptors of *Discovered Activities*, we calculate their Fisher Vectors (Figure 5.4). We store the predicted labels in the root node of the models. During testing, for a new unknown video, we create the activity trees in online mode following the same steps we have performed for training *Discovered Activities* models. We find the most similar generated activity model of the training phase to this test instance tree. For model matching during the recognition, we follow the Bayesian approach explained in section 4.4.11.

5.3.8 Experiments

In this section, we report the performance of the proposed framework on both datasets:

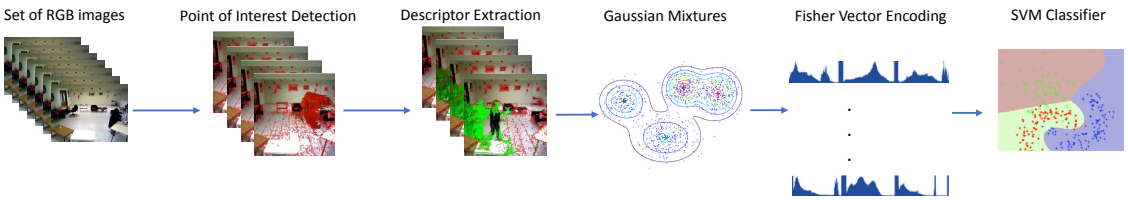


Figure 5.4: The pipeline for training the supervised classifiers to predict labels of activities represented with Fisher Vector.

- GAADRD
- CHU Nice Hospital

The evaluation of our activity recognition framework reflects the accuracy of the activity discovery procedure. As mentioned in chapter 3, we use a test protocol which has been used in other works on these datasets. In both datasets, 3/5 of the videos in these datasets are used for training and the remaining 2/5 of the videos are used for testing. As the training and testing steps of the framework explained in the previous sections, the evaluation of the framework starts with learning activity regions in the scene and continues with modeling the targeted activities on the training videos. The evaluation finishes with the recognition step on the unseen test videos. The explained procedure is illustrated in figure 5.5 and can be described in three stages: In the **training process**, the scene models are learned and the activity models are trained from the instances of the training subjects. **The recognition** is performed on new unseen videos of the test subjects. Given the trained activity models and the scene model, the recognition procedure returns a set of time intervals indicating the delineation of the located target activities. The **evaluation** of the recognition is done by using manually annotated ground-truth information. The provided ground-truth for each video comprises of start and end time and label of activities in the video. The detected intervals are compared against the ground-truth intervals and an overlap higher than 80% of the ground-truth interval is considered as a True Positive detection of that activity (Figure 5.5). The figure illustrates how the evaluation is done, however, to measure the accuracy of the recognition we formally define:

$$TP = \#\{TruePositive_i\} \tag{5.1}$$

Which is the total number of activities of type i that are correctly classified.
Similarly, we define False Positive as the number of activities of type i that is recognized

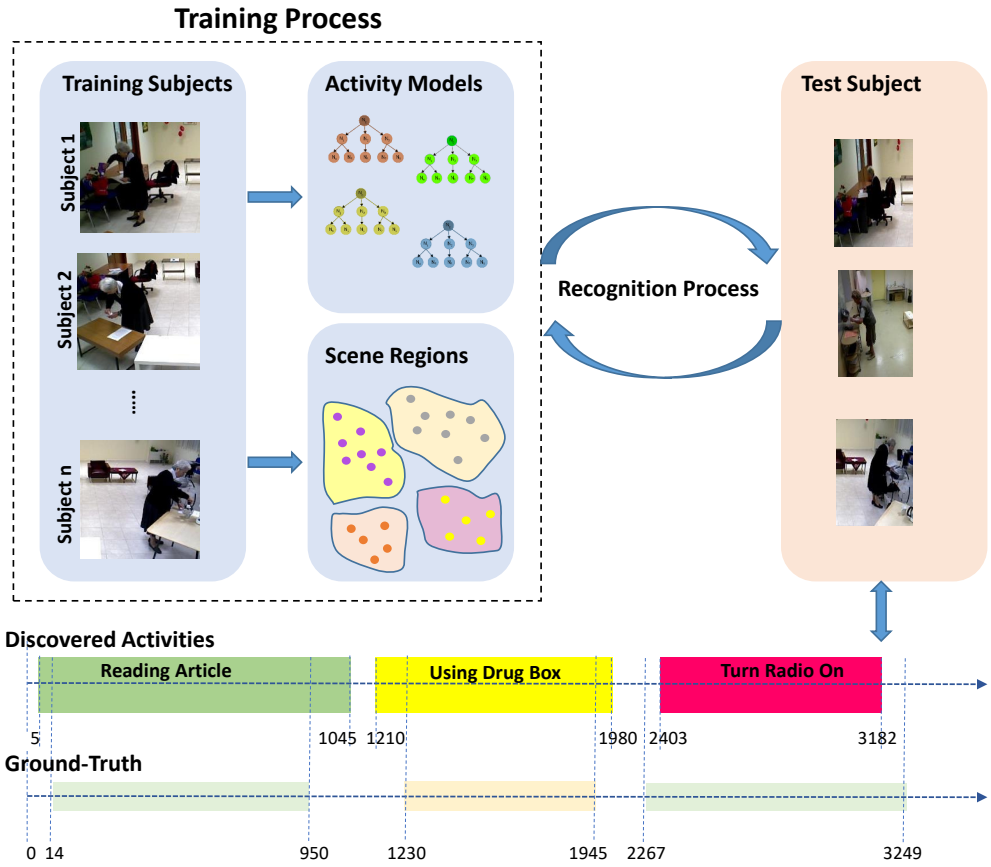


Figure 5.5: Illustrates the process of activity recognition in our framework which is divided into three steps of: Training, Recognition and Evaluation. As a result of this process, an input video is segmented into detected activity intervals with assigned labels to multiple recognized activities.

by the framework but such activity is not annotated in the ground-truth:

$$FP = \#\{FalsePositive_i\} \tag{5.2}$$

False Negatives are the instances that are annotated in the ground-truth but not recognized by the framework:

$$FN = \#\{FalseNegative_i\} \tag{5.3}$$

These measures supply hit or miss of the instances in the videos and consequently, provide an intuitive insight for evaluating the recognition procedure of the system.
Using these defined metrics we define *Precision* and *Recall* metrics similar to the one we have defined for the supervised framework.

True Positive Rate (TPR) or recall is the proportion of actual positives which are identified correctly:

$$TPR = \frac{TP}{TP + FN} \tag{5.4}$$

The higher the value of this metric the better is the performance. Similarly, Positive Predictive Value (PPV) or precision is defined as:

$$PPV = \frac{TP}{TP + FP} \tag{5.5}$$

Higher value of this metric also indicates better performance of the recognition system.

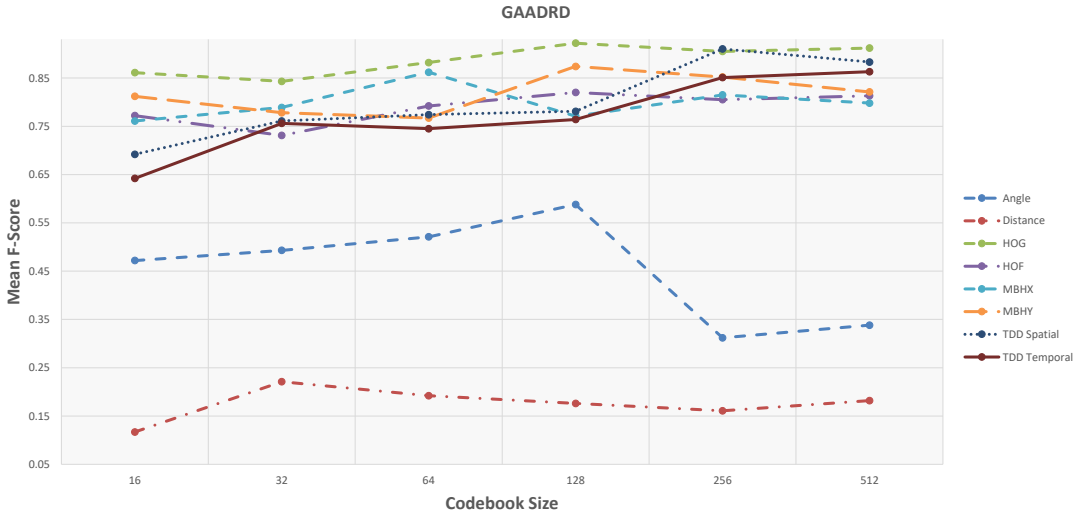
5.3.8.1 GAADRD Dataset

The details about this dataset are explained in section 3.7.1 of chapter 3. As explained in chapter 3 we are using Fisher Vector encoding for the supervised classifier. We have tried the SVM classifier with different parameters and codebook sizes for the FV dictionaries (16, 32, 64, 128, 256, and 512). Here we report the results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset. Table 5.3 shows overall accuracy with Precision and Recall metrics using different feature types to predict labels by the supervised classifier. The plot on top illustrate F-Score information of the table. Based on the obtained results we can conclude that:

- Compared to the supervised-only method, the hybrid method performs better as it takes benefits from global information in addition to the supervised cues from the classifier.
- Similar to the supervised framework, for most of the features, medium size codebook (128) works best for this dataset. This might be because of the medium size of the dataset. For most of the features, accuracy grows with increase in size of the codebook and then there is a drop in accuracy when the codebook size continue to grow.
- Appearance features (e.g. HOG) perform better than motion features. Also among deep features appearance features performs better than temporal features.
- Similar to supervised framework, geometrical features have poor performance. However, unsupervised information of models help to slightly improve their performance. Angle feature achieves better accuracy since angular posture features are more informative in daily activities. Distance feature performs the worst among all features since the distance features in daily activities is not a discriminative feature.

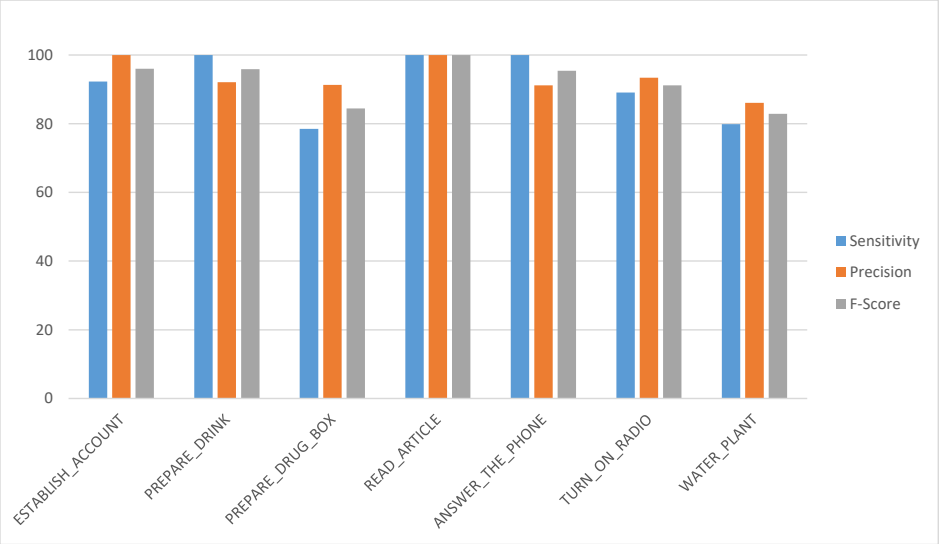
- Deep spatial TDD features along with HOG feature achieve the best results. This high performance might be because of capturing contextual information by the appearance features.
- With a deeper analysis of the performances, we find out that this framework's worst performance is on "Watering Plant" activity. The poor performance on this activity is due to lack of sufficient information that can be extracted from few frames that this activity contains.
- Performance of all feature types improved with hybrid framework except MBHX descriptor. Adding the supervised information from this descriptor to the unsupervised models does not improve the accuracy of the recognition. The performance drops with 0.01 in F-Score (-1%). This happens usually because of the conflict in scoring process between the supervised label information and the similarity score coming from the unsupervised HAM models
- MBHY outperforms significantly the models with geometrical features and performs comparable to the other hand-crafted and deep features. This descriptor relies on motion information and performs poorly on activities that there is a lack of motion.
- Further analysis shows that activities with similar motion patterns and similar duration are mostly confused with each other.
- In this dataset, TDD spatial outperforms TDD temporal feature showing that appearance information is more important in long-term activities where no t much motion information is available.

Table 5.3: Results of using the hybrid framework with different feature types on GAADRD dataset. The plot shows F-Score values w.r.t. codebook size.



	16			32			64			128			256			512		
	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score
Angle	60.2	39.5	0.47	55.1	44.2	0.49	55.8	50.3	0.52	63.4	53.7	0.58	44.2	24.7	0.31	37.8	29.8	0.33
Distance	12.1	10.7	0.11	20.5	24.7	0.22	21.1	18.2	0.19	16.2	18.7	0.17	14.1	20.6	0.16	17.3	18.8	0.18
HOG	88.6	84.2	0.86	82.7	87.2	0.84	87.5	88.7	0.88	94.1	90.1	0.92	87.4	92.9	0.90	89.2	93.4	0.91
HOF	75.9	79.1	0.77	75.9	72.1	0.73	78.3	79.9	0.79	84.2	80.4	0.82	78.1	82.4	0.80	84.7	78.5	0.81
MBHX	74.4	79.1	0.76	77.4	79.2	0.78	87.6	86.4	0.86	78.7	76.5	0.77	79.9	83	0.81	82.4	77.4	0.79
MBHY	82.1	81.9	0.81	78.2	77.8	0.77	78.2	75.2	0.76	88.7	86.2	0.87	86.4	84.7	0.85	85.1	80.6	0.82
TDD Spatial	76.4	64.2	0.69	77.1	75.3	0.76	80.2	75.9	0.77	81.1	76.5	0.78	93.7	89.2	0.91	89.7	87.9	0.88
TDD Temporal	67.1	62.4	0.64	81.2	70.9	0.75	78.8	71.2	0.74	75.2	77.1	0.76	88.1	83.3	0.85	87.4	86.1	0.86

Figure 5.6 show more the details about the HOG feature that achieves the best result on GAADRD dataset. Detailed analysis of all descriptors are provided in the appendix A.



	HOG		
	Precision [%]	Recall [%]	F-Score
Establish Account	100	92.3	0.95
Prepare Drink	92.1	100	0.95
Prepare DrugBox	91.3	78.5	0.84
Read Article	100	100	1
Answer the Phone	91.2	100	0.95
Turn On Radio	93.4	89.1	0.91
Watering Plant	86.1	79.9	0.82
Average	93.44	91.40	0.92

Figure 5.6: Results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset using the HOG descriptor for the supervised classifier. Combining HOG descriptor with the unsupervised models achieves the best performance for this dataset (0.92) where the hybrid method outperforms the supervised method by 3% margin in F-Score. HOG is an strong appearance feature and when combined with the global trajectory information it achieves a superior performance. Since the activities are performed in different scene regions, background information of different zone can get encoded in the features and improve the performance. Among the classes of activities, the best performance belongs to "Read Article" activity with 1.00 which means all activities of this type is detected and recognized correctly. The worst performance belongs to the "Watering Plant" activity with 0.82 of the F-Score. Global information of the hybrid models compensates the lack of supervised descriptors and helps to achieve this high performance. However, detection and recognition of very short activities are still challenging.

5.3.8.2 CHU Dataset

The details about this dataset are explained in section 3.7.2 of chapter 3. As explained in chapter 3 the Fisher Vector encoding is used for the supervised classifier where, we have tried different parameters for the SVM classifier and different codebook size for the FV dictionaries (16, 32, 64, 128, 256, and 512). Next, the results of the hybrid framework on the CHU Nice Hospital dataset are reported. Table 5.4 shows overall accuracy with Precision and Recall metrics using different feature types to predict labels by the supervised classifier. The plots on top illustrate the best F-Score results of each feature type. The results show that:

- Similar to GAADRD, the hybrid method performs better than the supervised classifier using most of the feature types as it utilizes more information.

- Unlike the other dataset, the performance of the framework is not largely affected by the size of the codebook. In most of the feature types there is a trend of increasing accuracy by increased codebook size, however, the fluctuations of the curves are not significant. The most visible variation is of the Angle feature.

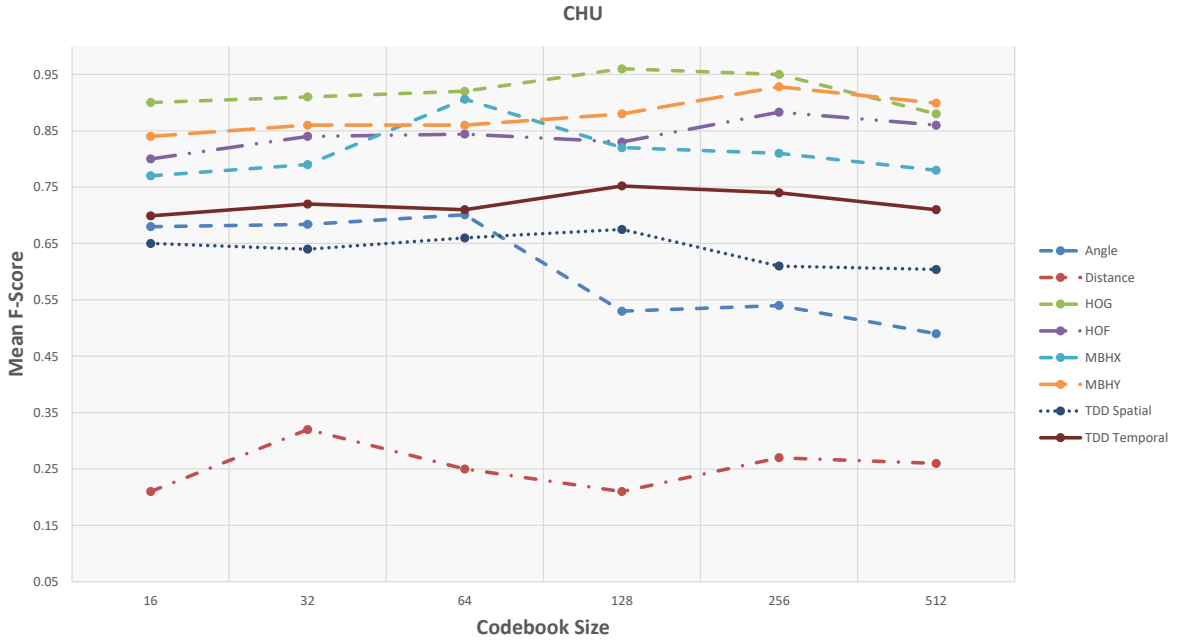
- Appearance HOG feature achieves the best performance where the codebook size 128 maximizes its performance.

- TDD spatial performs poorly on the CHU dataset. The performance of the supervised classifier is not high using this feature (0.65 F-Score) which causes a lower rate in the recognition task. The hierarchical models try to compensate the loss in accuracy but it does not boost the performance significantly (0.67 in F-Score). Tdd temporal outperforms the TDD Spatial deep feature on this dataset. Although the ADL types are similar in both datasets, appearance features in one and motion features in other perform the best. This reveals that the performance of these features are dataset dependant. In CHU dataset the temporal duration of the activities are longer than the GAADRD dataset. It is where the temporality gains more importance and help the models to obtain enough information about each activity. It might be the reason that TDD temporal outperforms TDD spatial on CHU.

- Further analysis show that the best performances on activities of this dataset are achieved on the "Prepare DrugBox" activity which is performed clearly in front of the camera from a side view (Hand motions are clearly visible, hence appearance and motion features can capture most of the details). The lowest rate of recognition is on the "Watering Plant" activity which is performed far from the camera with high speed and back of the subject to the camera where the discriminative information

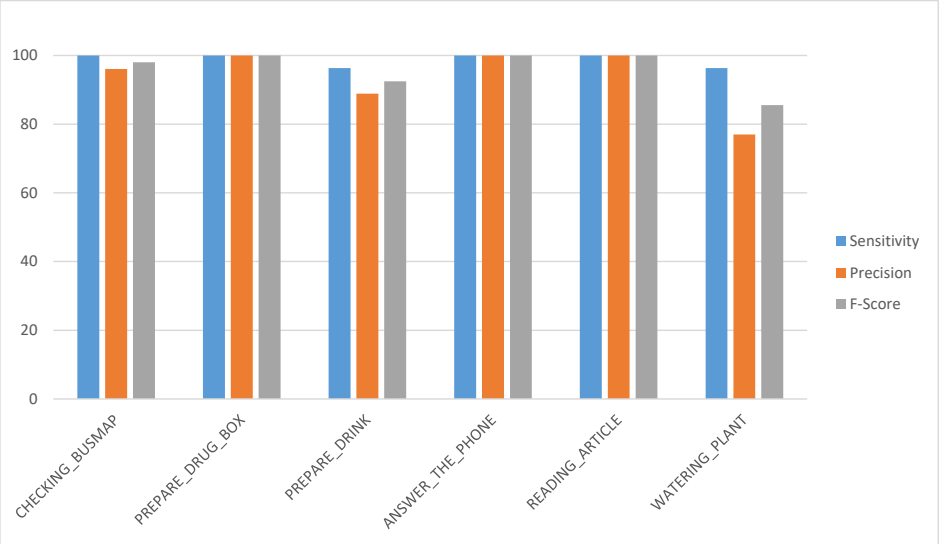
can not be captured (descriptor-wise detailed analysis of activities are provided in the appendix A).

Table 5.4: Results of using the hybrid framework with different feature types on CHU dataset. The plot shows F-Score values w.r.t. codebook size.



	16			32			64			128			256			512		
	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score
Angle	72.1	65.8	0.68	67.8	69.2	0.68	76.1	65.6	0.70	56.8	51.2	0.53	57.1	52.1	0.54	53.7	46.6	0.49
Distance	24.3	19.2	0.21	31.4	33.1	0.32	22.8	28.2	0.25	19.5	25.2	0.21	24.9	29.7	0.27	25.1	28.2	0.26
HOG	92.4	88.1	0.90	90.2	92.3	0.91	93.2	90.9	0.92	96.7	95.4	0.96	96.2	93.9	0.95	90.3	86.2	0.88
HOF	78.3	82.1	0.80	85.7	84.1	0.84	82.6	85.7	0.84	81.3	86.2	0.83	86.5	89.6	0.88	86.9	87.1	0.86
MBHX	76.8	77.3	0.77	72.9	86.7	0.79	87.5	92.8	0.90	80.5	85.6	0.82	78.6	84.2	0.81	77.13	80.9	0.78
MBHY	84.7	83.5	0.84	85.4	86.9	0.86	88.9	83.7	0.86	87.2	90.4	0.88	90.3	93.8	0.92	88.9	90.7	0.89
TDD Spatial	71.1	61.3	0.65	70.2	60.5	0.64	63.1	69.4	0.66	68.2	66.4	0.67	60.3	62.1	0.61	58.2	62.4	0.60
TDD Temporal	71.2	67.5	0.69	76.9	68.4	0.72	74.2	69.6	0.71	77.4	73.8	0.75	75.9	72.8	0.74	72.4	70.9	0.71

Figure 5.7 show more the details about the HOG feature that achieves the best result on CHU dataset. Detailed analysis of all descriptors are provided in the appendix A.



	HOG		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	96.1	100	0.98
Prepare DrugBox	100	100	1.00
Prepare Drink	88.9	96.3	0.92
Answer the Phone	100	100	1.00
Reading Article	100	100	1.00
Watering Plant	77.0	96.3	0.85
Average	93.67	98.77	0.96

Figure 5.7: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the HOG feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. The hybrid framework achieves the best performance on the CHU dataset using this feature. Previously, this feature achieved the best results using the supervised classifier. This good performance of the supervised information is reverberated on the hybrid framework achieving the best performance on this dataset compared to the other features in both supervised and hybrid settings. Three out of six activities are recognized with a hundred percent accuracy. These activities include "Prepare DrugBox", "Answer the Phone", and "Reading Article" activities. The worst performance is again obtained on the "Watering Plant" activity with a F-Score of 0.85.

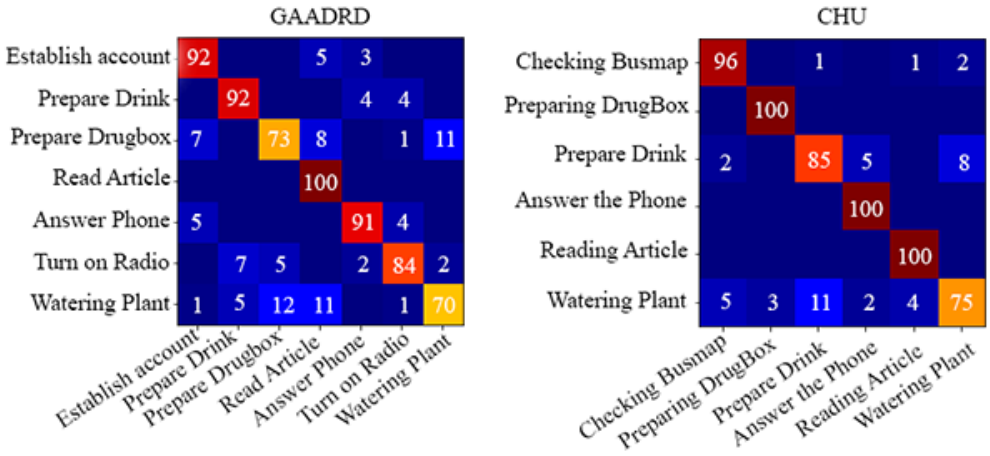


Figure 5.8: Confusion matrices of the best configuration of hybrid framework on GAADRD and CHU datasets (with HOG descriptor). The values show mean accuracy (%).

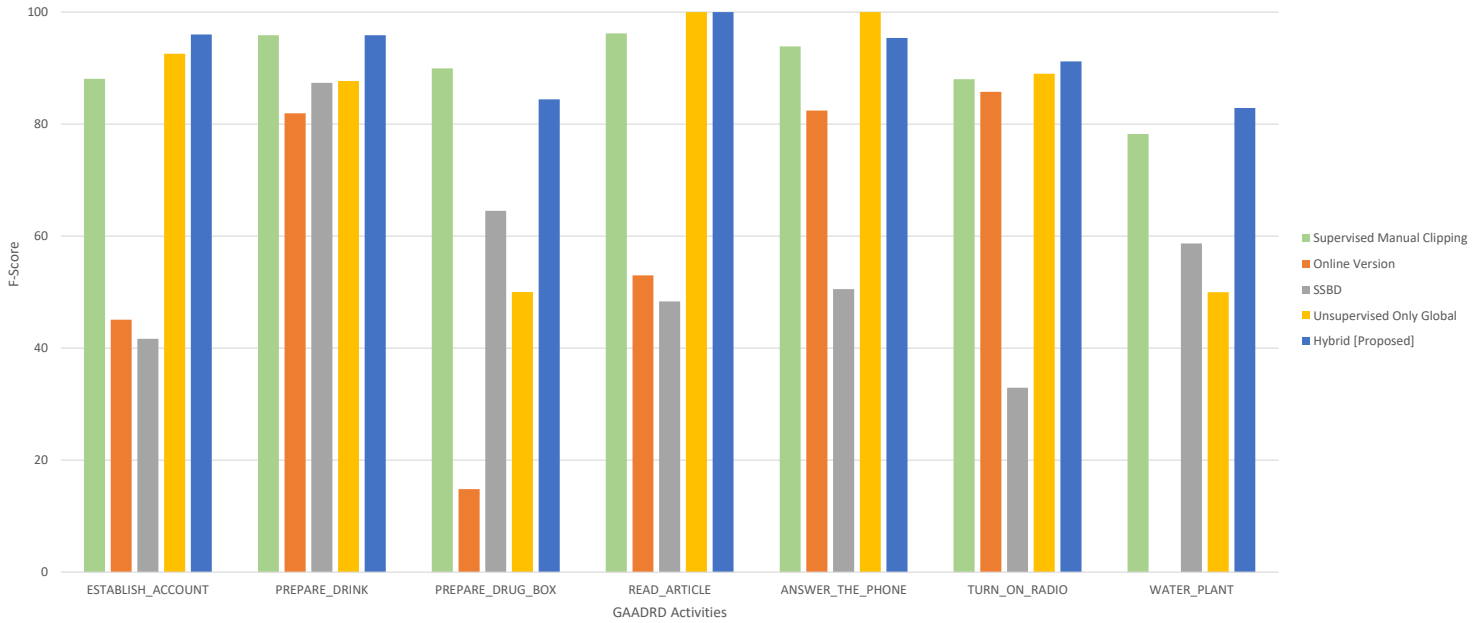
5.3.9 Discussion and Comparison

In both evaluated datasets, the recognition results demonstrate high rate of true positives (TP) and low rate of false positives (FP). This results in high recognition rate which is reflected in the rate of F-Score metrics for each table. These evaluation demonstrates that the developed framework is capable to accurately recognize most of the targeted activities with low error-rate. The activity models help to understand the reason behind the occurrence of false positives and false negatives. As it can be seen from confusion matrices in figure 5.8, most of the failures are because of the similarity between the motion pattern of the subjects doing an activity which makes the supervised classifier confused. This results in wrong label embedded to the test activities HAM. High accuracy in the supervised classification coincides with the higher recognition rate with hybrid models. Among the HAM models, the models integrated with HOG and MBHY descriptors achieve the best results. For the GAADRD dataset TDD Spatial deep features also achieve high accuracy. The reason for having false positives in the recognition is related to the activities which subject finishes one activity but stops at that region. For example, the subject stays for a while at the "Coffee" region after "Prepare Drink". This waiting is not labeled as "Prepare Drink" in the ground-truth, however, can be considered as an instance of this activity by the models. It might be possible to resolve this problem by having refined topologies.

We have compared our method with the results of three other approaches evaluated on these datasets. We compare with the supervised approach in [233] where videos are manually clipped. We did also a comparison with an online supervised approach that

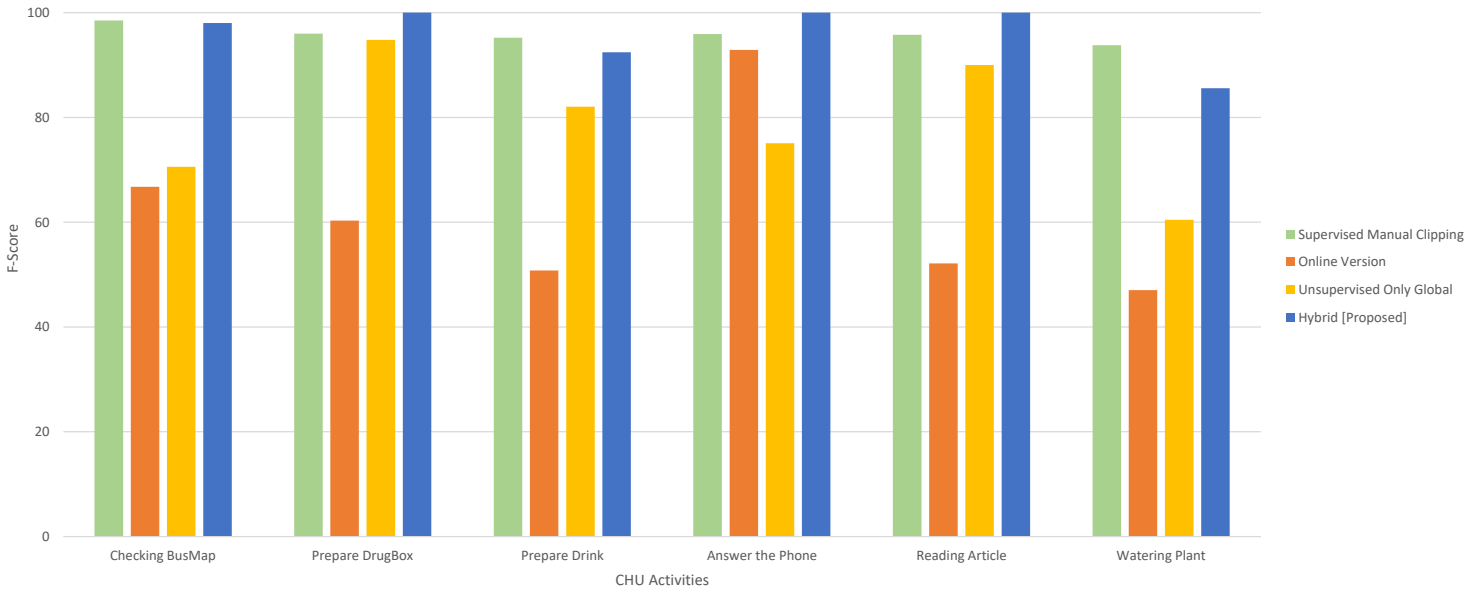
follows [233]. We compare the activity models with another version of the models [62] that no supervised label is embedded (In this version, only score of the label attribute is omitted and not considered in the final score). We additionally compare GAADRD with the produced results of another study in [8].

Table 5.5: Comparison of different recognition frameworks with ours on the GAADRD dataset. The methods are differentiated by using different color codes. The diagram shows class-wise accuracy of each method based on F-Score metric. The table in below shows the detailed results of each method with respect to each class in the dataset. The best results in each section is indicated in bold.



	Supervised (Manual Clipping) [233] with HOG, Dict sz=512			Online Version of [233]			Classification by Detection SSBD [8]			Unsupervised Using Only Global Motion [62]			Hybrid (Proposed Method)		
	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score
Establish Account	92.2	84.3	0.88	29.1	**100**	0.45	41.67	41.67	0.41	86.2	**100**	0.92	92.3	**100**	0.95
Prepare Drink	92.1	**100**	0.95	69.4	**100**	0.81	80.0	96.2	0.87	**100**	78.1	0.87	**100**	92.1	0.95
Prepare DrugBox	94.9	85.5	0.89	20.2	11.7	0.14	51.28	86.96	0.64	**100**	33.34	0.50	78.5	91.3	0.84
Reading Article	96.2	96.2	0.96	37.8	88.6	0.52	31.88	**100**	0.48	**100**	**100**	1.0	**100**	**100**	1.0
Answer the Phone	88.5	**100**	0.93	70.1	**100**	0.82	34.29	96.0	0.50	**100**	**100**	1.0	**100**	91.2	0.95
Turn On Radio	89.4	86.7	0.88	75.1	**100**	0.85	19.86	96.55	0.32	89.0	89.0	0.89	89.1	93.4	0.91
Watering Plant	**84.8**	72.6	0.78	0	0	0	44.45	**86.36**	0.58	57.1	44.45	0.49	79.9	86.1	0.82
Average	91.16	89.33	0.90	43.1	71.4	0.51	43.34	86.24	0.54	90.32	77.84	0.81	**91.4**	**93.44**	**0.92**

Table 5.6: Comparison of different recognition frameworks with ours on the CHU dataset. The methods are differentiated by using different color codes. The diagram shows class-wise accuracy of each method based on F-Score metric. The table in below shows the detailed results of each method with respect to each class in the dataset. The best results in each section is indicated in bold.



	Supervised (Manual Clipping) [233] with HOG, Dict sz=256			Online Version of [233]			Unsupervised Using Only Global Motion [62]			Hybrid (Proposed Method)		
	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score
Checking BusMap	**100**	97.1	0.98	50.1	**100**	0.66	54.54	**100**	0.70	96.1	**100**	0.98
Prepare DrugBox	**100**	92.3	0.95	43.2	**100**	0.60	**100**	90.1	0.94	**100**	**100**	**1.0**
Prepare Drink	**93.1**	**97.4**	0.95	38.1	76.1	0.50	80.0	84.21	0.82	88.9	96.3	0.92
Answer the Phone	92.2	**100**	0.95	86.7	**100**	0.92	60.1	**100**	0.75	**100**	**100**	**1.0**
Reading Article	97.5	94.1	0.95	36.4	92.0	0.52	**100**	81.82	0.90	**100**	**100**	**1.0**
Watering Plant	**100**	88.3	0.93	33.9	76.9	0.47	53.9	68.9	0.60	77.0	96.3	0.85
Average	**97.13**	94.87	0.95	48.06	90.83	0.61	74.75	87.50	0.78	93.66	**98.76**	**0.96**

In the online approach (sliding window), a SVM classifier is trained using the action descriptor histograms. For training this classifier the descriptors are extracted using intervals obtained from ground-truth. In online testing, actions are localized with sliding window of size: 10, 12, 18, 24 frames. We slide the window with step of 1 frame. Since we use more than one windows size we employ non-maximum suppression algorithm [172] to select final temporal location of the action. The results with the best parameters of the classifier are reported.

Tables 5.5 and 5.6 show the comparison of different methods on GAADRD and CHU Nice Hospital datasets respectively. Our approach always performs equally or better than online supervised approach in [233]. Our approach outperforms totally supervised version of [233](off-line manually clipped) in overall accuracy and also in most of the class-wise accuracy rates. This reveals the effectiveness of our hybrid technique where combining information coming from both constituents could contribute to enhance the recognition. Our recognition mechanism helps each element to correct the others, i.e. if the classifier predicts a wrong label for a test instance, duration score or scores from sub-activities could be more informative and then turn over the final decision. E.g. if a subject performs an action in "Phone Region" that the action's motion pattern is similar to "Prepare Coffee" it obtains a high score by the motion descriptors, however, it gets very low score in terms of scene region and sub-activity scores. This results in low final score for the test instance to be matched with "Prepare Coffee" activity.High score of scene region, time duration and sub-activities help the framework to figure out the correct model. The most similar approach to ours [62] does not use local motion information in their models. Using models that represent both global and local motion enable to distinguish activities occurring inside the same region, thereby it reduces false alarms compared to the models using only global motion. We have increased the average recall and precision rates in most of the activities. Since the motion representation of models in [62] contains only global information, it fails to distinguish activities inside the zones, e.g., passing by the phone zone and answering phone in the phone zone could be considered as the same activity (since it does not utilize local descriptors). Hence, [62] results in high false positive rates. In addition, we can observe that the proposed approach improves the true positive rates and increases sensitivity rates using various descriptors in the models.

In CHU dataset, since people tend to perform the same activities in different places (e.g. preparing drink on the coffee desk and on the phone desk), it is not easy to obtain high precision rates. However, compared to the online version of [233], our approach detects all activities except two (one "Prepare Drin" and one "Watering Plant") and achieves a much better sensitivity rate compared to the other methods. The online version of [233] fails to detect activities accurately, thereby misses some of the "Prepare Drink" and "Reading Article" activities and gives many false positives for all activities.

Compared to the unsupervised approach that use global motion features, we can see that, by combining both features, our approach achieves more discriminative and precise models, thereby improves both sensitivity and precision rates.
On the GAADRD public dataset, we also compared our results with recent approach proposed in [8] which uses a statistical method to detect delineation of activities. In spite of some of the activities which they perform better than ours in recall (3 out of 7 activities), in turn, our approach significantly outperforms theirs in precision rates. For these activities their approach is better in recall but fails in precision. However, ours, always perform better in recognition compared to them. It is worth to mention that in their method the values in the table are for 10 percent overlap ratio between ground-truth and detected intervals, and recognition accuracy drops significantly when overlap ration increases –from higher than 80% average accuracy with 10% overlap to lower than 20% when overlap ratio is 90%. Performance of our approach does not fluctuate by changing overlapping ratio (which is set to 80% overlap) since it is capable to detect precise delineations (Fig. 5.9). In overall, we can conclude for both datasets, in most of the activities we have increased the true positive and decreased the false positive rates. Thanks to the complete representation of activities with global and local motion descriptors, our approach provides a more precise recognition of activities and better performance.

Figure 5.9 illustrates the performance of clipping and activity discovery on one video from CHU dataset. More than the quality of the recognition process, performance of automatically clipping is crucial for real-world settings. The activities are precisely detected compared to the manually annotated ground-truth intervals. In worst case ("Reading Article"), there is around 120 frames (4 seconds which is less than 3%) gap between ground-truth intervals and automatically detected intervals. This shows the efficiency of clipping mechanism once the scene models with hierarchical topologies are learned. In most of the cases delineations of activities are precisely detected compared to ground-truth intervals. Also, the comparison of recognition accuracy against the approach with manually clipped videos is implicitly the comparison of detection and clipping efficiency. Because recognition in the supervised approach is based on activity intervals retrieved directly from ground-truth but in the proposed approach recognition is based on automatically clipped activity intervals.

One critical question which needs to be answered is how representative a model could be, if it is generated under specific scene regions? There are some activities that could prolong in different regions. For example people could talk to a portable phone while they are walking through different zones. Moreover, There are some activities which are

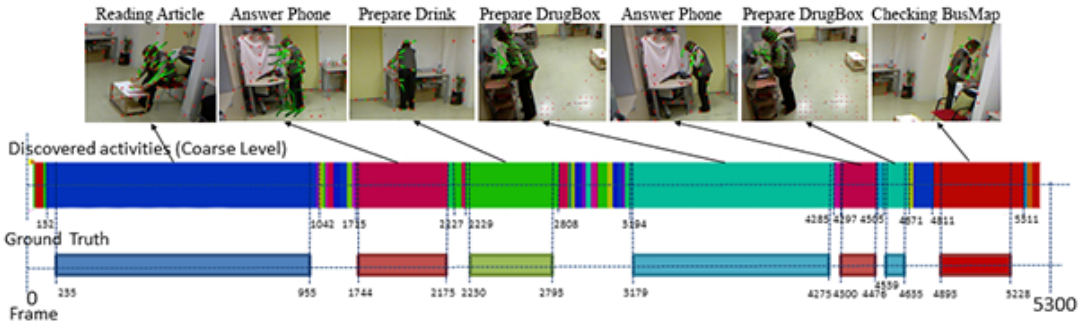


Figure 5.9: Example of automatically clipping and discovering activities for a video of one person performing everyday activities in CHU dataset.

not necessarily located within one region. Thanks to automatic clipping of the videos, our approach could also capture these kinds of activities. While person passes through one region to the other one, the clipping takes place and the algorithm treats it as another activity and evaluates it separately. These kinds of activities could happen in any location (e.g. drinking in the kitchen or in the living room). However, a priori probability of an activity is computed during offline training. In general, there is a restricted amount of activities that could happen in one region. Assuming that, our method narrows the search space down and explores only those activities. Therefore, it is capable to discriminate between different activities happening under the scene regions. There is no presumption of a fixed mapping between a region and activities under that region. Time duration distribution and FV representation of the training data is used to learn a priori assumption for probabilities of different activities. In this way, the generated models assume that the activities with a specific duration and motion pattern are more likely to happen than the others in a specific region. This also helps our approach to be independent from the clusters. If several activities happen inside one large zone, the algorithm is capable to separate them using their local motion pattern information coming from the supervised classifier.
There are some scene regions where a more accurate clustering would lead to a better recognition. For example in CHU dataset there is a narrow boundary between ”Preparing Tea“ region and ”Answering Phone“ region which causes some false cluster ID assignments for trajectory point. Although these assignment errors are limited, they can be avoided with a better metric to define clusters. This could result to even better and more precise detection.

The generated activity models are inherently generic and a trained activity model in one environment could potentially be transformed and used for recognition in a new one.

But this transformative learning is challenging and several issues should be addressed. For example scene region information is an important issue. It is highly likely that scene region partitions would have different IDs and delineations in different environments. In such cases, the scene regions need to be rediscovered in targeted environments as an extra step. However, previously learned FV encoders for the local dynamics in source environment, do not need to be relearned. In the next chapter, we describe an approach to match different zones coming from different environment or a handcrafted model.

5.4 Unsupervised Activity Recognition Framework

As explained, recognizing activities in long-term videos is a challenging problem. Most of the proposed methods are applied in very well-organized datasets that contain manually clipped videos, thereby require excessive supervision of the user. In this section, we describe the second variation of our activity recognition framework which is an unsupervised approach. This framework provides a complete representation of human activities by incorporating both global and local motion and appearance information. Similar to the previous architecture, it automatically finds important regions in the scene and creates a sequence of primitive events in order to localize activities in time and learn the global motion pattern of people. In addition, it uses a large variety of features (*eg.* HOG, HOF, deep features) as an implicit hint to perform accurate activity recognition.

5.4.1 Overview

As Figure 5.10 shows, by using the unsupervised approach, we create a high-level summary of activities in the scene. Most of the building blocks of this framework are similar to the previous ones which use activity discovery and modeling approaches described in chapter 4. However, the most important difference between the two architectures is the way they handle the extracted features. In the previous architecture, after the feature calculation and encoding, a supervised classifier based on the encoded features is trained. During the testing, when a new video instance is provided, after activity discovery and construction of hierarchical model, the supervised classifier is used for classification of the video using its extracted features. Then, the obtained categorical information from the classification process is embedded in the unsupervised model and assists the created hybrid model to achieve better performance. The current architecture does not employ the supervised classifier, though it still benefits from the set of extracted informative features. Vector clustering of scene regions helps to find the distribution of the feature vectors and later, these distributions are considered for finding the distance of unseen vectors with the calculated feature spaces. The given feature vector is assigned to the

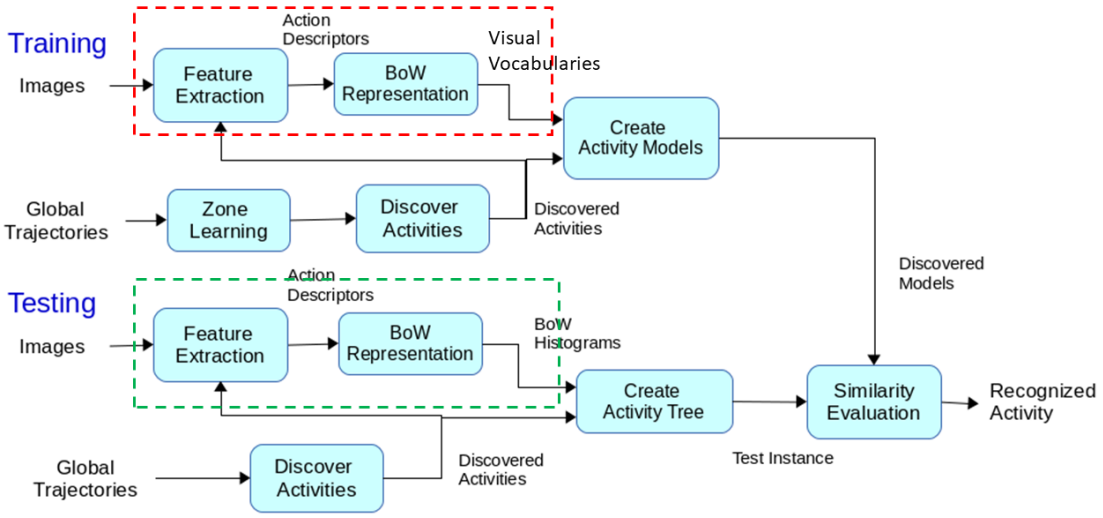


Figure 5.10: The flow diagram of the unsupervised framework: Training and Testing phases. The red dashed box shows the training of the visual vocabularies of the descriptors. The green box in the testing phase shows the descriptor matching procedure. The rest of the framework is similar to the previous architecture.

activity model with the closest distance from its distribution.

Similar to the previous architecture, first, the long-term videos are processed to obtain trajectory information of people's movement. This information is used to learn scene regions. We learn such regions by finding the parts of the scene where people spend most of their time, i.e. dense regions in terms of trajectory points. A common approach in unsupervised approaches is to assume that there is only one kind of action occurring inside a region [62, 87, 154]. However, in unstructured scene settings this assumption may not be valid. In order to distinguish actions occurring inside the same region, we benefit from the local motion and appearance features. The learned regions are employed to create primitive events which basically determine primitive state transitions between adjacent trajectory points. Based on the acquired primitive events, a sequence of discovered activities is created to define the global motion pattern of people, such as staying inside a region, moving between regions. For each discovered activity, motion statistics, such as time duration, are calculated to represent the global motion of the person. Finally, a model of a certain activity is constructed through integration of all extracted feature and attributes.

In fact, discovered activities provide a summary of video as a sequence of clips. By extracting action descriptors for each discovered activity and following the well-known bag-of-words representation, we represent the local motion of the discovered activities. In training phase, for each action inside a scene region, an activity model with a comprehensive activity representation is created by combining both global and local motion information. The obtained quntized features of the activities (visual vocabularies) are combined with each activity model created in unsupervised phase and result a complete model of individual activities inside each region. Later, these models will be used to recognize activities. During the testing phase, the learned regions are used to obtain primitive events of the test video. Again, the video is clipped using discovered zones and action descriptors are extracted for each discovered activity. Similar to the training phase, for each discovered activity, by combining the local motion information with global motion and other attributes an activity model is constructed. To recognize activities, a comparison between trained activity models and acquired test activity takes place. A similarity score between the test instance and trained activity models is calculated by comparing global and local motion information of the models. The activity model with maximum similarity score is considered as the recognized activity of the test instance. In the following section we describe the main difference of the two architectures. We explain how the extracted features are used in the activity models without the classification step.

5.4.2 Descriptor Matching

Our descriptor matching can be seen as a method capturing similarity of a given local information of an activity with a set of calculated multi-dimensional distributions. The obtained descriptor vectors (H) characterizes local motion and appearance of a subject. Having the descriptors of discovered activities with vector representation helps to use a distance (Eq. 5.6) measurement to characterize the similarity between different activities. The drawback of the proposed approach is that the descriptor is calculated only on the root node (DA level) and characterizes an activity as a whole without considering the descriptors of the nodes in lower levels of resolution and the hierarchical link of the sub-activities. This may cause serious problems in modeling of short term actions such as a slight "movement of arm". Although this information is present in the DA's descriptor, including hierarchical dependencies of such short actions rather than adding to the complexity of the model seems not make a significant contribution in the whole characteristics of long-term activities.

As it is shown in figure 5.11, during the training phase, the scene model is used to clip the long videos to the short clips belonging to each region. Next, the descriptors of

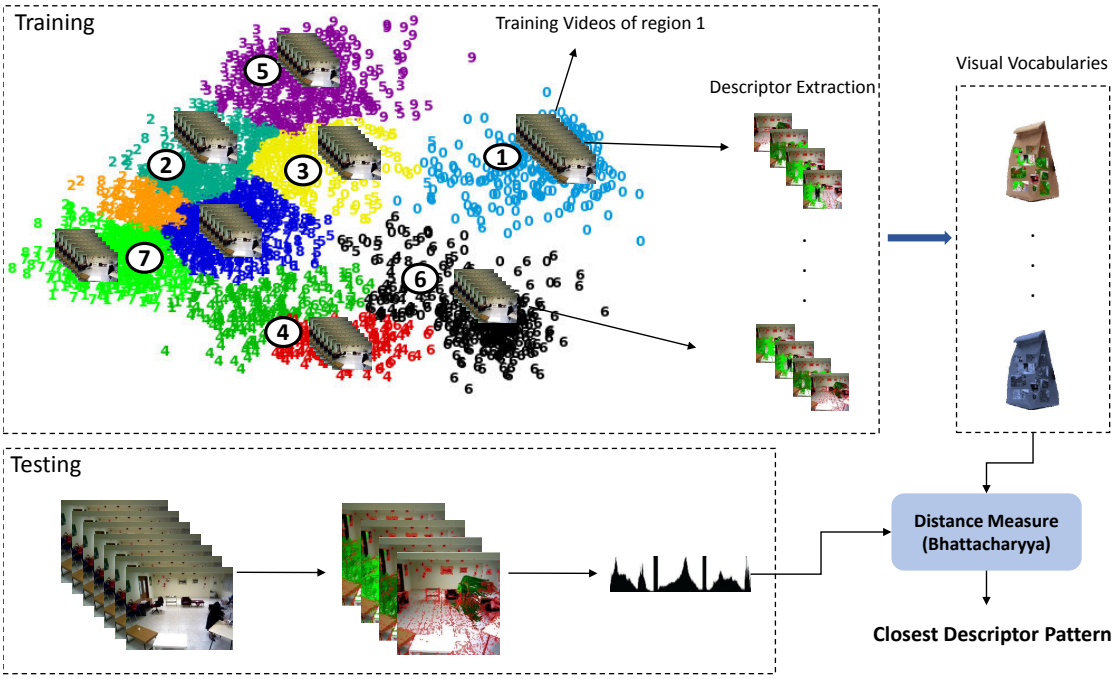


Figure 5.11: The diagram showing the process of learning visual vocabulary for each activity model and matching the given activity's features with the most similar dictionary. Training and Testing phases.

the clipped videos are extracted and employed to learn a visual vocabulary (one for each region) by clustering of the descriptors (Using k-means). The vocabulary of each region is stored in the created activity model of that region. During the testing phase, when a new video is detected by the scene model, its descriptors are extracted and the feature vectors are created. Then these feature vectors are encoded with the learned dictionaries of the models. The distance of the current descriptor is calculated with the trained vocabularies of all regions (to find the closest one) using the Bhattacharyya distance:

$$Distance(H, V) = \sum_{i=1}^{N} BC(H, V_i) \tag{5.6}$$

where N is the number of learned code words and BC is the Bhattacharyya coefficient:

$$BC = \sum_{x,y=1}^{N,M} H(x)V_i(y) \tag{5.7}$$

where N and M are the dimensions of the descriptor and the trained vocabularies. The

most similar vocabulary is determined by the minimum distance score acquired. That vocabulary (and its corresponding activity model) is assigned by a higher score in the calculation of the final similarity score with the test instance in the recognition phase.

5.4.3 Experiments

In this section, we report performance of the proposed unsupervised framework on the two datasets:

- GAADRD
- CHU Nice Hospital

We use the same protocol as we used in the previous experiments. In both datasets, 3/5 of the videos in these datasets are used for training and the remaining 2/5 of the videos are used for testing. Figure 5.5 describes the procedure of training, testing, and evaluation of the framework. The only difference of current framework's evaluation procedure with the previous versions is in the evaluation step. Since the recognized activities are not labeled, there is no matching ground-truth activity label for them. For example, the recognized activities are labeled as "Activity 2 in Zone 1". In order to evaluate the recognition performance, first, we map the recognized activity intervals on the labeled ground-truth ranges. Next, we evaluate one-to-one correspondence between a recognized activity and a ground-truth label. For example, we check which ground-truth activity label co-occurs the most with "Activity 2 in Zone 1". We observe that in 80% of the time this activity coincides with "Prepare Drink" label in the ground-truth. We infer that "Activity 2 in Zone 1" is "Prepare Drink" activity. For this purpose, we create a correspondence matrix for each activity. The correspondence matrix is defined as a square matrix where its rows are the recognized activities and the columns are ground-truth labels. Each element of the matrix shows the number of co-occurrences of that recognized activity with the related ground-truth label in that column:

$$COR(RA, GT) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

Where $a_{ij} \in Z^+$ shows the correspondence between activity instance i and ground-truth label j. RA is the set of recognized activity instances and GT shows the set of ground-truth labels. We evaluate performance of the framework based on the inferred labels. These labels are used for calculating the *Precision*, *Recall*, and *F-Score* metrics.

5.4.3.1 GAADRD Dataset

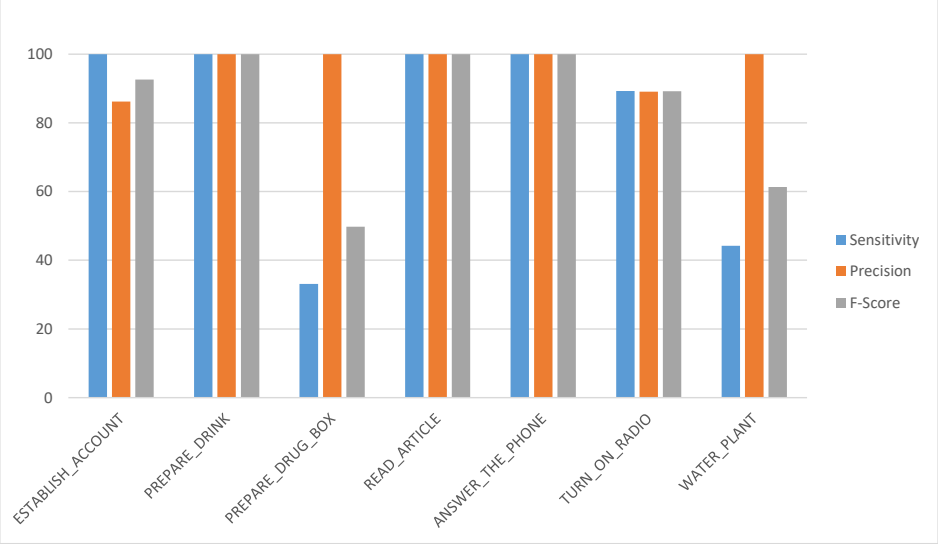
The details about this dataset are explained in section 3.7.1 of chapter 3. As explained in chapter 3 we are using Fisher Vector encoding for activity representation. We have tried different codebook sizes for the FV dictionaries (16, 32, 64, 128, 256, and 512). Here we report the results of applying the unsupervised framework on the GAADRD dataset. Table 5.7 shows overall accuracy with Precision and Recall metrics using different feature types. The plot on top illustrate F-Score information of the table. Based on the obtained results we can conclude that:

- Based on the obtained results, the unsupervised framework achieve competitive results with the hybrid and supervised frameworks on this dataset. It significantly outperforms the performance of sliding window approach. It also surpass the version of framework when only global information is used.
- There is no special trend regarding the codebook size. For some features (MBHY and TDD spatial) the performance increases codebook size increases and drops when the codebook size becomes bigger. For TDD temporal feature, performance increases linearly with the codebook size. For the geometrical features, specially for Angle feature, there is a big drop of performance with bigger codebook sizes. For others (HOG,HOF), medium size codebook performs the best. Finding optimal codebook size is challenging. Usually, small datasets work better with smaller codebook size and as the datasets' size grows bigger codebook performs better.
- Regardless of the codebook size MBHY descriptor performs better than other features. The MBH descriptor composed of X and Y components. Since the activities involve many vertical motion, MBHY descriptor is able to model the activities better compared to other dense trajectory descriptors and even deep features.
- Motion features (TDD temporal, MBHY) performs better than appearance features.
- Temporal deep features perform better than spatial TDDs. The activities are performed in a hospital environment, hence, the background does not contain discriminative information that can be encoded in activity models. Moreover, the performance of temporal deep features gets better as codebook size gets bigger.
- Similar to supervised and hybrid frameworks, geometrical features perform poorly. Daily activities consist of many sub-activities with similar motion pattern which are related to object interactions. It seems that geometrical features do not contain sufficient information to encode these kinds of motion. Further analysis show that the activities with similar motion are confused with each other the most.

Table 5.7: Results of using the unsupervised framework with different feature types on GAADRD dataset. The plot shows F-Score values w.r.t. codebook size.

	16			32			64			128			256			512		
	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score
Angle	54.8	32.9	0.41	57.6	33.2	0.42	61.2	36.1	0.45	46.9	30.2	0.36	28.1	22.4	0.24	26.7	19.8	0.22
Distance	8.1	5.2	0.06	12.9	9.7	0.11	18.2	14.9	0.16	20.7	16.1	0.18	14.7	12.1	0.13	14.7	15.2	0.14
HOG	80.2	75.4	0.77	81.4	75.2	0.78	84.7	79.6	0.825	77.5	74.3	0.75	82.7	77.6	0.80	84.7	79.8	0.82
HOF	61.2	62.7	0.61	64.6	61.9	0.63	64.9	67.7	0.66	66.1	68.1	0.67	65.4	67.9	0.66	57.4	62.1	0.59
MBHX	66.1	70.2	0.68	71.3	77.2	0.74	74.8	78.2	0.76	79.8	76.1	0.77	67.6	72.1	0.69	69.4	72.8	0.71
MBHY	73.8	73.2	0.73	71.5	68.4	0.69	78.8	76.1	0.77	82.7	84.9	0.83	83.1	85.7	0.84	80.2	79.4	0.79
TDD Spatial	63.8	58.2	0.6	74.5	72.9	0.73	72.8	71.2	0.71	77.5	74.3	0.75	77.5	76.9	0.77	76.4	73.5	0.74
TDD Temporal	57.9	61.6	0.59	73.4	69.1	0.71	73.9	70.6	0.72	72.5	69.9	0.71	79.4	76.2	0.77	81.9	76.9	0.79

Figure 5.12 show more the details about the MBHY feature that achieves the best result on GAADRD dataset. Detailed analysis of all descriptors are provided in the appendix A.



	MBHY		
	Precision [%]	Recall [%]	F-Score
Establish Account	86.2	100	0.92
Prepare Drink	100	100	1.0
Prepare DrugBox	100	33.1	0.49
Read Article	100	100	1.0
Answer the Phone	100	100	1.0
Turn On Radio	89.1	89.3	0.89
Watering Plant	100	44.2	0.61
Average	96.47	80.94	0.84

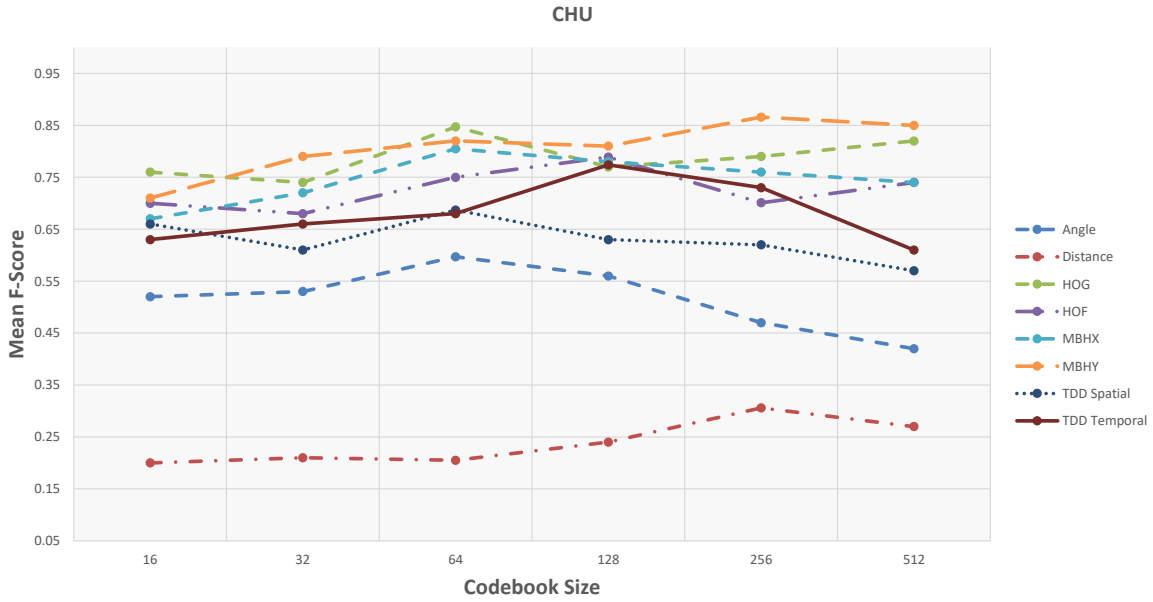
Figure 5.12: Results of applying the unsupervised framework on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the MBHY feature for descriptor matching procedure. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. This feature type achieves the best accuracy when it is used with the unsupervised architecture. It achieves 0.84 in F-Score which is only 0.03 lower than the performance of this method using the hybrid framework. It is interesting that using this feature, the unsupervised framework outperforms the hybrid framework almost in all of the activity classes on the precision metric which shows the descriminative characteristics of these type of descriptors. In the Recall metric, the unsupervised technique achieves equal or better performance in 6 activities. The two activities which the hybrid method has a clear advantage over the unsupervised method is "Prepare DrugBox" and "Watering Plant" activities. The unsupervised framework obtains its highest performance on the GAADRD dataset when MBHY descriptor is combined with the generated activity models.

5.4.3.2 CHU Dataset

The details about this dataset are explained in section 3.7.2 of chapter 3. As explained in chapter 3 the Fisher Vector encoding is used for calculating the histograms. We have tried different codebook sizes for the FV dictionaries (16, 32, 64, 128, 256, and 512). Next, the results of the unsupervised framework on the CHU Nice Hospital dataset are reported. Table 5.8 shows overall accuracy with Precision and Recall metrics using different feature types. The plots on top illustrate the best F-Score results of each feature type. The results show that:

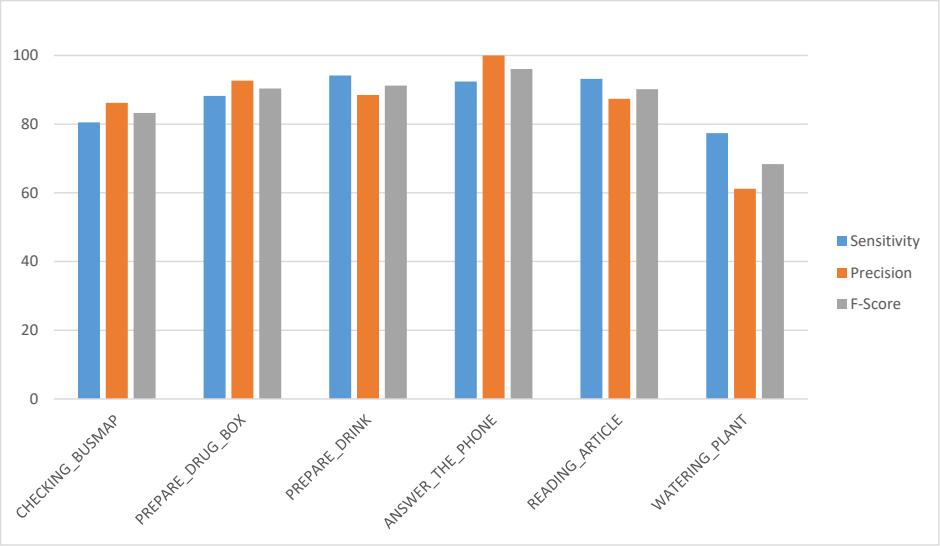
- On this dataset, the unsupervised framework achieves promising results. It obtains lower performance when it is compared with fully supervised and hybrid frameworks. However, it achieves best performance when descriptor information is removed from the models.
- Similar to the GAADRD dataset, the effect of codebook size is different for different descriptor types in this dataset. For MBHY descriptor, the accuracy increases as codebook size grows, whilst, it has opposite effect on TDD appearance features. Differently, the accuracy increases and then decreases for TDD temporal feature. We can say that bigger codebook size results in a better performance. This is different from GAADRD dataset and the reason might be because of larger size of this dataset.
- TDD temporal features achieves better performance than deep appearance features. Due to similar background of the daily activities, temporal information achieve better results.
- MBHY achieves the best performance on this dataset. Abundance of vertical motion in the performed activities helps the MBH descriptors to achieve better recognition performance.
- Among appearance features, HOG descriptor achieves better performance since it can encode the appearance information efficiently. It even outperform deep appearance features.
- Detailed analysis (figure 5.14) shows that this framework has difficulty in recognition of "Watering Plant" activity. It confuses this activity with all other activities. Short duration of this activity leads to insufficient capture of information which results recognition issues. The reason for confusion of the other activities lies mainly on similar motion patterns of the sub-activities. Moreover, this dataset consists of activities that are recorded from subjects lateral view which makes recognition of these activities challenging (detailed discussion about challenging cases in these datasets has given in section 5.5).

Table 5.8: Results of using the unsupervised framework with different feature types on CHU dataset. The plot shows F-Score values w.r.t. codebook size.



	16			32			64			128			256			512		
	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score	Prec. [%]	Rec. [%]	F-Score
Angle	58.6	48.1	0.52	58.4	49.7	0.53	60.7	57.8	0.59	58.6	55.2	0.56	50.3	45.9	0.47	41.7	44.1	0.42
Distance	23.7	18.9	0.2	23.9	19.2	0.21	22.7	19.5	0.20	27.8	21.7	0.24	29.2	31.9	0.30	28.8	27.1	0.27
HOG	78.9	74.6	0.76	77.7	71.9	0.74	85.7	82.9	0.84	80.8	74.9	0.77	81.9	76.3	0.79	84.9	79.8	0.82
HOF	69.7	72.1	0.7	68.2	69.8	0.68	73.9	76.4	0.75	77.1	79.1	0.78	68.4	71.9	0.70	73.4	74.9	0.74
MBHX	67.1	67.7	0.67	73.4	72.1	0.72	81.3	80.4	0.80	78.6	79.2	0.78	75.2	78.3	0.76	73.4	76.2	0.74
MBHY	71.3	72.1	0.71	80.5	77.9	0.79	84.3	79.9	0.82	83.9	79.3	0.81	88.6	83.6	0.866	87.4	83.1	0.85
TDD Spatial	69.4	64.3	0.66	65.8	58.4	0.61	71.9	64.7	0.68	67.2	60.9	0.63	65.9	60.1	0.62	60.0	55.9	0.57
TDD Temporal	64.8	61.5	0.63	67.7	65.7	0.66	69.7	66.1	0.68	79.2	76.1	0.77	74.4	73.5	0.73	61.8	62.1	0.61

Figure 5.13 show more the details about the MBHY feature that achieves the best result on CHU dataset. Detailed analysis of all descriptors are provided in the appendix A.



	MBHY		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	86.2	80.5	0.83
Preparing DrugBox	92.7	88.2	0.90
Prepare Drink	88.5	94.2	0.91
Answer the Phone	100	92.4	0.96
Reading Article	87.4	93.2	0.90
Watering Plant	61.2	77.4	0.68
Average	86.00	87.65	0.86

Figure 5.13: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the MBHY feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Like the GAADRD dataset, MBHY descriptor when combined with the HAM models achieves the best results among all the descriptors on the unsupervised framework. It achieves relatively close performance with its hybrid counterpart (0.87 versus 0.93 F-Score). On the hybrid framework the HOG was the winning descriptor on both datasets, while the MBHY descriptor is dominant when the unsupervised framework is applied. In most of the activity classes MBHY achieves similar performance to the one in the hybrid and most of the supervised method except its optimal codebook size in FV encoding procedure.

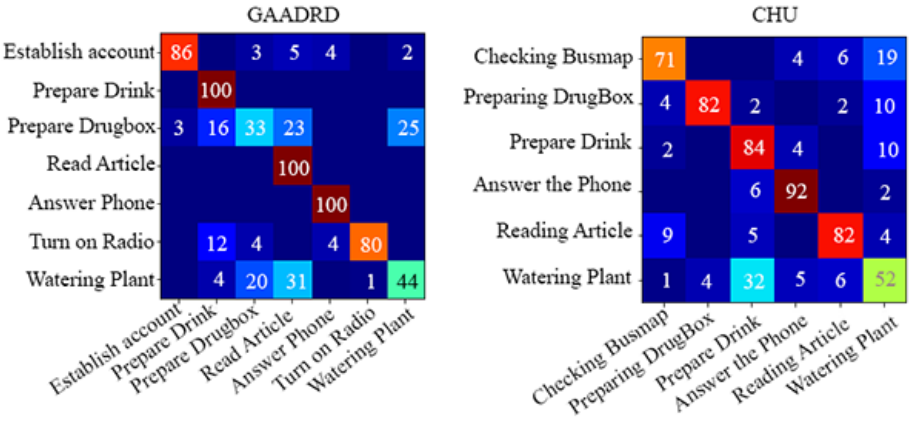
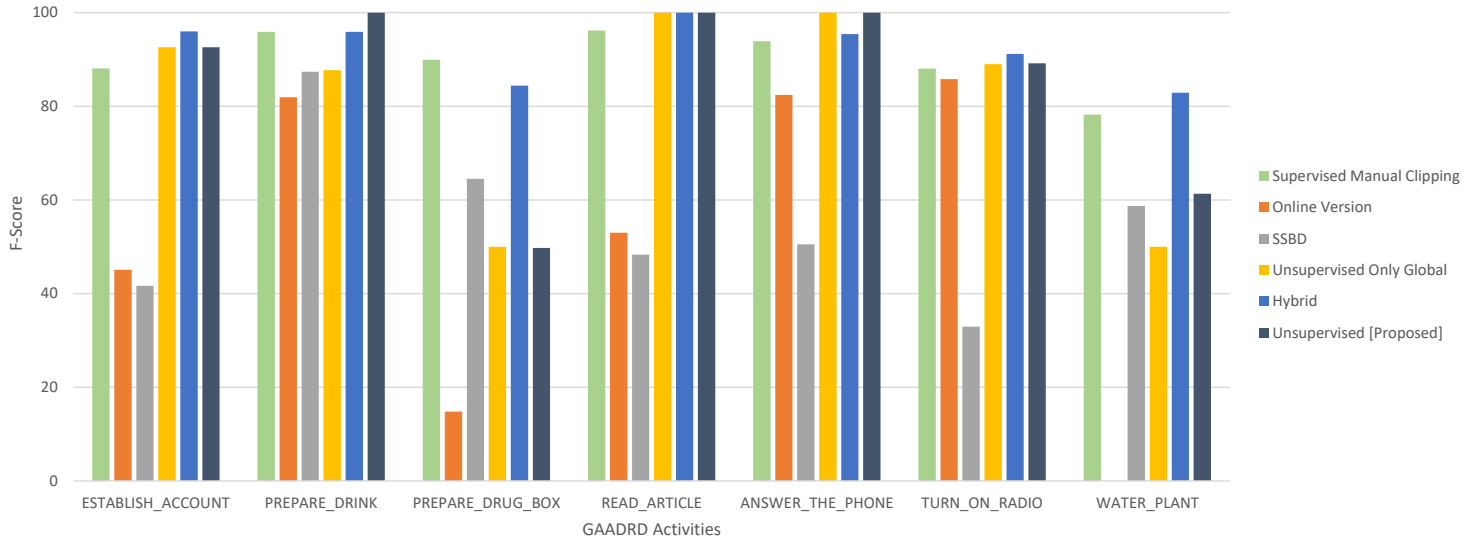


Figure 5.14: Confusion matrices of the best configuration of the unsupervised framework on GAADRD and CHU datasets (with MBHY descriptor). The values show mean accuracy (%).

5.4.4 Discussion and Comparison

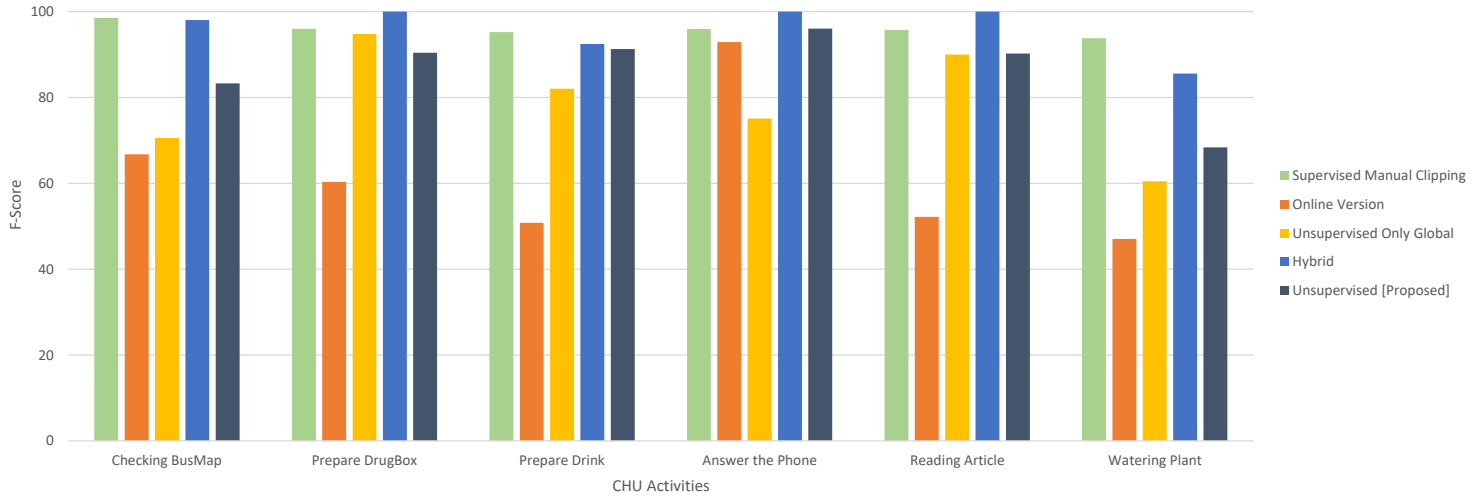
The results of unsupervised framework on two datasets reported in the previous section. Although the unsupervised framework did not utilized any supervised information, it achieved promising recognition performances. The evaluations showed that the framework is capable of recognizing targeted activities with acceptable error-rate. The unsupervised framework could not achieve the same recognition rate as of the hybrid model. However, the high performance of the hybrid method comes with a cost. The cost is human supervision. In the hybrid method a supervised SVM classifier is trained with the ground-truth annotation provided by human. However, in the unsupervised method no supervision is imposed. So, if we can say that the privilege of the hybrid method over the fully supervised method is its online recognition procedure providing automatic detection of the activities, we can emphasize that the main benefit of the unsupervised method is automatic online clipping and detection of activities as well as unsupervised modelling and recognition. With all these benefits, mediocre recognition rate of the unsupervised method is admissible. Unlike the hybrid method that combining HOG descriptor with the hierarchical models resulted in the best performance, the unsupervised framework achieved the best performance by using the MBHY descriptor. The MBHY performed the best on both datasets. It is worth to mention that the deep features showed the highest stability of performance when we changed from hybrid framework to unsupervised. It is also interesting to see that the deep TDD features are the only feature type that the generated unsupervised models based on them outperformed hybrid models. We have compared our approach with the results of the same methods we did for the hybrid framework.

Table 5.9: Comparison of different recognition frameworks with ours on the GAADRD dataset. The methods are differentiated by using different color codes. The diagram shows class-wise accuracy of each method with respect to their F-Score values. The table in below shows the detailed results of each method with respect to each class in the dataset. The best results in each section is indicated in bold.



	Supervised (Manual Clipping) with HOG, Dict sz=512 [233]			Online Version of [233]			Classification by Detection SSBD [8]			Unsupervised Using Only Global Motion [62]			Hybrid			Unsupervised (Proposed Method)		
	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score
Establish Account	92.2	84.3	0.88	29.1	**100**	0.45	41.67	41.67	0.41	86.2	**100**	0.92	**92.3**	**100**	**0.95**	86.2	**100**	0.92
Prepare Drink	92.1	**100**	0.95	69.4	**100**	0.81	80.0	96.2	0.87	**100**	78.1	0.87	**100**	92.1	0.95	**100**	100	**1.0**
Prepare DrugBox	94.9	85.5	**0.89**	20.2	11.7	0.14	51.28	86.96	0.64	**100**	33.34	0.50	78.5	**91.3**	0.84	**100**	33.1	0.49
Reading Article	96.2	96.2	0.96	37.8	88.6	0.52	31.88	**100**	0.48	**100**	**100**	1.0	**100**	**100**	**1.0**	**100**	**100**	**1.0**
Answer the Phone	88.5	**100**	0.93	70.1	**100**	0.82	34.29	96.0	0.50	**100**	**100**	**1.0**	**100**	91.2	0.95	**100**	**100**	**1.0**
Turn On Radio	**89.4**	86.7	0.88	75.1	**100**	0.85	19.86	96.55	0.32	89.0	89.0	0.89	89.1	93.4	**0.91**	89.1	89.3	0.89
Watering Plant	84.8	72.6	0.78	0	0	0	44.45	**86.36**	0.58	57.1	44.45	0.49	79.9	86.1	0.82	**100**	44.2	0.61
Average	91.16	89.33	0.90	43.1	71.4	0.51	43.34	86.24	0.54	90.32	77.84	0.81	91.4	**93.44**	**0.92**	**96.47**	80.94	0.84

Table 5.10: Comparison of different recognition frameworks with ours on the CHU dataset. The methods are differentiated by using different color codes. The diagram shows class-wise accuracy of each method based on F-Score metric. The table in below shows the detailed results of each method with respect to each class in the dataset. The best results in each section is indicated in bold.



	Supervised (Manual Clipping) with HOG, Dict sz=256 [233]			Online Version of [233]			Unsupervised Using Only Global Motion [62]			Hybrid			Unsupervised (Proposed Method)		
	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score
Checking BusMap	**100**	97.1	**0.98**	50.1	**100**	0.66	54.54	**100**	0.70	96.1	**100**	**0.98**	80.5	86.2	0.83
Prepare DrugBox	**100**	92.3	0.95	43.2	**100**	0.60	**100**	90.1	0.94	**100**	**100**	**1.0**	88.2	92.7	0.90
Prepare Drink	**93.1**	**97.4**	**0.95**	38.1	76.1	0.50	80.0	84.21	0.82	88.9	96.3	0.92	**94.2**	88.5	0.91
Answer the Phone	92.2	**100**	0.95	86.7	**100**	0.92	60.1	**100**	0.75	**100**	**100**	**1.0**	92.4	**100**	0.96
Reading Article	97.5	94.1	0.95	36.4	92.0	0.52	**100**	81.82	0.90	**100**	**100**	**1.0**	93.2	87.4	0.90
Watering Plant	**100**	88.3	**0.93**	33.9	76.9	0.47	53.9	68.9	0.60	77.0	96.3	0.85	77.4	61.2	0.68
Average	**97.13**	94.87	0.95	48.06	90.83	0.61	74.75	87.50	0.78	93.66	**98.76**	**0.96**	87.65	86.00	0.86

The performance of the other approaches and our approach on GAADRD dataset are presented in Table 5.9. In all approaches that use body motion and appearance features, the feature types with the best performances are selected. It can be clearly seen that, using models that represent both global and body motion features, our unsupervised approach enables to obtain high sensitivity and precision rates. Compared to the online version of [233], thanks to the learned zones from positions and discovered activities, we obtain better activity localization, thereby better precision. However, since the online version of [233] utilizes only dense trajectories (not global motion), it fails to localize activities. Hence, it detects the intervals that does not include an activity (e.g. walking from radio desk to phone desk). For the "Watering Plant" activity this method can not detect and recognize any instances of this activity, hence the Precision, Recall, and F-Score rates are zero. Compared to the unsupervised approach that either use global motion features or body motion features, we can see that, by combining both features, our approach achieves more discriminative and precise models, thereby improves both sensitivity and precision rates. For instance, for "Answer the Phone", "Establish Account", "Reading Article", and "Turn On Radio" activities, global motion features are more discriminative (global position might be enough to distinguish these activities) and for "Preparing Drink" and "Watering Plant" activities, body motion features are more discriminative and precise (since local motion help to distinguish them from the others). From the confusion matrices in figure 5.14 it is apparent that the "Watering Plant" activity is the most confused activity. This is mostly related to low frame number of this activity which makes it difficult to model, hence, it gets confused with other activities. By combining global and body motion features, our approach benefits from discriminative properties of both features. Table 5.9 also presents the results of the supervised approach in [233]. The supervised approach uses ground-truth intervals in test videos in an offline recognition scheme. As our approach learns the scene region model, we discover the places where the activities occur, thereby we achieve precise and accurate recognition with a lower cost. Since scene region information is missing in the supervised approach, it detects "Turning On Radio" while the person is inside the "Preparing Drink" region. On this dataset, the unsupervised method always performs better than the "Online Supervised" approach and significantly outperforms the sequential statistical boundary detection (SSBD) method. It also outperforms the another unsupervised version of the framework while no descriptor information is used in the activity models. Only the supervised methods surpass our unsupervised models. The reason is that the supervised method works with preclipped activity videos and overlooks the challenging task of temporally segment activity samples from the original video flow.

Table 5.10 shows the results of evaluated approaches and their comparison with ours

on CHU Nice Hospital dataset. In this dataset, since people tend to perform some activities in different places (*eg.* preparing drink at phone desk), it is not easy to obtain high precision rates. However, compared to the online version of [233], our approach detects all activities and achieves a much better precision rate. The online version of [233] again fails to detect activities accurately, thereby misses some of the "Prepare Drink" and "Reading Article" activities and gives lots of false positives for all other activities. It cannot handle the transition states that occur between activities (e.g. walking from telephone desk to DrugBox). For this reason, a random label is given for transition states, which consequently increases false positives. Compared to the Online Supervised method, we have increased the average precision rate from 48.06% to 87.654%. Compared to the unsupervised method without embedded descriptor information, we have decreased the false positive rates and increased the precision rates significantly. The highest improvements are on "Answering Phone" from 60% to 92%, "Checking BusMap" from 54.54% to 80.5%, "Prepare Drink" from 80% to 94% and "Watering Plant" from 53% to 77%. For "Reading Article" activity, there is small increase in false positive rates, thereby a decrease in precision rates. This might be because of lack of local motion because of long sitting position of this activity and non ideal activity detection compared to manual clipping. Since the motion representation of [62] contains only global information, it fails to distinguish activities inside the regions precisely, *eg.* , passing by the phone zone and answering phone in the phone zone is considered as the same activities in those models. Hence, the unsupervised approach results high false positive rates. In addition, we can observe that the proposed approach improves the true positive rates and increased sensitivity rates for most of the activities when it is compared to the "Only Global Motion" method. In overall, we can conclude that the unsupervised method can not achieve the high accuracy of supervised and hybrid models, however, it manages to obtain acceptable and competitive results in most of the tasks with the benefit of cutting the cost of highly expensive supervision procedures.

5.4.5 Results of Knowledge-based Region Refinement Framework

5.4.5.1 GAADRD dataset

From table 5.11, we can observe that adapting activity models enables us to detect activities missed by the hand-crafted method (knowledge-based approach [45] - KB). In [45] to try a different scenario, a new configuration of the hand-crafted regions should be set. The reason that the "Establish Account" activity's performance is 0, is because the user missed this region in calculations (drawing zone). Therefore, there is no way for the framework to recognize this activity. However, due to knowledge interaction between the two framework, the unsupervised framework detects this region and creates and activity model for it (As explained in section 4.5 of chapter 4). Hence, this activity can be recognized by

	KB [45]			Proposed Approach		
	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score
Establish Account	-	-	-	66.67	100	0.80
Prepare Drink	100	100	1	100	100	1
Prepare DrugBox	75	33.34	0.46	60	66.67	0.63
Reading Article	46.15	75	0.57	60	75	0.66
Answer the Phone	100	85.71	0.92	100	85.71	0.92
Turn on Radio	100	90	0.94	100	90	0.94
Watering Plant	50	55.56	0.52	40	66.67	0.50
AVERAGE	78.25	73.26	0.75	75.23	83.45	0.79

Table 5.11: The activity recognition results of KB (the knowledge-based approach in [45]), and our data-driven knowledge-based approach for the GAADRD dataset.

our framework. However, the recognition results are not impressive since no descriptor is used in the activity models.

Compared to KB, our approach increased recall rates of "Prepare DrugBox" and "Watering Plant" activities from 33.34% to 66.67% and 55.56% to 66.67%, respectively. However, for the same activities, there is an increase in the number of false positive events. The reason is that, for "Watering Plant" activity we do not have enough data in the training set to learn the duration distribution of the activity. Thus, the learned distribution does not completely represent the actual characteristics of this activity and, thereby, the tailored activity models are not accurate. For "Preparing DrugBox" activity, the duration and posture distributions are bi-modal distributions because of other activities occurring in the same zone, such as reading a paper inside pharmacy zone. Additionally, for "Reading Article" activity, we can observe that the proposed approach increased precision rates from 46.15% to 60%. On average, we have increased the recall rates from 73.26% to 83.45% with a slight decrease in the precision rates.

5.4.5.2 CHU dataset

Based on table 5.12, we can observe that adapting the constraints in activity models using data learned by the unsupervised module, in majority of the cases, enables us to detect activities missed by the hand-crafted models. Compared to KB [45], the proposed method increased the average recall rate from 82.44% to 93.96%. As the hand-crafted activity models of the knowledge-based approach are pre-defined, they do not match with all the activities of the monitored person, thereby requiring an update by the domain expert. Hence, the knowledge-based approach fails to detect activities when default parameters do not fit well with the monitored person. We can also see that the proposed approach increases the true positives for all activities. Especially, for "Watering Plant" activities, the proposed approach drastically increased the recall rates from 60% to 80%, respectively.

	KBcrispimjunioravss2013			Proposed Approach		
	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score
Checking BusMap	58.82	100	0.74	100	100	1
Prepare DrugBox	71.42	100	0.83	100	100	0.63
Prepare Drink	59.09	92.85	0.72	73.68	100	0.66
Answer the Phone	90.47	100	0.94	90.47	100	0.92
Reading Article	90.91	90.91	0.90	84.61	100	0.94
Watering Plant	60	60	0.60	66.67	80	0.50
AVERAGE	70.10	82.44	0.75	81.96	93.96	0.87

Table 5.12: The activity recognition results of KB (the knowledge-based approach in [45]), compared with our data-driven approach for the CHU dataset.

In addition, the false positives are decreased for the majority of activities. Our approach increased true positives and decreased false positives, achieving 100% of performance at recall and precision rates for "Checking BusMap" and "Preparing DrugBox" activities.

5.5 Experimental Challenges

Activities of daily living are complex and modeling these activities is challenging and recognizing their pattern in subjects are difficult. These challenges exist and are reflected on the recorded datasets that we used in our experiments which caused the recognition errors in our evaluations. Here we describe some of the important and challenging examples being faced.

One major problem which makes activity recognition task complex is activities with high intra-class variation. As it is illustrated in figure 5.15, on top right we see a casual way of "Answer the Phone" activity. In bottom right, we see the subject doing the same activity but with an unusual posture. The label in the ground-truth for both of the activities is "Answer the Phone", however, for a model that relies on posture, geometry and appearance features, correct recognition of both instances is challenging due to their dissimilar postures. Nevertheless, our hybrid model by benefiting from both local descriptors and the global tracking features (abstracted in the form of scene regions) handles such complexity and most of the time predicts the correct label of the activity. Sometimes the information coming from the descriptors can be misleading. The dynamics of the performed activity could be identical for two semantically different activities. Consider the example in top left of figure 5.15. The subject stands in the "Phone" scene region, hence it activates attributes regarding this scene region in the trained model for this targeted activity. If we look carefully, we will see that although the subject's posture is identical to "Answer the Phone" activity, she is "Scratching" her head. There is no

Figure 5.15: Challenges in the experiments: Answer the Phone activity. Top right: the correct activity performance; Top left: scratching head; Down right: Performing the activity in an improper posture; Down left: Reading Article in the Answer the Phone scene region.

"Answer the Phone" annotation for this interval in the ground-truth. In spite of having precise activity model equipped with different descriptors, this is always considered as a false positive in the evaluations. This is because both instances consist of similar motion patterns indistinguishable by the local motion descriptors.

Another challenging case regarding the recognition task is related to the activities that are not happening in the scene regions with a higher prior probability for that particular activity. For instance, the "Reading Article" activity in the CHU Hospital dataset has a high frequency in the region including chair and the desk where the subject sit and read the article. However, as it is shown in bottom left of figure 5.15, the subject stands in the "Phone" region and is "Reading Article". This is very challenging case for a model which only relies on global motion information. Thanks to the models including the appearance, posture and local motion descriptors we can recognize such activities. In such occasions,

Figure 5.16: Right: Preparing Drink activity with the subject heading back to the camera's viewpoint. Left: Preparing Drink activity while the subject stands beside the desk in a side view to the viewpoint.

the evaluation mechanism assigns a high score for descriptor component of the model turning over the overall decision of the framework.

As shown in figure 5.16, some challenging cases emerge because of the specific viewpoint. In the left side of figure 5.16, the subject prepares drink while he is visible with a side view to the recording camera's view point. In this instance the subject's hands and the objects are clearly visible and recognizable. Using the local motion descriptor this activity can be easily recognized. But the recognition is not that easy for the instance in the right side of the figure 5.16. The subject stands faced to the wall with his back to the camera. In this case neither his hands nor the objects on the table are visible. Even his body barely moves and the descriptors for catching motion patterns are useless. Moreover, the subject is far from the camera and makes it even more difficult to extract the descriptors properly. The generated model should rely solely on the global motion information to predict this activity.

Some challenges are directly related to the recorded datasets. There are some activities that fewer samples of them are recorded. Since there is no obligation to do activities in certain order or number, some activities happen to be recorded fewer times than the others. For example the "Watering Plant" activity (Figure 5.17) is performed the least in both datasets. This unbalance in the recordings causes some activities to have less data and consequently, less information to learn. The "Watering Plant" activity is among the lowest performance obtained by all of the evaluated frameworks. Having few instances results in a big contribution of a single instance in the overall class-wise performance. The other issue with this activity is related to the obtained scene regions.

Figure 5.17: Right: Watering plant activity in GAADRD dataset. Left: Watering plant activity in CHU dataset.

This activity occurs in the vicinity of the “Prepare Drink” activity. Despite the effort we put to perform a precise clustering of the trajectories, there is no clear boundaries between these two close regions. Sometimes, the test trajectories close to the borderline is assigned to the wrong region (not the targeted activity). This causes wrong detection of the primitive events and accordingly, ill-suited construction of the test instance’s HAM. Finally, a problematic model results in a wrong recognition.

Another problem regarding the recordings that affected the recognition process is their low frame rate. When there is no enough frames to capture a movement, there will be no descriptor extracted and some sub-actions will be missed. For example, consider the "Prepare DrugBox" activity in the figure 5.18. This is an activity in the GAADRD dataset. The frames shown in the image are the total frames annotated with this activity in the ground-truth. It can be noticed that there are only few frames recorded for this activity (10 frames). The minimum trajectory detection frame interval is set to 15 meaning that the trajectories available in these frames will be discarded for being too short. Accordingly, no descriptor will be extracted. This situation can be avoided with tracking of shorter trajectories but then it will add more to the processing time of long videos and will be more prone to noisy trajectories.

5.6 Conclusion

In this chapter, we present two frameworks constructed based on the hierarchical activity models explained in the previous chapter (Chapter 4). In the first architecture, the

Figure 5.18: Illustrates the low frame rate of the videos and short duration of some activities which causes problems for extraction of relevant descriptors.

hierarchical models created based on the global motion of the subjects in the scene are combined with the local descriptors of the activities. This combination takes place in a supervised manner. The extracted descriptors encoded with the FV and go through a training step to learn a SVM classification model. Then, during the test time these classifiers used for producing labels for the extracted descriptors of the given video which is then embedded in its HAM tree. This hybrid method (unsupervised models of global motion combined with supervised classification of local motion) shows impressive recognition results.

In the second variation of the framework, no supervision is imposed to the system. After the construction of activity models with global motion information, the extracted local descriptors are used for training a visual vocabulary. The learned vocabulary is also stored in the constructed models as an attribute. Given an unseen video in the testing phase, its descriptors are extracted and the calculated feature vectors are compared to the trained vocabularies of the activity models. The calculated distances are used as a similarity measure revealing the similarity of the local motion pattern of the given instance to one of the learned vocabularies associated to the scene regions and their corresponding activity models. The created models use neither annotation nor any kind of supervision. The detected activities are mapped to the ground-truth labels on the video time line and the labels with highest frequency of concurrency are assigned to the detected activities. This unsupervised version of the framework showed promising results with minimum or no supervision in modeling of activities. The hybrid method outperforms the unsupervised method in accuracy of predictions, however, this achievement came with the cost of pricey supervision.

In both variations of the framework, we use the superiority of unsupervised approaches on representing global motion patterns. Then, we have benefited from the discriminative local motion features in order to distinguish different actions occurring along with the global patterns.

By incorporating both global and body motion features, we have recognized more precise activities compared to pure supervised and pure unsupervised approaches. Thanks to the proposed scene model, we can perform online recognition of activities with reduced user interaction for clipping and labeling huge amount of short-term actions essential for most of the proposed methods.

Chapter 6

Gesture Recognition

"If I wanted to be a doctor today I'd go to math school."

- Vinod Khosla

Contents

6.1 Summary

In all previous chapters of this thesis, we assume that there is a camera installed in one corner of the room and is capturing videos from subjects performing daily activities. As we described at the end of previous chapter (Chapter 5) such system should confronted with various challenges. One of the major described challenges is the inability to capture the fine movement of the subjects. Inability in capturing these movements causes missing of subtle kinematic cues indispensable for an accurate recognition of activities. Among different features describing an action or activity, motion of upper-body limbs (specially of hands) plays a key role. An ideal recognition system must be able to benefit from this information to overcome the challenges and also to have a comprehensive description of activities.

In this chapter, we introduce a complete gesture recognition system, that later we plan to combine it with our activity recognition framework as a finer component of the system. This way, we can obtain detailed information of the subjects in the scene regions that can be embedded into the activity models to produce more accurate description of activities. In design of our gesture recognition framework, we use different modalities and various methods for analyzing upper-body gestures to have a reliable recognition system.

Similar to our activity recognition framework that its main target application is to monitor elderly people in nursing or smart homes, we also study gesture recognition in a medical context. The main goal is to have an accurate activity recognition framework and utilize it to diagnose specific disorder in older adults. In order to achieve this goal, we use the Praxis test in our study. Praxis test is a gesture-based diagnostic test which has been accepted as diagnostically indicative of cortical pathologies such as Alzheimer's disease. Despite being simple, this test is oftentimes skipped by the clinicians. In this chapter, we propose a novel framework to investigate the potential of *static* and *dynamic* upper-body gestures based on the Praxis test and their potential in a medical framework to automatize the test procedures for computer-assisted cognitive assessment of older adults.

In order to carry out gesture recognition as well as correctness assessment of the performances, we have recollected a novel challenging RGB-D gesture video dataset recorded by Kinect v2, which contains 29 specific gestures suggested by clinicians and recorded both experts and patients performing the gesture set. Moreover, we propose a framework to learn the dynamics of upper-body gestures, considering the videos as sequences of short-term clips of gestures. Our approach first uses body part detection to extract image patches surrounding the hands and then, by means of a fine-tuned convolutional neural network (CNN) model, it learns deep hand features which are then linked to a long short-term memory network to capture the temporal dependencies

between video frames.

We report the results of four developed methods using different modalities. The experiments show effectiveness of our deep learning based approach in gesture recognition and performance assessment tasks. Satisfaction of clinicians from the assessment reports indicates the impact of framework corresponding to the diagnosis.

6.2 Introduction

With overwhelming increase of computers in society and their ubiquitous influence in our daily activities, facilitating human computer interactions has become one of the main challenges in recent years. Hence, there has been a growing interest among the researchers to develop new approaches and better technologies to overcome this problem. The ultimate aim in this process is to achieve more sensor accuracy and efficiency of methods to bridge human-computer interaction gap and make it as natural as human-human interactions. Such methods will have a broad range of applicability in all aspects of life in a modern society from gaming and robotics to medical diagnosis and rehabilitation tasks. Considering recent progress of computer vision field, there has been an increasing urge upon medical domain. Computer-aided rehabilitation technologies are therefore gaining popularity among medical fraternity and are targeting more health-care applications [265]. Employing Gesture recognition where human-computer interaction is indispensable, becomes one of the most favorable applications owing to its natural and intuitive quality.

Nowadays with a rapidly aging population in most of the societies, the number of people suffering from cognitive disorders is on the rise. However, the healthcare sector has been facing acute shortage of skilled manpower and resources, especially in cognitive domain. Regardless of modality of the diagnosis and treatment, most of the developing countries are suffering from the lack of specialists. For example, according to [71], in India, there is an acute shortage of doctors, nurses and healthcare workers in various domains. The situation is not different in developed countries. Based on [27] Singapore, a country with high welfare level and social services, encounters seriously with a shortage of specialists (specialist-to-population ratio 1:1740). Accordingly, automatic and computer-aided diagnosis and rehabilitation technologies are becoming more accepted by medical teams. Therefore, contributions in this domain would have a significant impact on the society in general and quality of life of elderly people in particular.

Cognitive disorders such as Alzheimer's disease (AD) are prevalent among older

adults. Studies show a maximum correlation between AD and limb apraxia in all phases of the disease [36]. One of the effective tests which has been developed to diagnose these disorders is the Praxis test. Praxis is defined as the ability to plan and perform skilled movements in a non-paralytic limb based on the previously learned complex representations. Accordingly, limb apraxia is inability to carry out a learned motor act on command while there is no motor or sensory deficit in the subject [36, 81]. According to Geshwind's "disconnection model", apraxia is considered as failure (spatial or temporal error or failing to respond) of a subject to respond correctly with the limbs to a verbal command or having difficulty to imitate an action after being performed by an examiner [32]. Based on the American Psychiatric Association's report, Praxis test is accepted as diagnostically indicative sign of cortical pathologies such as AD [6]. However, the test is frequently neglected by clinicians despite being uncomplicated, straightforward and reliable estimate of the AD [177]. The clinicians skip the test mainly because: The whole process of the classical test takes longer time to conduct and even the developed countries face an acute shortage of well-trained specialists capable of performing and evaluating the test. Moreover, instruction of the test is not standardized and accordingly, not objective enough. Clinical practice reveals that, even when the test is performed, two kinds of problems can occasionally be observed: first one is the error in demonstrating the gestures to the subjects by an examiner and second, the errors in assessing subject's performance. In addition, most of the clinicians rely on memory and attention assessments in cognitive assessment process because memory and attention are the most frequent impairments in neuro-degenerative disorders. However, some of these disorders, in addition to memory impairments, have specific gesture impairments that distinguish them from the others. In order to diagnose those disorders, it is very important to systematically perform the praxis assessment. Therefore, automatic solutions to address these problems by providing an standardized test can be considered as a significant contribution in the field.

To capture changes in elderlies' behavioral pattern and to classify their cognitive status (Alzheimer's disease - AD, mild cognitive impairment - MCI, healthy control - HC), there has been a lot of studies on patient monitoring and surveillance [12, 25, 185, 162] with a main focus on recognition of activities of daily living (ADLs) [7, 113]. The main goal of such frameworks is mostly to provide cost-efficient solutions for in-home or nursing homes monitoring. These systems try to alert the healthcare providers about a significant change in the ADL behavior pattern which may lead to cognitive impairment, falling of the patient or other health related changes. However, ADLs usually have a complex and highly-variable structure and need to be evaluated for a long period of time so as to be useful for clinicians to timely detect health deterioration and assess Behavioural and Psychological Symptoms of Dementia (BPSD) in subjects.

However, such systems can provide patients with an autonomous living condition at home.

Meanwhile, contact-based and various sensors for rehabilitation tasks [218, 215] have been developed and found practical applications such as post stroke recovery [106] and limb rehabilitation [225]. Having their own advantages and disadvantages, they have been mostly utilized in rehabilitation and not for assessment and diagnosis. The most prevailed field which has been applied for computer-assisted diagnosis is image processing. Machine learning algorithms fed with X-Ray, CT scan, MRI, retina images, *etc.*, which are de-noised, segmented, and represented, assist the clinicians with diagnosis or surgical planning through finding meaningful patterns [179]. While these methods provide valuable diagnostic information for surgical purposes, their need to use advanced hardware and to process huge datasets, which result in high cost for image interpretation, is a big drawback compared to cost-effective gesture recognition tasks. However, using gesture recognition to obtain an objective classification of a person's performance, particularly for medical diagnosis, still remains as a novel and largely unaddressed challenge for the research community.

6.3 Motivation

Regarding the above-mentioned discussions, we propose a gesture recognition method by paying special attention to the Praxis test. The aim is to develop a robust and efficient computer-vision-assisted method to automatize the test procedure and to carry out assessments that help clinicians to have a more reliable diagnosis by providing a standardized method of performing the evaluations and a detailed analysis of subjectâs performances. Consequently, we have collected a challenging dataset [1] composed of dynamic and static gestures provided by clinicians for the Praxis test (Figure 6.1). We also adopt a gesture recognition framework, using a deep convolutional neural network (CNN) [126] coupled with a Longshort-term-memory (LSTM) [83], that jointly performs gesture classification and fine grained gesture correctness evaluation. As a result, we report performance of the proposed method and comparisons with other developed methods. With the evaluations we provide strong evidence about superiority of our representation learning method over traditional approaches, ensuring that robust and reliable assessments are feasible.

[1] https://team.inria.fr/stars/praxis-dataset/

Figure 6.1: The collected dataset consists of selected gestures for Praxis test. There are two types of gestures in the dataset: dynamic (14 gestures) and static (15 gestures) gestures. The dynamics are the ones including movement during the time that gestures are performed. The dynamic gestures are indicated with red arrows indicating their motion direction. On the other hand, the static gestures include body part orientation and position configuration without any movement during an amount of time. In another taxonomy, the gestures are divided to: Abstract, Symbolic and Pantomimes (starting with "A", "S" and "P" respectively).

6.4 Background

Contact based hand gesture technologies for upper limbs rehabilitation are already in use in hospital and in-house environments with acceptable accuracy. However, design of these technologies comes with certain advantages and obvious limitations [39, 256]. For example, pattern recognition based prosthesis upper limb control in [5] obtained good results in controlled lab settings but it did not achieve anticipated results when it was tested in clinical real-world settings. While contact based systems achieved viable accuracy in different studies, their acceptability among users became restrained because of their dependency on experienced users. In order to be beneficial, the user needs to get accustomed to such devices. Being uncomfortable or even posing a health hazard are other disadvantages of these devices, as those are in physical contact with the users [199]. Because of their physical contact, mechanical sensor materials cause symptoms such as allergic skin reactions.

Other similar systems that have benefited from various modalities were also developed targeting full or body part rehabilitation [56]. Even virtual reality based methods have been tried for rehabilitation to recover patients from different disorders like for phantom limb pain [156] or recovering from chronic pain using serious gaming [198]. In a recent work [225], authors use a Leap motion sensor equipped with a gesture recognition algorithm to facilitate palm and finger rehabilitation. There are also other approaches which have been proposed in various domains but potentially can be adapted for rehabilitation and diagnosis contexts. For example [4, 190] try to evaluate choreography movements based on a gold-standard obtained from professional dancers. There are also lots of work that address the sign language recognition problem [34, 183, 134], where it may also require accurate reconstruction of hand shape. The challenge is to match the gestures with corresponding words and construct conforming sentences.

Recently human action recognition has drawn interest among computer vision researchers due to its potential to improve accuracy of video content analysis [224, 234, 235, 236]. Although vision based systems are more challenging to develop and complex in configuration, they are more favorable in long term because of their user-friendly nature. Previously, most of the vision-based action recognition were based on sparse or dense extraction of spatial or spatiotemporal hand-crafted features [133, 203, 250, 35]. These methods usually consist of a feature detection and extraction step followed by a feature encoding step. For feature detection the most popular methods are Harris3D [119] and Hessian3D [247] while, for feature description HOG-HOF [119], HOG3D [111] and extended version of SURF descriptor [247] have found popularity in

recent years. The most famous descriptor in recent times is improved dense trajectories [234] which reached state-of-the-art result on various datasets. However, it turned out that most of these methods are dataset-dependent and there is no all-embracing method that surpasses all the others [237]. Consequently, there is a growing interest in learning low- and mid-level features either in supervised or unsupervised ways.

Skeleton-based gesture and action recognition approaches have received lots of attention due to the immense popularity of Kinect-like sensors and their capability in body part detection. In many works [248, 230, 251, 164, 66], using skeleton and RGB-D cameras have shown advantages over methods using RGB videos by providing novel representation and well-crafted algorithms. The main challenges in skeleton-based methods other than noisy joint information and the occlusion problem are to deal with the high variability of gestures and movements, high dimensionality of the input and having different resolutions in temporal dimension (variable speed of gestures). Generally skeleton-based action recognition methods treat actions as a time series problem where body posture characteristics and dynamic of movements over time represent the actions [76]. A common approach for modeling the temporal dynamic of actions is using Hidden Markov Models (HMMs) or Temporal Pyramid (TP) models [137, 139]. While TP methods are restricted by the temporal windows size, HMMs face difficulty in finding the optimal temporal alignment of the sequences and the generative distribution in modeling long term contextual dependencies.

Late advancements in hardware development –particularly powerful GPUs– have been important in the revival of deep learning methods. Convolutional neural network architectures have become an effective tool for extracting high-level features and shown outstanding success in classification tasks [117, 97, 58, 59]. Recently, deep networks have also been adapted for hand [73, 167, 221] and body [43, 26] pose estimation and also gesture segmentation and recognition [249], achieving state-of-the-art results on ChaLearn gesture spotting challenge and also other challenging datasets. However, unconstrained training of complex neural network models requires a big amount of data. The most popular approaches to restrain the complexity of the model is to reduce the dimensionality of the input by applying smaller patch sizes or training the model in an unsupervised fashion [125, 11]. Conventional Recurrent Neural Network (RNNs) have also proved to learn the complex temporal dynamics of sequential data, first by mapping the data to a sequence of hidden layers, and then connect the hidden layers to outputs. Although RNNs have shown efficiency on speech recognition and text generation tasks, it has been shown that they have difficulty to learn long-term dynamics due to vanishing gradient problem. LSTMs provided a solution for this issue by allowing the model to keep information in hidden

layer when it is necessary and update the layers when it is required. Since LSTMs are not confined to fixed length inputs or outputs they are practical for gesture recognition from video sequences and have shown success when unified with CNN features [55, 10, 192]. In this work, in order to avoid difficulties of temporal alignment in HMMs and learning long temporal dependencies in RNNs, we use LSTMs for modeling long temporal dependencies of the gesture sequences. Differently from [55, 10], we do not use 3D convolutions nor we train the CNN and LSTM jointly.Thus, our approach resemble most to [192], although, differently from the latter, we design our pipeline to receive hand patches instead of whole images and perform feature fusion. This makes our model even more memory efficient than the previous ones since hand patches are much smaller than the whole scenes. In [192], regression is performed over pain scores. Differently, since we want to detect few incorrect frames in very long sequences, we face a highly imbalanced classification task for which we choose a weighted classification loss function.

Segmentation is another complicated aspect of recognition task. In order to be able to perform precise recognition, a detection and segmentation process of different actions should precede the recognition which is mostly neglected in current action recognition research. It happens very often that action recognition frameworks presume pre-segmented video sequences are already available [119, 146, 118], however, this is not the case in real-world settings. It is common to use spatio-temporal sliding windows or fixed-size clipping of long videos [121, 246] to localize actions in space and time. For example in [57] actions are detected in videos using a sliding window and then spatiotemporal interest point are extracted and recognition is done following BoW approach. This endeavor is emphasized in more recent works which some try to localize actions in space [95, 142] while others perform temporal detection [263]. In [9] they perform both temporal and spatial localization of actions. Since sliding window framework requires sequential process of the whole videos to examine multiple spatial and temporal windows and their overlap, they are computationally expensive. To delineate the actions within the videos, there are also unsupervised methods that directly learn action models from the whole data (videos) [28, 52, 65, 70, 86, 155]. Although lots of methods were proposed for rehabilitation purposes [218, 215], these methods have not been applied in cognitive assessment context to help improve reliability of medical diagnosis.

6.5 The Praxis Test and Cognitive Disorders

6.6 Recognition Framework

We define four methods we have applied to evaluate the dataset (Figure 6.2). Each path (indicated with separate boxes) learns its representation and performs gesture recognition

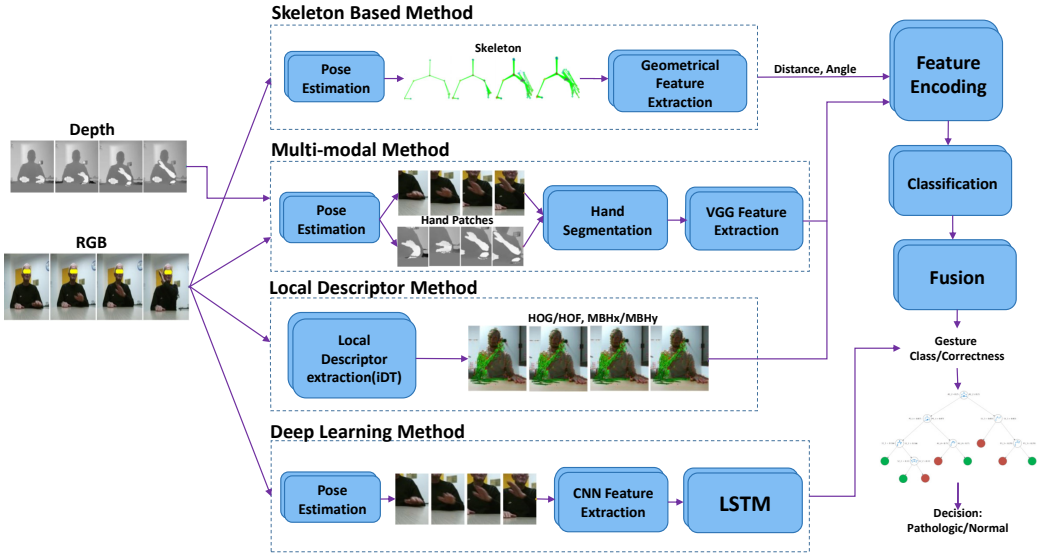


Figure 6.2: The data flow for the four methods applied on the Praxis dataset. The main components of each method are separated with dashed boxes.

independently given RGB-D stream and pose information as input. The skeleton based and local descriptor based methods are baseline methods and uses pipelines similar to the supervised architecture introduced in Chapter 3.

Skeleton Based Method: Similar to [257] the joint angle and distance features are used to define global appearance of the poses. Prior to the classification (different from [257]), a temporal window based method is employed to capture temporal dependencies among consecutive frames and to differentiate pose instances by notion of temporal proximity.

Multi-modal Fusion: The skeleton feature captures only global appearance of a person, while deep VGG features extracted from RGB video stream acquire additional information about hand shape and dynamics of the hand motion which is important for discriminating gestures, specially the ones with similar poses. Due to sub-optimal performance of immediate concatenation of the high-dimensional features of different types, a late fusion scheme for class probabilities is adopted.

Local Descriptor Based Method: Similar to action recognition techniques which use improved dense trajectories [236], a feature extraction step is followed by a fisher vector based encoding scheme.

Deep Learning based Method: Influenced by recent advancements in representation learning methods, a convolutional neural network based representation of hands is coupled with a LSTM to effectively learn both temporal dependencies and dynamics of the hand gestures. In order to make decisions about condition of a subject (normal vs pathologic) and perform a diagnostic prediction, a decision tree is trained by taking output of

gesture recognition task into account.

It should be noticed that for all of the developed methods we assumed that the subjects are in a sitting position in front of the camera where only upper-body of them are visible. In the following sub-sections, we explain each method in more details.

6.6.1 Articulated Pose Based Action Recognition (Skeleton-Based)

Current depth sensors provide 25 or fewer articulated skeleton joints through their associated middleware including 3D coordinates on an axis aligned with the depth sensor. However, in near-range applications where accurate joint information is required, whenever optimal range of the sensor was not respected, the joints could get missed or mis-detected or the extracted information is noisy. Given our task, most of the time almost half of the subject's body is occluded and the subjects are very close to the sensor and some body parts get even closer during performing of the gestures. This leads to missing or noisy part detections by the sensor. Instead of using unreliable joint information, we use CNN-based body part detector from RGB images in [186] which returns 14 body parts. For our purpose only 8 upper body part joints are relevant ($N_j = 8$): *right hand*, *right elbow*, *right shoulder*, *left shoulder*, *left elbow*, *left hand*, *chin* and *top of the head*.

We formulate a pose descriptor similar to [257]. Following them, first, we calculate pairwise joint distances and angles at each frame and then, to augment the characteristics of the final descriptor we describe spatial and temporal relations between consecutive poses similar to [216] and [129].

Pre-processing: We represent the skeleton as a tree structure where the chin node is considered as the root node. The joint coordinates are transformed according to the root coordinate in order to eliminate the influence of joint positions with respect to the sensor coordinates. Before representation, to reduce jitter in estimated joints trajectories we smooth joints position over temporal dimension by applying polynomial regression using weighted linear least squares and second degree polynomial model. Each subject performs similar gestures with variable speed resulting in variable frame sizes and joint trajectories. To achieve uniform performance speed along temporal dimension and to remove outliers in joints trajectories, once the smoothed joint positions are obtained, cubic interpolation of the values at neighboring joints is applied in the respective dimensions. Furthermore, to remove abrupt movements of the hand and elbow joints that are neither part of the gesture nor a jitter, a threshold is set which results in more stable joint values. Additionally, for the gestures in which laterality is not important (the subject is free to perform the gesture with either hand), we assume right hand as the dominant hand (considering that most of the subjects are right-handed) to reduce intra-class variability. Therefore, in these classes

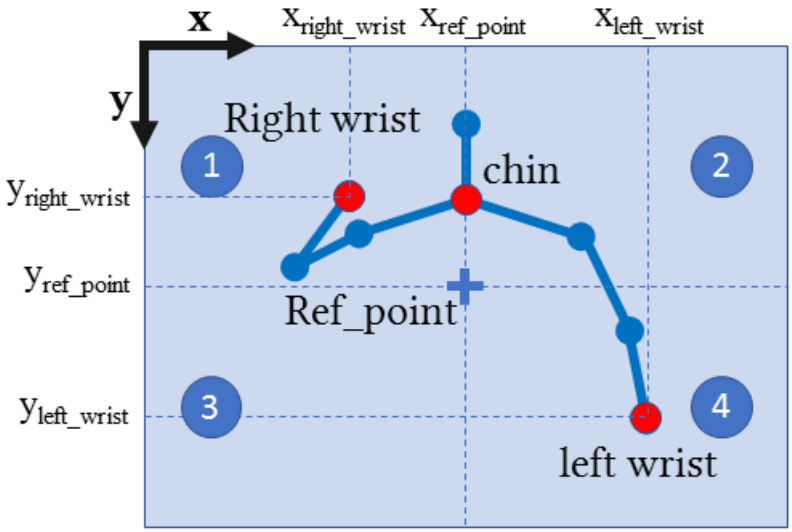


Figure 6.3: Dividing joint coordinates into four regions to detect the dominant hand in gesture performance

of gestures, we mirror the instances performed by left hand according to a vertical line through a reference point defined as:

$$ref\_point = [x_{chin}, (y_{chin} + (y_{rhand} + y_{lhand})/_2)/2] \tag{6.1}$$

To find the gestures performed by left hand, we divide the skeleton's coordinate into four regions by setting the center to the calculated reference point (Figure 6.3). Having the joint trajectories, we can decide handedness of the performed gesture. Moreover, to compensate variations in body size, shape and proportions, we follow method in [264]. Starting from the root node (chin), we iteratively normalize body segments between the joints to average bone size in the training data.

Feature Extraction: The feature extraction is carried out very similar to the one we explained in section 3.2.3 of chapter 3.

The skeleton sequences are encoded with this mechanism and are used to train a linear Support Vector Machines (SVM) classifier to recognize gestures.

6.6.2 Multi-Modal Fusion

Skeleton-based descriptors have shown good classification accuracy for action recognition tasks where entire body is involved in performing the actions. In case of our problem, other than relative body part positions and orientations, detailed hand pose and finger articulation are also essential for recognition task. Since skeleton joints do not provide such

detailed information, most of the gestures that can only be differentiated knowing subtle hand shape differences will not be recognized by a model that only relies on crude spatial information. We exploit depth data stream along with RGB images, first, to segment hand from the rest of body parts and then, to retrieve highly representative features only from the bounding-boxes surrounding the segmented hand (Figure 6.4).

Hand Segmentation: Since working directly with input image and depth data from Kinect is computationally demanding, we use cropped patches around hands using skeleton joint information. First of all, using the depth and RGB camera intrinsics and their extrinsic relations, the depth data are registered on RGB images (Figure 6.4a). Having depth and RGB registered, the hand skeleton joint is used for cropping the patches from the depth images. Accordingly, one big (160×160 pixels) and one smaller (80×80 pixels) square patches around the hand joints are cropped (Figure 6.4b). For the depth images we only take the bigger patches which are Z-normalized. Later, we cluster the gray-level values in depth patches (to obtain hand blobs) using multi-level image thresholding by Otsu's method [173] which obtains the thresholds based on the aggregated histograms to quantize images (Figure 6.4c). To detect the blob which most likely is the hand blob, we calculate the overlapping ratio of the blobs with the small patches' regions (Figure 6.4d). The blob with the maximum overlap is selected as the hand blob. Finally, this hand blob is used to define the segmented hand bounding-box in RGB images (Figure 6.4e).

Feature Extraction: Since CNNs have shown impressive results on various classification tasks, instead of hand-crafted image features, we use a pre-trained CNN model [211] (VGG-19) which is trained on a subset of the ImageNet [53] database to extract deep features from the retrieved RGB bounding-boxes. The model is trained on more than a million of images on a wide range of image classes (1000 classes). There are 19 layers to learn weights from which 16 are convolutional layers and 3 are fully connected layers. To extract features, we use the patches as input to activate the convolutional layers and collect the features from the fully connected layer "fc7" of size 4096 for each image patch. These extracted features are used to train the SVM classifier to perform gesture classification task.

Fusion: To combine the two modalities (skeleton+VGG image features) we follow a late fusion scheme by applying a simple linear combination of the obtained probabilities in the classification phase. If F is the final feature vector of the given video sequence v, $p(l_v|F)$ gives the probability of the predicted label l_v for that sequence and is calculated as:

$$p(l_v|F) \propto \alpha \cdot p(l_s|F^s) + (1-\alpha) \cdot p(l_d|F^d) \tag{6.2}$$

where l_s and l_d are predicted labels of the given video and $p(l_s|F^s)$, $p(l_d|F^d)$ are the

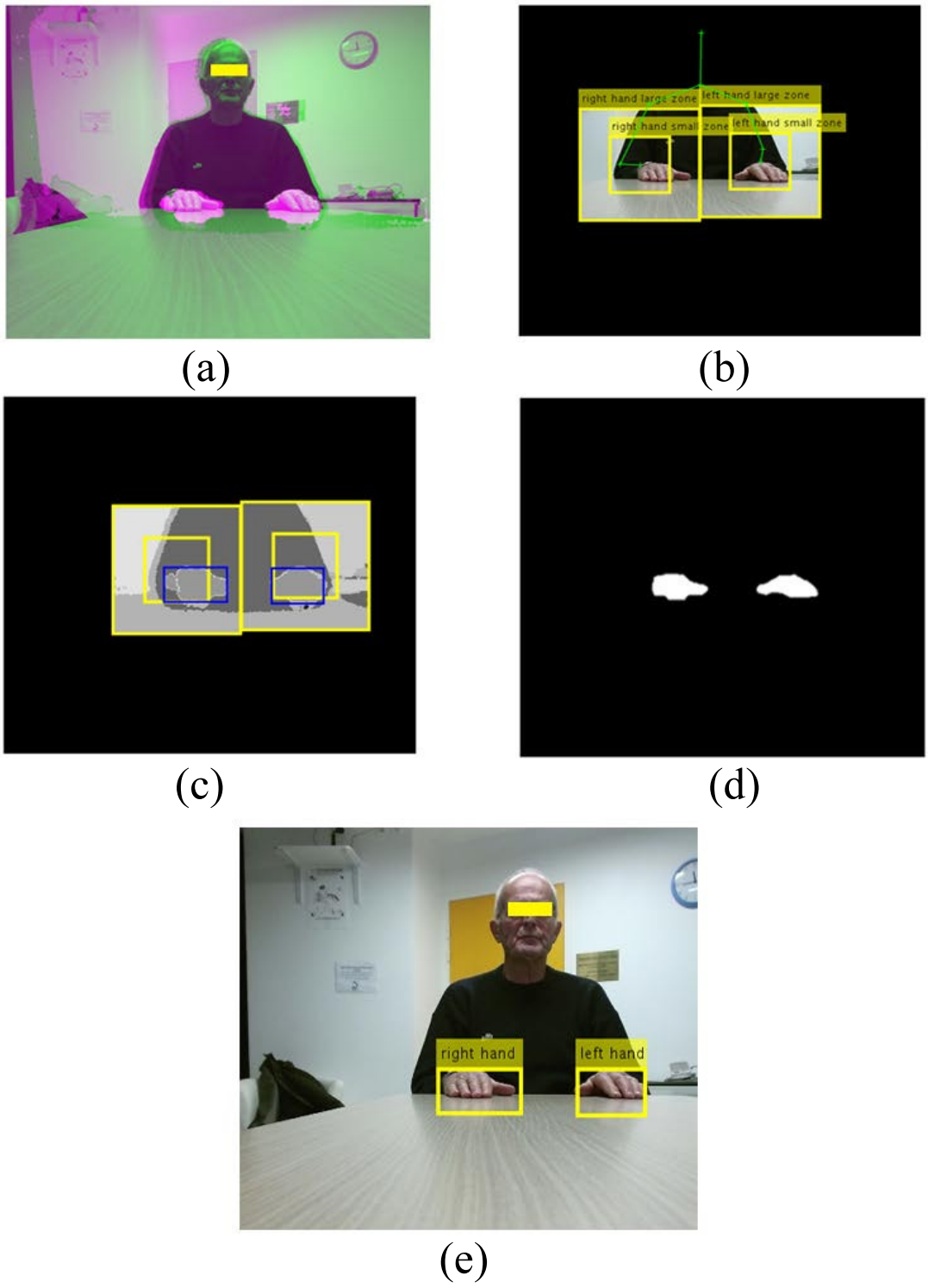


Figure 6.4: The steps of multi modal representation and recognition a) Registering depth image to align with RGB image b) Cropping the hand patches c) Clustering the depth values and detecting maximum overlap with the small patches d) Depth segmented hand blobs e) Register back accurate segmented hand blob on the RGB image and calculate bounding-box to extract image descriptors and fuse it with skeleton features.

probabilities of the skeleton and deep image patch descriptor modalities respectively. The coefficient α controls each modality's contribution which is set to 0.5 (through cross validation) indicating equal importance of the two modalities.

6.6.3 Descriptor Based Action Recognition

The same supervised method we introduced in chapter 3 is used in this section to produce the results.

Action Descriptor Extraction: We use improved dense trajectories (iDT) [236] to extract local spatio-temporal descriptors. Dense trajectories ensure coverage of whole dynamic of the gestures which results extraction of meaningful features. Length of trajectories are limited to $t = 5$ frames to capture slight motion in consecutive frames. Short trajectories are more reliable than long ones, especially when there is a gesture with fast irregular motion or when the trajectories are drifting. Moreover, short trajectories are suitable for short term gestures like the ones available in our dataset. Similar to [236], we choose a space-time volume (i.e. patch) of size $S \times S$ pixels and t frames around each trajectory. For each patch around the trajectories we compute the descriptor vector $\mathbb{X}$ consists of HOG/HOF and MBHx/MBHy local descriptors.

Action Representation by Fisher Vectors: The calculated descriptors are employed to create action representations based on Fisher vectors [180, 181]. Accordingly, first and second order statistics of a distribution of the feature set $\mathbb{X}$ are used for encoding a video sequence. Generative Fisher vector model is formed to model the features and the gradient of their likelihood are computed according to the model parameters (λ), *i.e.* $\Delta_\lambda \log p(\mathbb{X}|\lambda)$. The way the set of features deviates from their average distribution is depicted through a parametric generative model. To improve the learned distribution to further fit the observed data, a soft visual vocabulary is obtained by fitting a M-centroid Gaussian Mixture Model (GMM) into the training features within the preliminary learning stage:

$$p(x_i|\lambda) = \sum_{j=1}^{M} w_j g(x_i|\mu_j, \Sigma_j), \tag{6.3}$$

$$\textbf{s.t.} \quad \forall_j : w_j \geq 0, \quad \sum_{j=1}^{M} w_j = 1, \tag{6.4}$$

$$g(x_i|\mu_j, \Sigma_j) = \frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}} e^{-\frac{1}{2}(x_i-\mu_j)^T \Sigma_j^{-1}(x_i-\mu_j)}, \tag{6.5}$$

where $x_i \in \mathbb{X}$ represents a D-dimensional feature vector, $\{g(x_i|\mu_j, \Sigma_j)\}_{j=1}^{M}$ are the components of Gaussian densities and $\lambda = \{w_j, \mu_j, \Sigma_j\}_{j=1}^{M}$ are the parameters of the model: Respectively, $w_j \in \mathbb{R}_+$ is the mixture weights, $\mu_j \in \mathbb{R}^D$ is the mean vector, and $\Sigma_j \in \mathbb{R}^{D \times D}$

is the positive definite covariance matrices of each Gaussian component. The parameters λ are found using the Expectation Maximization restricting the covariance of the distribution to be diagonal. The GMM parameters are assessed through random sampling of a subset of $100,000$ features from the training set where the number of Gaussians is considered to be $M = 128$. Initialization of the GMM is performed ten times to obtain high precision and accordingly to provide the lowest error pertinent to the codebook. We define the soft assignment of descriptor x_i to the Gaussian j as a posteriori probability $\gamma(j|x_i, \lambda)$ for component j. Thereafter, the gradients of the j-th component can be calculated with respect to μ and σ ($G^{\mathbb{X}}_{\mu,j}$ and $G^{\mathbb{X}}_{\sigma,j}$). Finally, a set of local descriptors $\mathbb{X}$ as a concatenation of partial derivatives is encoded as a function of the mean $G^{\mathbb{X}}_{\mu,j}$ and standard deviation $G^{\mathbb{X}}_{\sigma,j}$ parameters for all M components:

$$V = [G^{\mathbb{X}}_{\mu,1}, G^{\mathbb{X}}_{\sigma,1}, ..., G^{\mathbb{X}}_{\mu,M}, G^{\mathbb{X}}_{\sigma,M}]^T. \tag{6.6}$$

The dimension of the Fisher vector representation is $2DM$. To perform action classification, linear SVM is employed. For multi-class classification, we implement the one-vs-all strategy.

6.6.4 Deep Learning Based Method

CNNs are a type of neural network architectures that are used for localization and extraction of local features in images in order to understand the visual content. Usually these networks are designed to deal with labeling of individual images. To upgrade the generated models to cope with the recognition problem in video data, temporal information should be taken into account. One way to use temporal information is to use the extracted features in the fully connected layer and feed to an Recurrent Neural Network (RNN). In such networks, neurons in addition to their connections to the next layer of the network, contain connections to themselves. This mechanism helps them to collect information from the previous inputs of the network. To train these networks, instead of backpropagation algorithm, an extension of it named Backpropagation Through Time (BPTT) [245] is usually employed.

Inspired by the recent advances on facial motion recognition [192], we propose to use a CNN to extract spatial static hand features, and learn their temporal variation by using Long Short-Term Memory (LSTM) [83]. LSTM is a variation of RNN with a special capability to learn long-term dependencies in sequential data. Intrinsically, RNNs are designed to learn long-term dependencies in sequential data. However, in practice, due to the problem of vanishing or exploding gradients [83], it is difficult to train these networks to keep long-term dependencies. Different from the original RNNs, LSTMs at each step

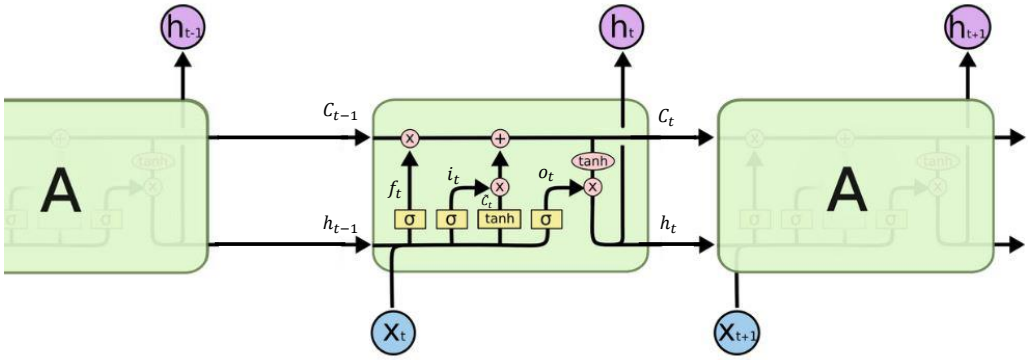


Figure 6.5: The repeating module in an LSTM contains four interacting layers.

have a cell state that is controlled by three different gates (input (i), output (o) and forget (f) gates). These gates decide that how much information can pass through the network (Figure 6.5). Each gate controls the amount of information that can pass via point-wise multiplication and sigmoid function. The output of the sigmoid function is in range of 0 and 1 which decides amount of information allowed to pass the gate. Input gate at each time-step is computed by the input of the LSTM at that time-step (x_t) and the hidden state of the previous step (h_{t-1}). If matrices W collect the weights of input and U collect the weights of the recurrent connections, the output of the forget gate f is computed as:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{6.7}$$

b is the bias vector parameter which is leaned during training. Then, to update the cell state, first the input gate layer in the current time-step (i_t) decides which values we need to update:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{6.8}$$

and second, a tanh layer calculates the candidate value $\widetilde{C_t}$ that can be combined with the input gate value to update the new state:

$$\widetilde{C_t} = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{6.9}$$

After that input gate and forget gate compute that which information should pass through the network and which one should get forgotten, by combining this information and the candidate state value $\widetilde{C_t}$ the new cell state C_t can be computed:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{6.10}$$

Finally, we can use the state to predict the output of the cell:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6.11}$$

$$h_t = o_t * tanh C_t \tag{6.12}$$

The output depends on the current cell state. Sigmoid layer computes which part of the state can go through and tanh function keeps the values between -1 and 1. The multiplication of tanh and sigmoid (output hidden state) makes sure that we output only the parts that are decided to. We used this model of LSTM RNN to predict the gesture correctness and also gesture class labels.

Differently from [192]:

1. the pipeline is modified to temporally align the patches from both hands,
2. domain-specific pre-training is used to extract better representations,
3. the LSTMs are adapted to cope with the static and dynamic gestures,
4. the output function is modified to be categorical instead of continuous,
5. a single prediction is performed instead of frame-wise predictions.

As a result, as it can be observed in Figure 6.6, the proposed pipeline is divided in three main stages: (i) hand patch extraction, (ii) CNN fine-tuning and feature extraction, and (iii) temporal aggregation with the LSTM. These three stages are next described in detail.

Hand Patch Extraction: Similar to the preprocessing steps in multi-modal method we extract body parts and using hand joints we extract image patches around both hands. In order to avoid the ambiguity in detecting the active hand, the same pre-processing step for flipping left and right hands in lateral gestures are also applied before sending the patches as input to the training network.

Hand Gesture CNN: In order to extract highly discriminative spatial features from the hand patches, we first fine-tune a CNN to classify the gesture and whether the gesture is correct or incorrect. For this purpose a GoogleNet architecture [217] is chosen since it has shown to provide competitive results while being lightweight compared to other models such as VGG [211]. Moreover following [174], we initialize the CNN with Deep Hand

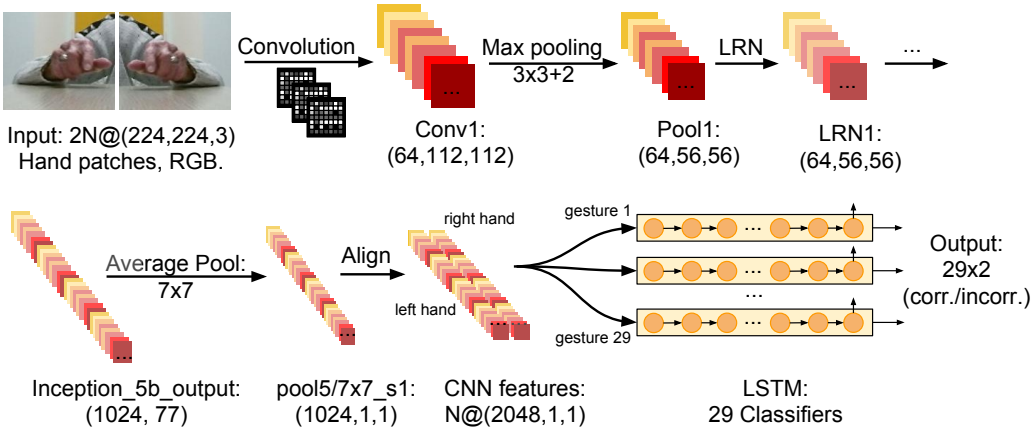


Figure 6.6: The proposed pipeline for hand configuration representation and gesture recognition. Spatial information is extracted from hand patches by feeding them to a CNN, and temporal information is leveraged using Long Short-Term Memory.

[112], a GoogleNet model trained with Expectation Maximization (EM) on approximately one million images to predict 60 different gestures.

Concretely, we reinitialize all the weights in the loss streams of the GoogleNet (GoogleNet has three classification layers), and fine-tune the network with the Praxis dataset (Section 6.7.1). In order to force the network to find highly discriminative features, the two output layers are reshaped to predict a probability distribution over 58 labels, where the first half corresponds to the 29 correctly-executed gestures, and the second half corresponds to their incorrect execution.

The hand gesture CNN is trained with Stochastic Gradient Descent (SGD) by minimizing the cross-entropy loss function using the Caffe Deep Learning Framework [98] during ten epochs, with a learning rate of 0.001 except for the reinitialized layers, for which is ten times higher. Standard data augmentation is performed by extracting random 224×224 sub-crops from the hand patches, and by randomly performing horizontal flips, *i.e.* randomly flipping the image crops along a central vertical axis following a *Bernoulli* distribution with $p = 0.5$.

After fine tuning, feature activation maps for the whole dataset are extracted from the last pooling layer. These feature vectors have a dimensionality of 1024. Once extracted, feature vectors from both hands in the same frame are concatenated, forming a 2048-dimensional feature vector. This concatenated vector is then fed to a LSTM, which is explained next, in order to leverage the temporal information present in the videos to make the final prediction.

Aggregating Temporal Information: Given a set of consecutive frames $F = \{f_1, ...f_n\}$ we

are interested in recognizing the gesture represented in those frames $p_g = p(gesture|F)$ and whether the gesture is correct or incorrect $p_c = p(correct|F)$. Hence, LSTMs are especially suited for this problem, since they are able to model long term dependencies by regulating the flow of information in the LSTM cell using their gates. Moreover, given that the gestures are executed at different speeds and thus, sequences have a variable number of frames, LSTMs fit perfectly to this problem since they allow information of variable-length sequences to aggregate.

Given the features of both hands extracted from the CNN that correspond to F, two independent LSTMs are trained by means of BPTT so as to model p_c (Probability of gesture performance correctness), and p_g (Probability of gesture class label) respectively. Note that p_c and p_g are separately trained since p_g is known at the test time. Thus, training p_c gesture-wise results in a higher performance since the model does not lose capacity to find the gesture class. To predict the correctness classification of each gesture an individual LSTM is trained for each gestures (29 LSTM with binary correct/incorrect output). However, for gesture class label prediction, one LSTM with 29 output is dedicated. Different from [192], where the Mean Squared Error (MSE) is minimized on each frame, the LSTMs used in this work are trained to minimize the cross-entropy error of single prediction on whole video sequences, thus zeroing out the output and gradients of intermediate frames. In order to overcome the bias towards correct predictions due to the data imbalance (In 26 gesture classes out of 29 classes, the number of correct performances are higher than incorrect performances.), the loss function for p_c was weighted to increase the sensitivity to the correct examples. Without weighting, we found that the model always predicted the majority class. Concretely, it is changed from:

$$loss(\mathbf{O}, c) = -\mathbf{O}_c + log(\sum_{j=1}^{M} e^{O[j]}), \tag{6.13}$$

where M is the number of samples, $\mathbf{O}$ is a 2-d vector containing p_c, and $c \in \{0, 1\}$ is the class label (incorrect, correct), to:

$$loss(\mathbf{O}, c) = (1 - p(c))(-o_c + log(\sum_{j=1}^{M} e^{o_j})). \tag{6.14}$$

Since $p(c)$ corresponds to the fraction of training video sequences labeled as c, and given that incorrect gesture sequences are underrepresented in the dataset, multiplying the loss by $1 - p(c)$ increases the penalty of misclassifying an incorrect gesture.

The LSTMs are trained with torch[2] using Adam [109] until they reach a plateau.

[2] torch.ch

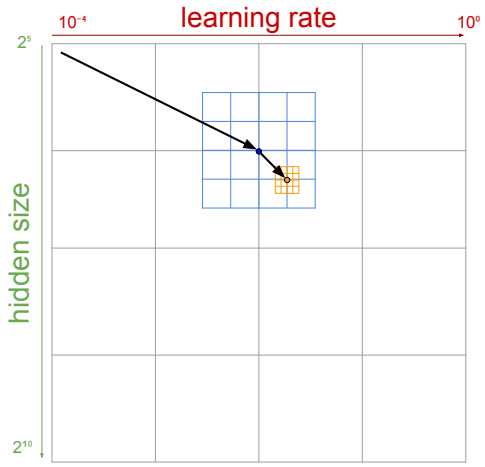


Figure 6.7: 2D gridsearch example. Best combinations are found iteratively from coarse to fine.

Weights are initialized by sampling from a uniform distribution $unif\{-0.8, 0.8\}$, and the network architecture and hyperparameters are chosen by gridsearch, see Figure 6.7 for an example.
In order to compare the diagnostic performance of LSTM classifier with clinician's decision, a decision tree is trained using outcome of gesture correctness test. The best pruning level of the decision tree is calculated with cross validation method. Therefore, the correctness results of a subject performing the gestures are exposed to the decision tree and resulted in a decision whether a subject is healthy or pathologic. Another decision tree is trained using ground-truth labels of gesture correctness test which is annotated by the clinicians. Comparison between the classification performance of the two decision trees interestingly shows how the LSTM classifier outperforms clinicians in diagnostic decisions based on a subject's performance. Accordingly, it develops an objective criteria by learning dynamics of the gestures globally in the whole dataset.

6.7 Experiments and analysis

6.7.1 Dataset

We collected a new challenging RGB-D upper-body gesture dataset recorded by Kinect v2. The dataset is unique in the sense that it addresses the Praxis test, however, it can be utilized to evaluate any other gesture recognition method. List of the gestures, their assigned ID and a short description about them is shown in table 6.1. Each video in the dataset contains all 29 gestures where each one is repeated for 2-3 times depending on the subject. If the subject performs the gesture correctly, based on decision of the clinician,

Category	Uni/Bimanual	ID	Type	Description	Similar gestures
Abstract	Unimanual	A1-1	Static	Left hand on left ear	A1-2, A1-3, A1-4, S1-1, S1-2, S1-5, P1-5
		A1-2	Static	Left hand on right ear	A1-1, A1-3, A1-4, S1-1, S1-2, S1-5, P1-5
		A1-3	Static	Right hand on right ear	A1-1, A1-2, A1-4, S1-1, S1-2, S1-5, P1-5
		A1-4	Static	Right hand on left ear	A1-1, A1-2, A1-3, S1-1, S1-2, S1-5, P1-5
		A1-5	Static	Index and baby finger on table	P1-3, P1-4, A2-2
	Bimanual	A2-1	Static	Stick together index and baby fingers	S2-1, S2-4, P2-1, A2-2, A2-5, A2-3, A2-4
		A2-2	Dynamic	Hands on table, twist toward body	P2-2, P1-4
		A2-3	Static	Bird	A2-1, A2-4, A2-5, S2-1, S2-4
		A2-4	Static	Diamond	A2-1, A2-3, A2-5, S2-1, S2-4
		A2-5	Static	ring together	A2-1, A2-3, A2-4, S2-1, S2-4
Symbolic	Unimanual	S1-1	Static	Do a military salute	A1-1, A1-2, A1-3, A1-4, S1-2, S1-4, P1-1, P1-3
		S1-2	Static	Ask for silence	A1-1, A1-2, A1-3, A1-4, S1-1, S1-4, P1-1, P1-3, P1-5, S1-3
		S1-3	Static	Show something smells bad	S1-2, S1-5, S2-4, P1-2, P1-5
		S1-4	Dynamic	Tell someone is crazy	P1-1, P1-3, A1-1, A1-2, A1-3, A1-4
		S1-5	Dynamic	Blow a kiss	S1-2, S1-3, P1-5
	Bimanual	S2-1	Dynamic	Twiddle your thumbs	S2-4, P2-1, A2-5
		S2-2	Static	Indicate there is unbearable noise	S2-3, S2-4, P2-4, P1-1
		S2-3	Static	Indicate you want to sleep	S2-2, S1-1, S2-4, A1-1, A1-2, A1-3, A1-4
		S2-4	Static	Pray	S1-2, S1-3, S1-5, S2-3, A2-5
Pantomime	Unimanual	P1-1	Dynamic	Comb hair	S1-1, S1-4, P1-3, A1-1, A1-2, A1-3, A1-4
		P1-2	Dynamic	Drink a glass of water	S1-2, S1-3, S1-5, P1-5
		P1-3	Dynamic	Answer the phone	P1-1, S1-1, S1-4, A1-1, A1-2, A1-3, A1-4
		P1-4	Dynamic	Pick up a needle	P2-1, P2-3
		P1-5	Dynamic	Smoke a cigarette	P1-2, S1-2, S1-3, S1-5
	Bimanual	P2-1	Dynamic	Unscrew a stopper	S2-1, P2-5, A2-5, P2-4
		P2-2	Dynamic	Play piano	P2-5, A2-2
		P2-3	Dynamic	Hammer a nail	P1-4, P2-5, P2-4
		P2-4	Dynamic	Tear up a paper	P2-3, P2-1, P2-5
		P2-5	Dynamic	Strike a match	P2-1, P2-3, P2-4

Table 6.1: List of the available gestures in the dataset and corresponding information.

the avatar continues the experiment with the next gesture, otherwise, they repeat it for 1-2 more times. Using the new Kinect v2 we recorded the videos with resolution of RGB: 960×540, depth: 512×424 without human skeleton information. The videos are recorded continuously for each subject. The dataset has a total length of about 830 minutes (with average of 12.7 minutes for each subject).

We ask 60 subjects to perform the gestures in the gesture set. From the subjects, 29 were elderly with normal cognitive functionality, 2 amnestic MCI, 7 unspecified MCI, 2 vascular dementia, 10 mixed dementia, 6 Alzheimer patients, 1 posterior cortical atrophy and 1 corticobasal degeneration. There are also 2 patients with severe cognitive impairment (SCI). We didn't use the two SCI patients' videos in the experiment since their performances were erratic and noisy and not useful for current study. However, we kept them in the dataset for further studies.

All of the videos are recorded in office environment with fixed position of the camera while subjects sit behind a table where only their upper body is visible. The dataset is composed of fully annotated 29 types of gesture (14 dynamic, 15 static). All of the gestures are recorded with fixed ordering, though the repetition of each gesture could be different. There is no time limitation for each gesture which makes the participants to finish their performance naturally. Laterality is important for some of the gestures. Therefore, if these gestures are performed with the opposite hand, those are labeled as "incorrect" by the clinician. To standardize the test procedure, a 3D animated avatar administrates the experiments (Figure 6.8). First, she starts with performing each gesture

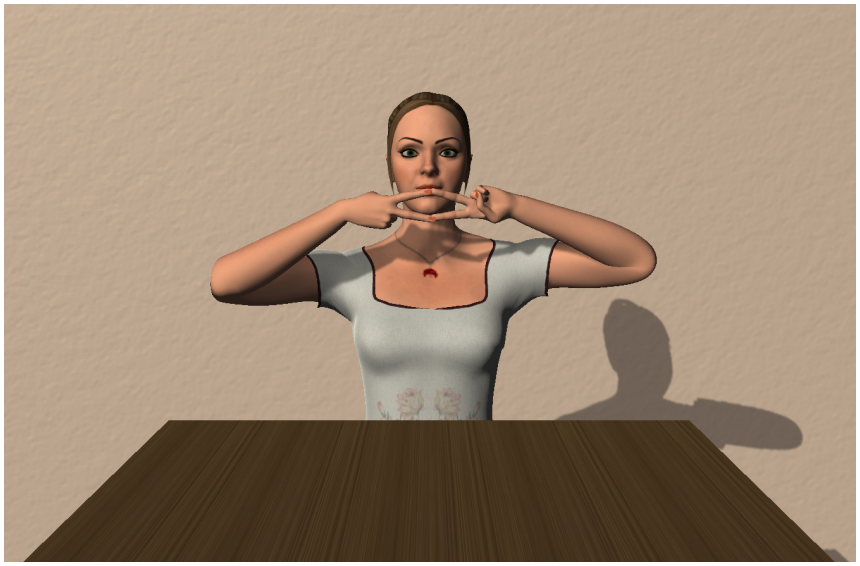


Figure 6.8: The virtual avatar guides the patients in a virtual environment.

by precisely explaining and giving instructions about how the participant should perform it. Next, she asks the participant to perform the gesture by sending a "Go" signal. The gestures are also divided into three main categories: Abstract, Symbolic and Pantomime gestures abbreviated by A, S, and P, respectively (Figure 6.1).

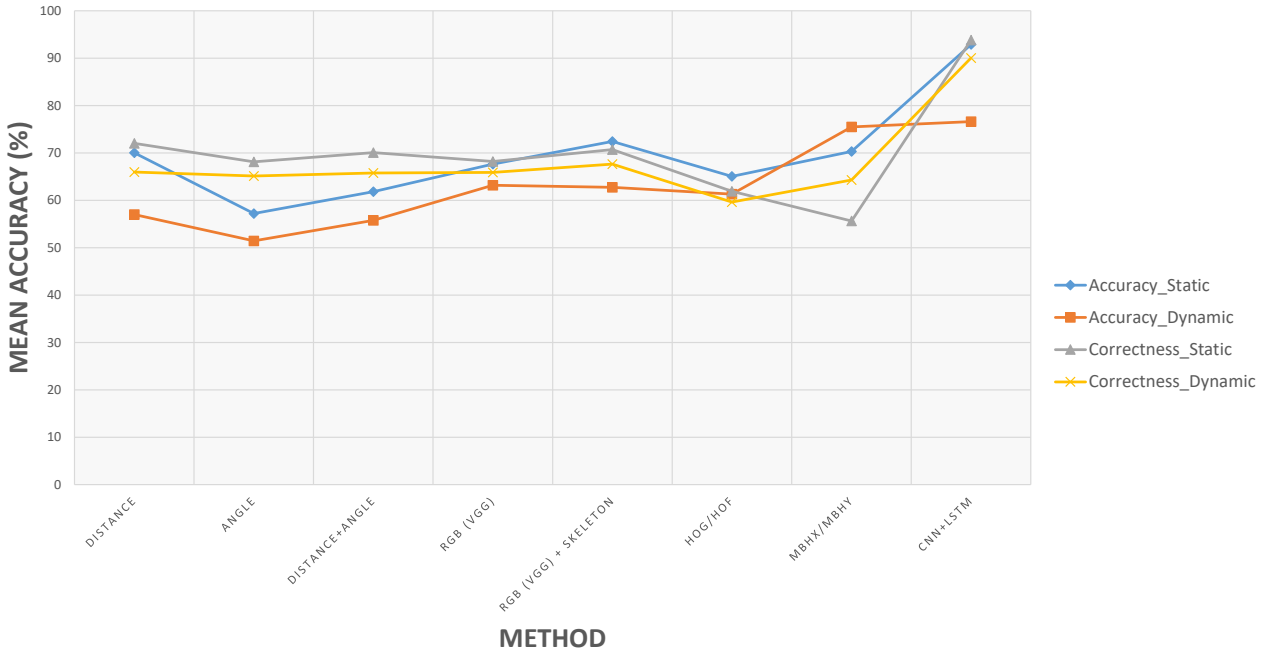
Although the dataset was collected using the same setting for all of the subjects, it is still challenging because of the selected gestures and the subjects who are real cognitive patients coming to memory center. For some of the gestures in the dataset only hand pose differs but the whole body part configuration and gesture dynamics are very similar as shown in Figure 6.9.

The main focus in the dataset is on two tasks: "gesture recognition" which consists in learning to recognize gestures from several instances of each category performed by different subjects and "correctness of performance" which is the evaluation of gestures based on quality of performance by each subject. The second task is more challenging since the "correctness" is subjective and depends on the professional opinion of the clinician and is not obvious all the times. The dataset is now publicly available for research community to bring more contributions on this task.

For the experiments we follow three-folds cross validation protocol, in which we divide the dataset into three nearly balanced subsets (patients 1-16, 17-37, and 38-58) . At each fold we run the training with the videos in the current fold and we use the two other subsets for validation and monitoring of training performance and also hyper-parameters optimization and finally testing.

Figure 6.9: Examples of challenging cases in Praxis gesture dataset. Some of the gestures are very similar in upper-body and arm movement and only differs in hand pose (a) and (b). Almost half of the gestures require both hands to perform e.g. (c, g). Some dynamic gestures are very similar and just differ in speed and range (c, d). Performer variation in upper body dynamics: some of the subjects keep their upper-body steady, while the others aim toward the camera (g, h). For some other gestures, dynamic of the gesture differs totally from subject to subject where some subjects gesticulate more (e, f). In some gestures subtle hand movements make the difference between correct and incorrect performances which makes the recognition task very challenging (i and k are incorrect samples of gestures in j and l, respectively).

Table 6.2: Comparison of the obtained results using proposed method in terms of accuracy of gesture classification and correctness of performance with other methods. The plot shows each method with respect to their average performance accuracy.



		Accuracy			Correctness		
Method		Static	Dynamic	Average	Static	Dynamic	Average
Skeleton	Distance	70.04	56.99	63.51	72.04	59.93	65.98
	Angle	57.21	51.44	54.32	68.13	62.16	65.14
	Distance+Angle	61.83	55.78	58.80	70.06	61.49	65.77
Multimodal Fusion	RGB (VGG)	67.63	63.18	65.40	68.21	63.54	65.87
	RGB (VGG)+Skeleton	72.43	62.75	67.59	70.72	64.55	67.63
improved dense trajectories (iDT)	HOG/HOF	65.04	61.31	63.17	61.89	57.37	59.63
	MBHx/MBHy	70.32	75.49	72.90	55.63	72.93	64.28
Deep Learning	CNN+LSTM	**92.88**	**76.61**	**84.74**	**93.80**	**86.28**	**90.04**

6.7.2 Evaluation Metric

6.7.2.1 Mean Class Accuracy Metric

To evaluate a performance of the proposed methods we use Mean Class Accuracy metric. We define accuracy metric for selected class *c* as:

$$accuracy_c = \frac{1}{num_{samples}} \sum_{i=1}^{num_{samples}} 1(y_i = c) \tag{6.15}$$

where y_i is the label assigned to sample i, 1(x) is the indicator function. Then, the Mean Class Accuracy metric can be defined as:

$$Acc = \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} accuracy_c \tag{6.16}$$

where C is set of action classes. Due to random factors such as random initialization we run each experiment two time and then we calculate the Mean Class Accuracy in order to make the comparisons fair and possible.

6.7.3 Results and Discussion

In this work we made a stride towards non-invasive detection of cognitive disorders by means of our novel dataset and an effective deep learning pipeline that takes into account temporal variations, achieving 90% average accuracy on classifying gestures for diagnosis. The performance measurements of the applied algorithms are given in table 6.2. In both tasks (gesture and correctness classification) concatenated dense trajectory based local descriptors performs relatively better than the others (except CNN+LSTM), especially, in dynamic gesture category. Particularly in gesture classification of dynamic gestures its performance is almost identical to CNN+LSTM approach. One possible explanation is that MBH descriptors are good in encoding motion pattern and since dynamic gestures include lots of motion they are capable of capturing them. This feature handle temporal information better than the others and reach the same level as the LSTM models. They perform poorly in correctness of static gestures since 60 to 70 percent of frames in static gestures are static gestures that do not contain any motion and the subject is in stable position in a specified gesture's key frame. CNN+LSTM does not perform good in dynamic gestures as good as static one, possibly because of the high variation in dynamic gestures. It is interesting to see that, by using distance feature in articulated skeleton based approach, we obtain competitive results compared to the others. We hypothesize that the good results are obtained due to the robust skeleton joint information and highly varied data in the dataset. However, this method performs poorly when it comes to dynamic gesture

classification. The reason for its poor performance might be lack of enough articulation in hand poses when we solely rely on the joint information specially in the gestures which upper-body configuration does not differ between gestures (e.g. Fig. 6.9 e, f). The results also demonstrate that the combination of both modalities (skeleton with image patches) is more robust and reduces confusion as shown by increase in the recognition rate of gesture classification of static category and correctness of static and dynamic categories.

As can be observed the proposed method (CNN + LSTM) outperforms all the methods in all of the tasks. It is important to note that these results are obtained by using gesture-wise LSTMs on hand patch data extracted from a CNN trained for classifying correctness and gesture simultaneously. Hence, since the task performed by the CNN is harder, it has to learn more discriminative features which then could be used by the LSTMs to better classify the video sequences. The existence of static and dynamic gestures did also condition the decision of using individual LSTM classifiers since 1 layer and 32 hidden units sufficed for most of the static sequences while the dynamic sequences needed up to 6 layers and 256 hidden units. This was expected since LSTMs that classified dynamic gestures had to model complex temporal relationships while the static gesture LSTMs needed only to find the exact frame where the gesture was performed and apply a linear classifier on the frame CNN features. Additionally, the fact that the LSTMs were trained gesture-wise allowed us to use sequences from other similar gestures as negative samples during training. It is interesting to see how our representation learning method outperforms all of the hand-crafted feature methods' performance. It is unlikely that having more data will improve hand-crafted methods' performance. However, it is highly expected that as more training data become available, the representation learning approach will achieve even more accuracy and better suited for independent settings.

The confusion matrices in figure 6.10 and 6.11 illustrate the behavior of our CNN+LSTM method in gesture classification task. The superior performance of the classifier in static gestures classification is immediately apparent. It can be noticed that some gestures are easily classified. This is the case for gesture $A1\_2$ that is always classified correctly and its highest false positive (FP) belongs to the class $S1\_3$ whose arm configurations during the static frames are identical. In dynamic gestures there are more confusions which most of them are because of resemblance in body and arm configurations and also variations coming from performer that gesticulate more or does extra arbitrary motions. The clearest example of this confusion is between gesture $P2\_4$ and $P2\_5$ (figure 6.9) where the pantomime gesture "tearing a paper" is very similar to "lighting a match" gesture and the only difference to separate the two is the speed of performing the gesture.

From clinician point of view fine-grained gesture classification is not important. What concerns them is evaluation of gesture correctness. They already know which gesture the

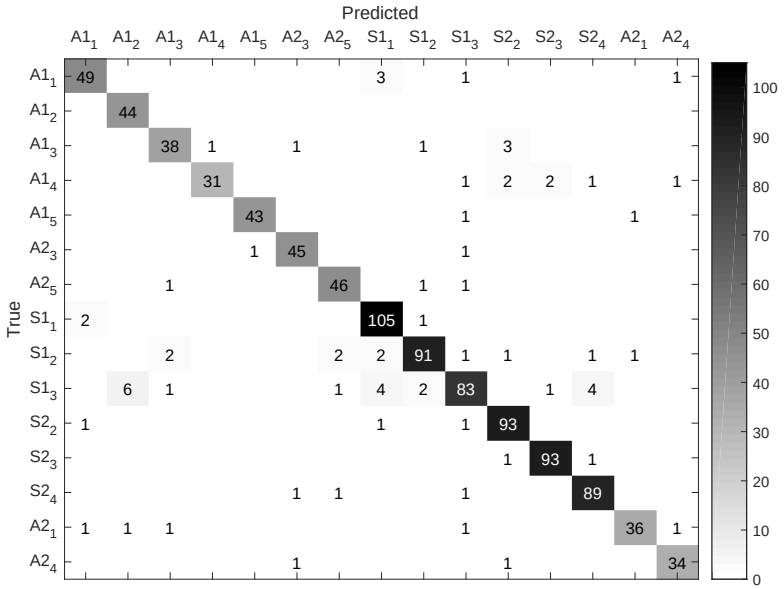


Figure 6.10: Static gestures

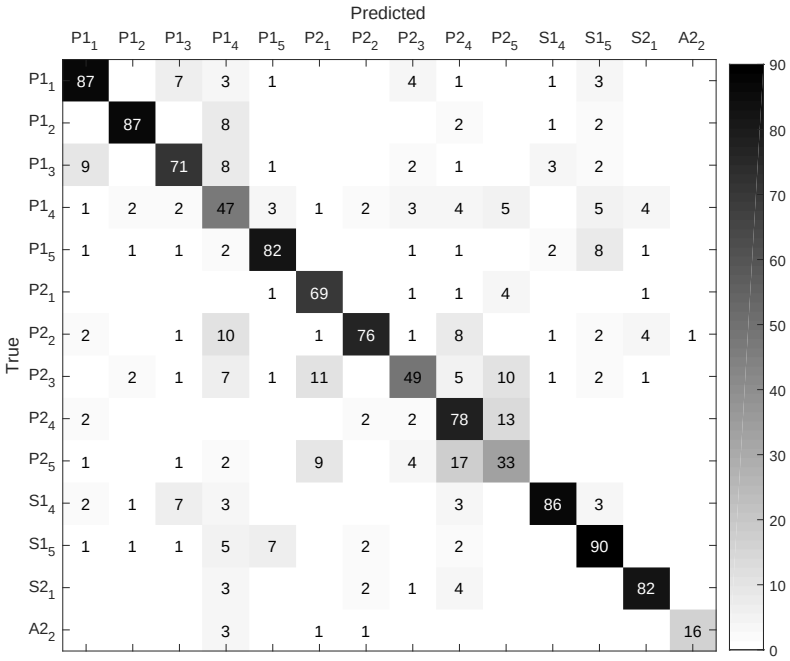


Figure 6.11: Dynamic gestures

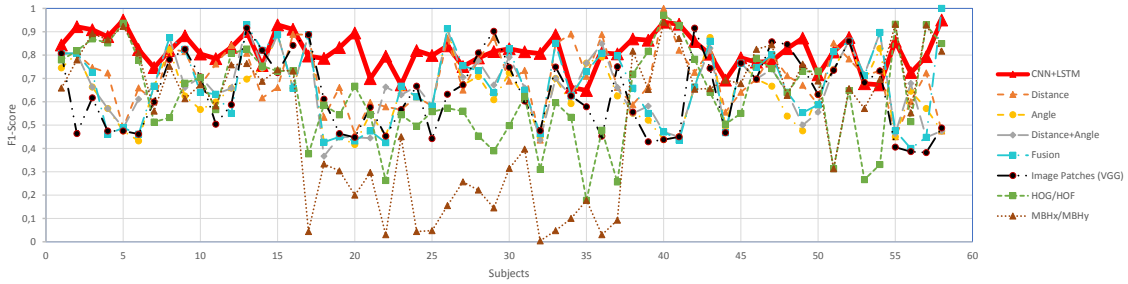
Figure 6.12: Confusion Matrices for the predicted gestures. The number in each element of the matrices indicates the number of predicted instances.

Static				
Gesture	Folds			
	1	2	3	Average
S1_1	1	0.952	1	0.984
S1_2	0.955	0.930	1	0.961
S1_3	0.906	0.925	1	0.943
S2_2	1	0.906	0.968	0.958
S2_3	0.978	1	1	0.992
S2_4	0.933	0.951	0.885	0.923
A1_1	1	1	1	1
A1_2	1	1	1	1
A1_3	0.968	1	1	0.989
A1_4	0.969	1	1	0.989
A1_5	0.903	0.900	1	0.934
A2_1	0.833	0.742	0.789	0.788
A2_3	0.870	0.851	0.900	0.874
A2_4	0.833	0.694	0.800	0.775
A2_5	0.923	0.920	1	0.947

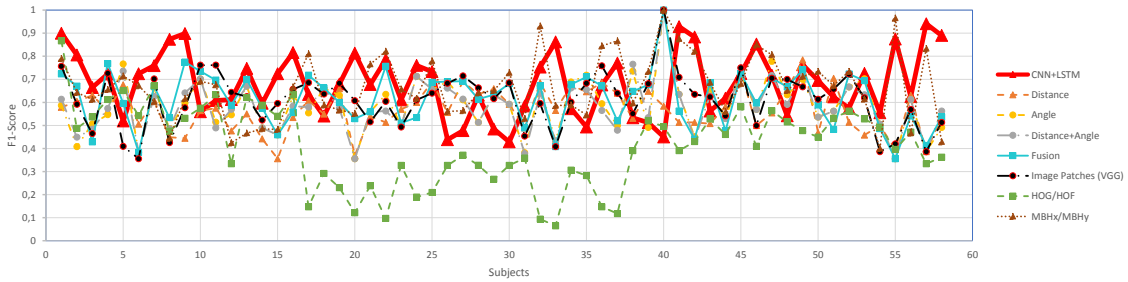
Table 6.3: Results in terms of correctness of performance for each fold in static gestures.

Dynamic				
Gesture	Folds			
	1	2	3	Average
S1_4	0.976	1	0.941	0.972
S1_5	0.891	1	1	0.963
S2_1	0.882	0.906	0.937	0.908
P1_1	0.895	0.854	0.968	0.906
P1_2	0.800	0.866	0.875	0.847
P1_3	0.730	0.888	0.937	0.852
P1_4	0.745	0.836	0.781	0.787
P1_5	0.869	0.880	0.968	0.906
P2_1	0.769	0.795	0.875	0.813
P2_2	0.857	0.906	1	0.921
P2_3	0.814	0.750	0.810	0.791
P2_4	0.869	0.880	0.777	0.842
P2_5	0.666	0.711	0.795	0.724
A2_2	0.846	0.794	0.880	0.840

Table 6.4: Results in terms of correctness of performance for each fold in dynamic gestures.



(a) Static gestures



(b) Dynamic gestures

Figure 6.13: The comparison of F1-scores with respect to subjects obtained by different methods for (a) static and (b) dynamic gestures. The proposed method (highlighted by red) shows better F1-score for most of the subjects and is less erratic compared to the others.

subject is asked to perform (class label) and what is important is to know if that specified gesture is carried out correctly or not. Tables 6.3 and 6.4 illustrate detailed gesture correctness evaluation at each fold on static and dynamic gestures respectively. For each gesture we achieve an acceptable accuracy that ensures robustness of the classifier which is very important for diagnosis task. Again it immediately becomes evident that the performance in static gestures (12 out of 15 class's accuracy is higher than 90%) category surpass dynamic category, although, there are more instances of dynamic gestures in the dataset and intuitively it is more likely for the classifier to learn the dynamics of these gestures. But it seems that complexity of these categories and nuances of gesture correctness of some of the gestures are too much to be learned with available number of trials. It also underlines the fact that handling temporal information is more difficult and even LSTM is not completely successful in modeling time. all This also gives a hint for clinical aspect of the work that the static category is more appropriate one and should contribute more in later data collections and more gesture classes of this category should be included in order to have more reliable evaluations. Capturing incorrect performances are of utmost importance that small nuance can affect accuracy of the diagnosis reports. This is because some gestures are simple enough for the subjects and most of the time are

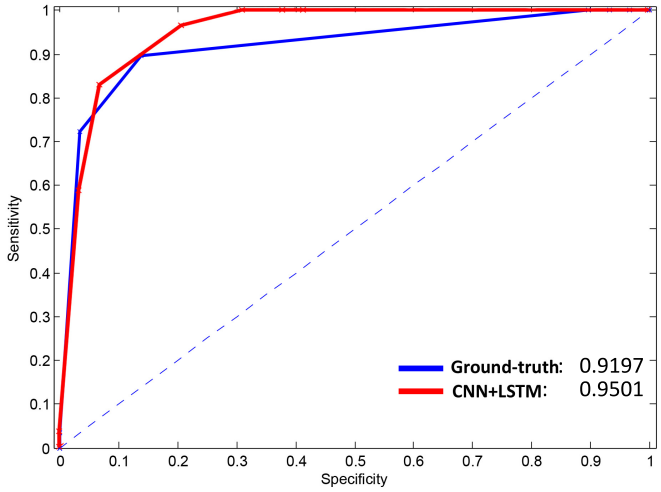


Figure 6.14: ROC of diagnostic classification using decision trees.

performed correctly while it is important and decisive to capture incorrect performances. This problem is rooted in unbalanced dataset where some classes have a few instances of incorrect performances. Although, the problem rectified somehow using similar gestures and employing the loss function, the nature of incorrect performances still remains undefined. Incorrect gestures could include anything and this makes these classes highly variable. Similar gestures stay far from real incorrect instances of a class and in some cases it might cause even more confusion. For example, we take gesture $P2\_2$ which is "playing piano" gesture as similar gesture for abstract gesture $A2\_2$ but in practice when a patient performs $P2\_2$ incorrectly, the incorrect performance is very close to $P2\_2$ and far from $A2\_2$. Moreover, in practice there are some subject specific redundant movements. For example, some subjects have specific mannerism and repeat it sporadically (one subject fixes his glasses before every performance and another one aims towards the examiners and asks questions). Although these subjects perform the gestures correctly but these additional movements hinder the proper evaluation. Ideally these subject specific movements could be learned and filtered out during pre-processing phase. In order to show the effectiveness of the proposed approach on evaluation of performance across individuals which is essential in terms of diagnosis, we conduct a comparative analysis using F1-score (figure 6.13). It can be observed that for most of the subjects CNN+LSTM surpass the other methods acquiring higher F1-score underlying that CNN+LSTM is more consistent and reliable as compared to the other methods specially when static gestures are taken into consideration. The highest F1-score fluctuations happen for subjects #15 to #40 where it can be verified that CNN+LSTM shows less fluctuations with an average score of 82% when compared to the others. Finally, to delve deeper into the details of

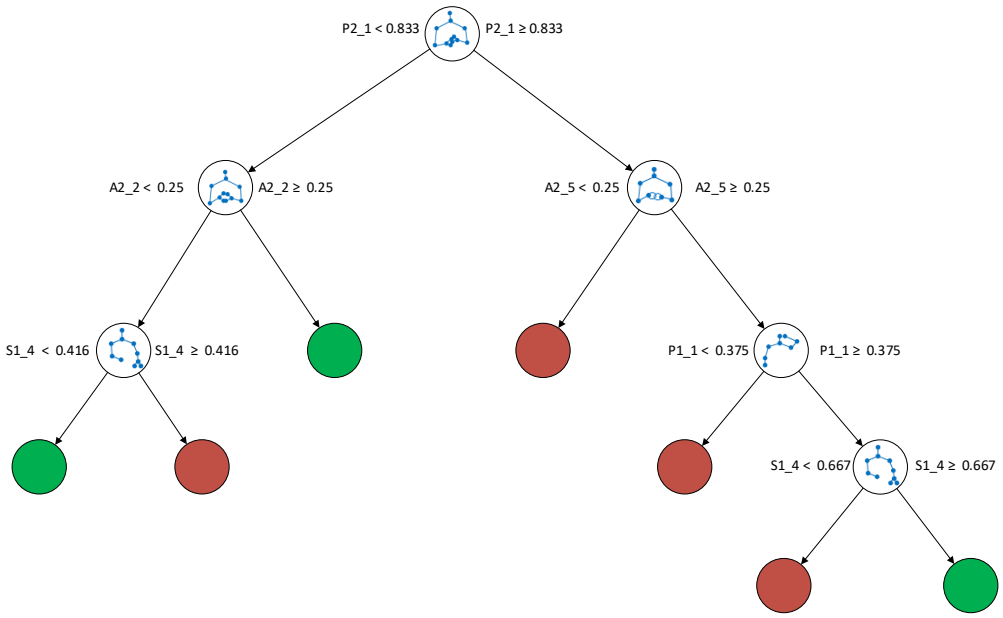


Figure 6.15: Ground-Truth

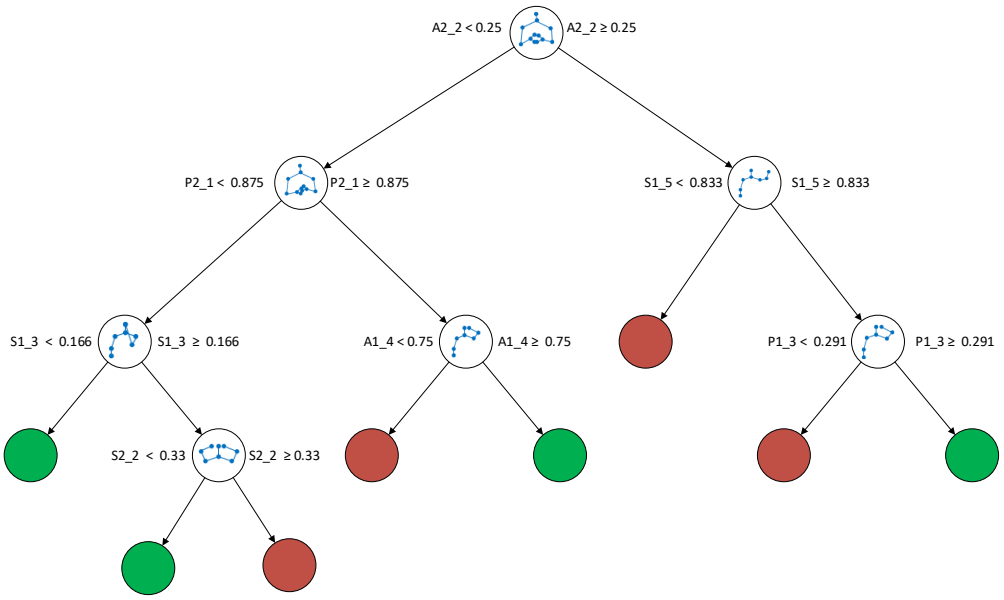


Figure 6.16: CNN+LSTM

Figure 6.17: Resulted trees illustrated using the trained decision tree classifier. Green leaves represents "Normal", while red leaves indicates "Pathologic" subject.

cognitive assessment of the subjects, we need to highlight the importance of the correctness classification of the gestures. As the classifier is only trained on correctness labels of the given instances, there is no immediate correlation between correctness of a gesture and condition of a subject. For example, a subject can perform one gesture correctly and the condition of the subject could be either normal or pathologic and therefore can not be inferred by relying on the correctness of that specific gesture. To ascertain the link between the correctness information of the gesture performances and the health status (healthy versus pathologic) of a subject, a pattern analysis needs to be carried out. Knowing knowledge discovery quality of decision trees and their high predictive performance, a tree model is trained given both overall performance of subjects on the gesture set and their condition as input. $F = \{f_i | i = 1 \dots 29\}$ is the normalized feature vector of a subject where f_i belongs to a gesture in the dataset showing the performance of the subject on that gesture. To verify the efficacy of the predictions obtained by the LSTM classifier, two feature vectors are created for each subject; one from ground-truth correctness values (labeled by clinicians) and the other one using correctness labels produced by the classifier. Then, the decision tree is trained to predict the condition of the subject whether it is normal or pathologic.

Figure 6.14 illustrates performance of the trained classifiers. Using the ground-truth labels, the decision tree can decide about condition of the subjects with 92% accuracy, whilst this rate is 95% when predictions related to the LSTM classifier are used. The accuracy difference of the two predictions (3%) is related to only two patients (Patients number 23 and 40). The low rate of discrepancy between the ground-truth and classifier's diagnostic predictions encourages that the objective assessment is achievable. This also implies that all the diagnostic information can not be mined only observing the gestures and the clinicians subjective opinions play an important role in providing final diagnoses.

The trained decision trees are depicted in figures 6.15 and 6.16. The most decisive gestures in diagnosis can be seen in nodes of the generated trees. Gestures $A2\_2$ and $P2\_1$ appear on root and first child node of both trees denoting their high impact contribution in diagnosis. Although it was observed that the accuracy of the classifications of the static gestures is higher than that in the dynamic gestures, the most important gestures appeared in the node of the trees belong to both categories (4 static and 6 dynamic). In total, there are 10 different gestures selected by the decision trees showing that an optimal subset of gestures and subsequently a shorter Praxis test consisted of lower number of gestures could be practiced. However, the trees are self-explanatory and very easy to follow and they are therefore comprehensible by the clinicians and even if it is required they can explain the performance of a subject and argue about the decision. Moreover, using the trees, a descriptive set of rules can be generated which explains what kind of performance would lead to an specific opinion. Further analysis can be carried out by applying different

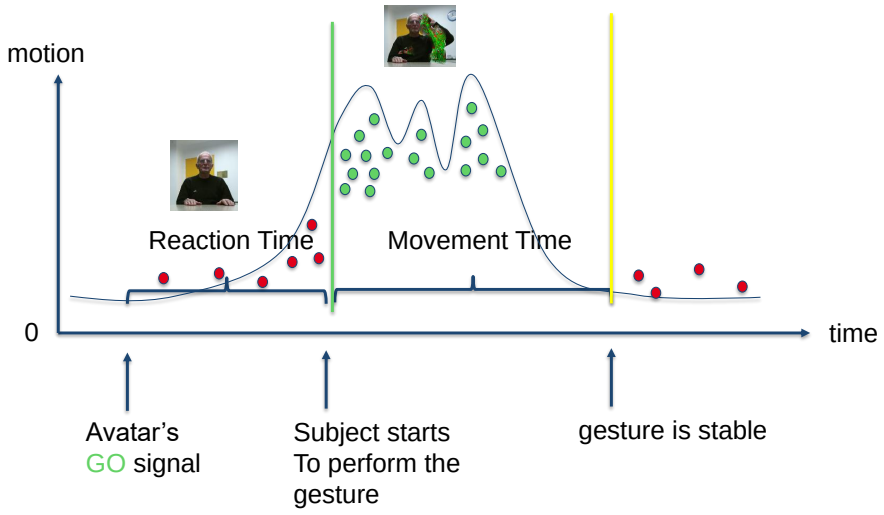


Figure 6.18: Shows the procedure of finding reaction and movement time using statistics of the extracted local descriptors (MBH). Green circles show dense scatter of the descriptors (Lots of motion), whilst, the red circles illustrate a sparse distribution (Indicating existence of less or no motion).

data mining techniques to interpret the results and this will be investigated in our future study.

6.8 Gesture Recognition Framework for Medical Analysis

To make the best use of the automatic assessments we need to deliver the obtained results and evaluations in an appropriate way to the clinician. Such presentation enables the clinicians to check the evaluations and analyze the assessments and make proper decisions regarding the patients. Therefore, we have developed a user-friendly gesture recognition tool including all of the extracted information from the data as well as additional useful information requested by the clinicians. This application provides doctors with detailed information about individuals and every gesture they perform, hence it helps doctors to have a thorough analysis of condition of a subject. The developed tool makes our method practical in real-world practices.

6.8.1 Reaction/Movement Time Detection

Other than correctness of the performed gestures, there are various parameters that plays an important role in diagnosis. We need to provide those required information to the clinicians: **Reaction Time** is the time duration between the instant clinician or

avatar gives the "GO" signal to the subject and the instant which subject actually starts to perform the gesture. Reaction time is also an important factor in diagnosis of cognitive disorders. It shows how responsive a subject is toward a signal. The reaction time could be a positive or a negative value. Some subjects do not wait for the signal and start to move their body parts performing gestures. In such cases the reaction time is negative because it happens before the "GO" signal. If the subject performs the requested gesture after the signal, the reaction time is positive. A negative value or a high positive value of the reaction time could be an indicator of a disorder. Because in both cases the subject is unable to pay enough attention to the given signals. **Movement Time** is the time duration between the reaction time instant and the instant that the performance is finished or the upper-body become stable (depending on the gesture type). The movement time is also an important factor for the clinicians in the analysis of a patient's cognitive status. A high value of movement time shows that the subject is unable to perform the gestures correctly and in a reasonable time.

To calculate these two values (Reaction and movement times), we use statistics of extracted motion descriptors. We use the local descriptors that we extracted previously for gesture classification (We get best results with MBH descriptors since these features are more sensitive to motion). But here, instead of representing a gesture with the descriptors, we count the number of descriptor to detect a change in the pattern of motion. As it is shown in figure 6.18 right after the "GO" signal is given by the avatar, a few descriptors are detected that show no change in the position of the subject's body parts. We chose these number as our reference. Reference data is chosen from the first w frames. We continue and check number of descriptors in a windows size of w and compare its containing descriptor number to the reference window's descriptor number. If no change is detected, we continue to the next window. The procedure continues until a big change occurs in the number of descriptors (bigger than a threshold learned from the training data). The time instant the big change occurs is chosen as the moment that the subject starts to perform the gesture, hence, is the moment of reaction. We use this instant to calculate the reaction time by finding its distance to the moment of the issued "GO" signal. The same procedure continues until a drastic increase in number of descriptors is observed. This means that either the performance is finished (in dynamic gestures) or the position of body is stable (in case of static gestures). We use this moment to calculate the movement time (6.18). This time is the difference between the reaction moment until the detected last movement moment.

According to the clinicians, the most important part of the performance in static gestures is the last part of performing the gesture. For example if the subject is asked to perform an abstract gesture such as creating a diamond shape with the hands, only the

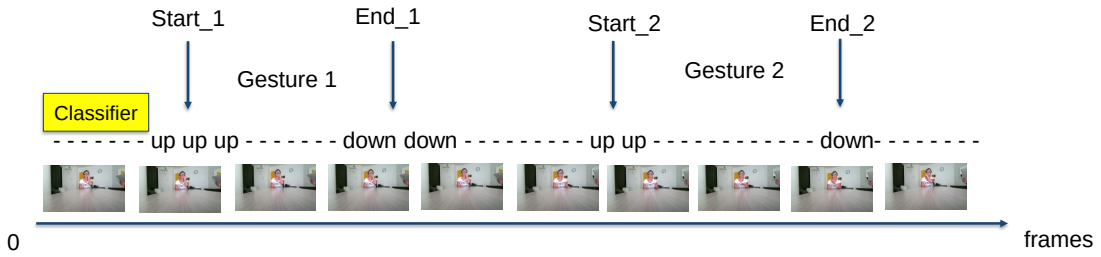


Figure 6.19: Shows the procedure of gesture spotting by finding beginning and ending time instant of gestures.

last part where the subjects thinks that the requested gesture is performed is important. The subject can try for a while until s/he manages to perform the gesture. However, when s/he thinks that the gesture (for example diamond shape) is performed, s/he stays for a while in this position until s/he put her/his hands back on the table. Based on the same principle for the detection of reaction/movement times, we can detect this crucial moment called the key-frame.

6.8.2 Key-Frame Detection

In static gestures we have two peaks of motion if we plot the motion descriptors like figure 6.18. One peak is at the beginning when the subject starts to perform the gesture and the next one is when the subject puts his/her hands back on the table. Most of the frames between the two detected peaks belongs to the time where the gesture is performed and contains the key-frames. Using the statistics of the descriptors we find the two peaks and we chose one of the frames between the peaks as the key-frame.

6.8.3 Gesture Spotting

The recorded dataset consists of gesture instances recorded in a continuous fashion. A video recorded from a subject contains all of the performances by that subject. Therefore, before evaluation of gestures (either gesture recognition or correctness evaluation), the gestures should be spotted in the long videos. We need to segment the videos by detecting beginning and ending of the gestures. To achieve this we follow a sliding windows approach. Our dataset is recorded in a controlled settings and we know that all the subjects starts their performances by their hands on the table. Later, when they finished performing the gesture, they put their hands back on the table. Knowing that, we train a classifier

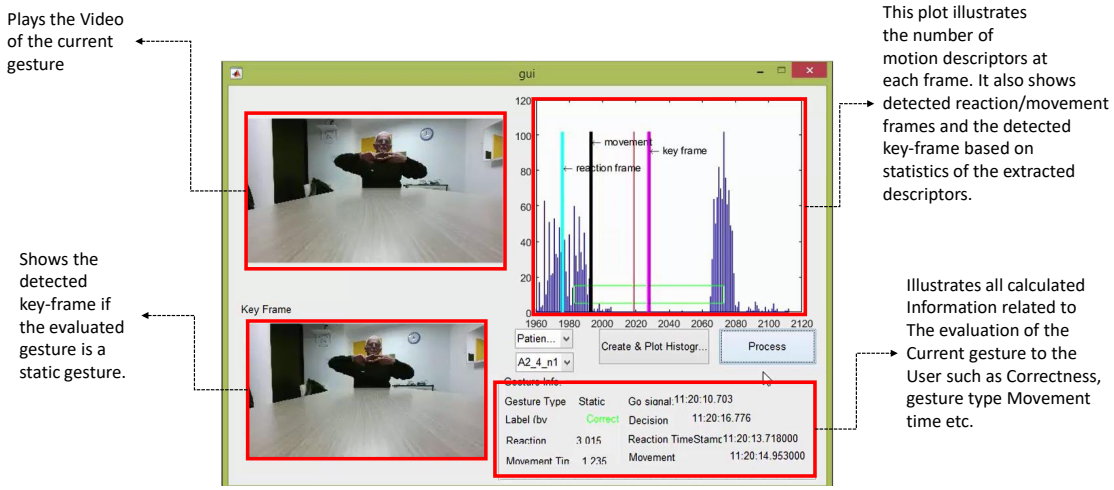


Figure 6.20: illustrates the user-interface of the application developed for gesture analysis. The clinicians can easily use the tool to analyze every gesture performed by each subject and receive useful information about them.

that only recognizes when a subject takes hands off the table or put them back on the table (The classifier is trained on two hands up and hands down classes and a background class). We have this prior knowledge that a "taking hands off the table" will be followed by a "Put hands back on the table". We slide the classifier through the videos to detect hands up/down frames. The long video is segmented to short clips of gesture instances using the spotted hands up and down frames (Figure 6.19).

6.8.4 The Application

Figure 6.20 illustrates the user-interface of the developed application. The application provides the clinicians with a comprehensive report about all aspects of the performed gestures. Using the methods suggested in this chapter extra information regarding the gesture performances are provided to the clinician that otherwise was not possible to obtain. Therefore, this information helps the clinicians to have detailed analysis about each patient and take reliable decisions.

6.9 Conclusion

Early diagnosis of cognitive impairments are essential to provide better treatment for older adults. Praxis test is accepted as diagnostically indicative sign of cortical pathologies such as AD. Despite being simple, straightforward and reliable estimate of the AD, the test is frequently ignored by clinicians. To avoid such situations which arise during this process, we propose a computer-assisted solution to undergo evaluation of automatic diagnosis process with help of computer vision. The evaluations of the system can be delivered to the clinicians for further assessment in decision making processes. We have collected a unique dataset from 60 subjects targeting analysis and recognition of the challenging gestures included in the Praxis test. To better evaluate the dataset we have applied different methods using different modalities. The algorithm based on geometrical features (Angle and distance) obtained competitive performance to the fusion models. Fusion of geometrical features does not improve the recognition performance because of low discriminative power of angle features that reduces the power of the aggregated descriptor. However, fusion of geometrical and VGG features always outperform the individual and combined geometrical features. The framework based on dense trajectory descriptors outperformed other baselines. In the evaluation of dynamic gestures, dense trajectory based approach showed competitive results to the deep learning framework. Experiments showed higher recognition rate for correctness and gesture label of static gestures. Using CNN+LSTM we have shown strong evidence that complex near range gesture and upper body recognition tasks have potential to be employed in medical scenarios. However, there is a big margin between the performance of static and dynamic gestures, suggesting that temporal information is still difficult to model even when LSTM RNNs are used. We have also developed a gesture evaluation application to provide the clinicians with the obtained results and evaluations. The clinicians can go through performance of individual subjects and analyze this performance by checking information of every gesture instance in the set. In order to be fully practical, the system must be evaluated with larger population. However, satisfactory feedback of clinicians from our preliminary evaluations is a promising commencement.

Chapter 7

Conclusion and Future Work

"Neither the human condition in particular nor our explanatory knowledge in general will ever be perfect, nor even approximately perfect. We shall always be at the beginning of infinity."

- David Deutsch

Contents

In this thesis we propose a comprehensive framework for human activity recognition. We localize the activities in untrimmed videos using unsupervised learning of the scene regions. The activities inside the scene regions are described through Hierarchical Activity Models that handle the spatial and temporal aspects of occurring activities. The hybrid framework takes benefits of supervised classifiers to label the generated models, while, unsupervised framework stores learned dictionaries to measure activity models similarities. Two supervised frameworks based on Bag-of-words and Fisher Vector encoding is also developed to produce the baseline evaluations. Our conducted experiments demonstrated that the proposed hybrid methods achieve state-of-the-art results compared to the supervised evaluated baselines. For more precise recognition of activities in a region, a comprehensive gesture recognition framework is developed in order to conduct fine grained recognition gestures and actions. Our gesture recognition framework based on deep neural networks outperforms the evaluated hand-crafted methods. This framework will be

integrated to the activity recognition framework to produce more precise activity models. We also provide an interactive user-friendly tool for clinicians enabling them to analyze conditions of patients and make the right diagnostic decision. We conclude our work pointing out key contributions (Section 7.1) and their limitations (Section 7.2). Finally, we discuss about the short-term and long-term perspectives (Section 7.3) emphasizing on the directions we will take in our future research in the field.

7.1 Key Contributions

Unsupervised scene modeling and activity discovery
We propose an unsupervised zone scene modeling using the trajectory information of moving agents in the scene. This way, we found the most interesting areas in the scene where the probability of occurring activities are higher. Additionally, unsupervised scene models combined with a knowledge-based hand-crafted model, enabled modifying (merge/split) of the scene models in order to find the optimal shape of the activity regions.

Dynamic length unsupervised video segmentation
Using the calculated zone scene models, we propose an unsupervised dynamic-length video segmentation method. Unlike sliding window methods or fixed-size video segmentation for detection of activities, scene model helped to discover activities (Beginning and ending of activities) happening in the scene. To break down long-term videos into smaller chunks, we use global motion information along with the boundaries of each region to detect enter/exit moments to the regions.

Hierarchical Activity Models
The scene models are used for extracting multiple abstraction layers out of global motion information. We used these layers to construct hierarchical tree-structured activity models.

Generating Hybrid Activity Models
The constructed hierarchical models are based on global motion information. In order to benefit from the local motion information, we propose a hybrid model that combines global motion information with higher level knowledge produced by a supervised classifier based on local descriptors. Empowered by the knowledge from supervised annotations, generated hybrid models are suitable for describing the observed activities in the scene. The hybrid framework outperforms supervised and unsupervised frameworks in our experiments.

Generating Unsupervised Activity Models

We also proposed an unsupervised framework based on hierarchical models. In this version, rather than training a supervised classifier with extracted descriptors, we use them to train a visual vocabulary (dictionary or codebook) for each descriptor type. The generated codebooks are combined with the hierarchical models, represented the final activity description models.

Zone Refinement with Hand-crafted Activity Models

To cope with inaccurate scene region shapes, we proposed a zone refinement method based on knowledge interaction. The scene region information learned from the training data in an unsupervised way is shared with an ontology-based hand-crafted model. If the defined regions do not comply with each other, a split or merge operation applied to rectify their spatial form. The knowledge interaction between the two methods continues in a loopy way until an optimal segmentation of regions is acquired.

Online Activity Recognition

The proposed frameworks are capable of performing online activity recognition, thanks to the automatically created scene models. During the test, the input video stream can be segmented by the scene regions provided by the scene model and feed into the recognition block of the framework.

Gesture Recognition

In this thesis, we present four gesture recognition frameworks working with one or combination of different modalities. The modalities include: RGB, depth map, and skeleton information. The frameworks utilize these data modalities to retrieve meaningful information for gesture classification. They targeted two classification challenges: *gesture classification* and *gesture correctness*. The second task is more challenging due to high intra-class variability of the performances. The proposed framework based on deep learning (CNN+LSTM) surpasses the other frameworks in both tasks.

Assessment of cognitive disorders and new PRAXIS dataset

Being informed by dementia experts, the Praxis gesture test is indicative of cognitive disorders such as Alzheimer's disease. The proposed gesture recognition frameworks are employed to assess cognitive disorders in senior adults. To evaluate the developed methods, we have collected a gesture dataset based on the Praxis test from 60 real dementia patients (830 minutes of video including RGB, depth, and skeleton information). Thanks to the annotation of gesture correctness by the clinicians, we introduced new challenges which were missing in the state-of-the-art datasets. Based on our evaluations, the proposed expert system outperforms human experts in the accuracy of diagnosis.

We have also developed an assessment tool with a user-friendly interface for the clinicians.

Evaluations on the recorded and public datasets
We provided extensive experiments on the recorded gesture dataset to evaluate our proposed gesture recognition methods. We also performed extensive evaluation of hand-crafted and deep-learning based activity recognition frameworks on daily-living activity recognition datasets.

7.2 Limitations

The proposed methods in this dissertation have some limitations. Some of these limitations can be investigated and resolved as an extension of current work in the near future. However, some limitations are intrinsic to the computer vision community and still are open problems. In this section we describe these limitations and the final section is dedicated to the discussion of short-term and long-term perspectives.

Unsupervised scene modeling and activity discovery
Modeling long-term activities requires even a deeper exploration of the topic. For example, in performing long-term daily activities, no temporal constraint is observed. Performing an activity can take a few seconds by a subject whilst the same activity takes minutes by another one. An unsupervised method estimating such temporal constraints (If there is any), can generate more precise activity models.

Dynamic length unsupervised video segmentation
Our segmentation relies on triggering enter and exit events from a scene region. In a more complex daily living scenario, this assumption can be misleading and results in a inaccurate video segmentation. For example a subject can sit on a chair (Inside "Chair" scene region) and perform first "Reading" and then change to "Drinking Coffee" activity. In this scenario our segmentation method will rely on enter/exit timestamps producing wrong delineation of activities.

Hierarchical Activity Models
In the proposed hierarchical model for describing activities, the nodes in the same resolution layers are assumed to be independent from each other. This assumption lacks the understanding of potential dependencies between the neighboring nodes. A more complex model accounting for such dependencies can results in better and more descriptive models.

Online Activity Recognition
Online recognition can be affected by the problem of inaccurate segmentation in more complex scenarios (Explained in the limitation of dynamic length segmentation). In some cases activities can change without being noticed by the system (using global information).

Evaluations of the frameworks
Complexity of evaluation is one of drawbacks of systems developed for understanding long-term activities. The evaluation procedure requires ground-truth annotation of long activities. The annotation process of such long videos in large quantity is a challenging task especially when a detailed labeling of sub-activities is required. Similarly, annotation of gesture performances for diagnostic purposes is also challenging. For medical diagnosis a precise objective opinion is required regarding the performances. To annotate data for such tasks, expert clinicians should contribute in the experiments.

Gesture Recognition
There are several gestures in the PRAXIS dataset (Especially static gestures) that configuration of the fingers are very important in assessing the correctness of the performances. A subtle change in the configuration of the fingers can change the opinion of the doctors from right to wrong. In evaluation of those gestures, the proposed models face difficulty in capturing detailed information regarding patterns of fingers. A more accurate model accounting for arrangement of fingers and their relationships between both hands can improve the accuracy of evaluations.

7.3 Future Work

7.3.1 Short-Term Perspective

The proposed activity recognition frameworks provides a complete evaluated pipeline for describing long-term videos. We believe that the designed frameworks are mature enough and they can be applied to real-world applications for analyzing indoor activities. Nevertheless, there is still plenty of room for improvement.

Activity modeling
To delve deeper in activity modeling a more precise temporal segmentation model for estimation of temporal constraints of activities should be explored. Currently, only global motion information is utilized to discover activities and when the target activity changes inside a region, the new activity is not discovered. Multi-resolution scene model handles this situation into some extend, however, a more precise segmentation technique

that relies more on salient changes in local descriptors rather than global should be considered. More research can shed light on this aspect.

In modeling the activities we used different types of descriptors from geometrical to deep features. However, we have not tried the feature combination or feature selection strategies. We could provide extended analysis showing effect of such strategies on the quality of video descriptions.

Other than different feature types, different data modalities can also be utilized. For example the collected data from different sensors such as velocity or audio can be appended to the feature set. The frameworks are designed in a way that any new feature can be easily plugged into the models. However, adding new type of feature should be carefully considered. Overloading the models with different features can hinder generating models with easy semantic description.

One major problem that makes evaluation of methods addressing long-term daily living activities difficult is lack of data. We are interested in videos recorded for hours –even days or weeks– in nursing houses or other indoor environments with cameras installed in various locations covering different regions in the environment. Usually benchmark datasets includes activities that subjects perform activities in front of the camera in a zone. Recently we found a new dataset called DAHLIA smart home [227] which is a perfect match for evaluating our frameworks. We have started to conduct extensive experiments on this dataset and we will report them in near future.

Gesture recognition
We are planning to record more data from real patients visiting the memory center. We also try to find a more accurate representation of the fingers. Further analysis will be carried out by applying different data mining techniques to interpret the recognition results for diagnostic purposes. Additionally, other important criteria introduced in this work such as reaction and movement times. It is not still confirmed that these criteria are indicative of the disease. Their impact on diagnosis of Alzheimer's disease will be deeply investigated.

7.3.2 Long-Term Perspective

We believe that the current work establish a firm foundation for future research in this domain. Hence, there are some potential research directions which could be addressed in the future. Here, we point out some of them.

As described in chapter 4, we utilize an ontology based hand-crafted model to refine the scene region model. Different combination modes with hand-crafted models can be done. The knowledge provided by the ontology models can contribute to the creation of richer hierarchical models. Conversely, the unsupervised models can contribute to the construction of more precise ontologies with less effort (supervision). Model matching can be carried out by the method we proposed in section 4.5.1 of chapter 4 and then, the recognition can be performed by the ontological models.

Although appearance features learn contextual information from the scene, to improve model's power of description, other more effective approaches and types of obtaining contextual information can be utilized. Object detection can contribute a lot in the creation of expressive scene models. Most of the daily activities are strongly related to specific objects. Object information can be also used in identifying scene regions as well as activities.

More research could be carried out on totally unsupervised framework. We are planning to create unsupervised video descriptions using topic models. By assigning different codewords to the extracted descriptors, topic models can encode the codeword sequences by learning different topics from the entire corpus (Videos). A probabilistic model will be employed to infer the pattern of codewords in the current time frame and also the patterns in the entire sequence.

We also investigate more on transfer learning and zero shot learning. We are interested in generating generic unsupervised activity models. Such model can be trained with a dataset and employed in testing of an unseen data with minimum modification. For example we want to learn "Making Coffee" activity in CHU dataset and be able to recognize this activity in the GAADRD dataset with the same model we learned. There are several issues need to be considered. For example the scene regions will be different in the two environments. A mechanism adapting the scene region models of different environments should be devised. Additionally, the activities can be performed differently from an environment to the other. Also, the camera view point may differ such as "Making Coffee" activity in CHU and GAADRD. In the first one the subject's back is to the camera, while, in the second one there is a side view of the subject preparing coffee. To create a generic model, all these issue should be taken into consideration.

Although in gesture recognition task we produced acceptable results in terms of accuracy of predictions, our dataset includes only 60 subjects and to achieve a reliable

system producing robust diagnosis, extensive training and evaluation will be required.

Finally, the two categories (Gesture and activity recognition frameworks) are not independent. A framework combining the two models is desirable. In a lower level of semantic hierarchy, gesture recognition can obtain more precise information about the gestures and help to recognize sub-activities. In higher semantic level activities can be described with the models that are mixture of gestures and composed sub-activity information.

Appendix A

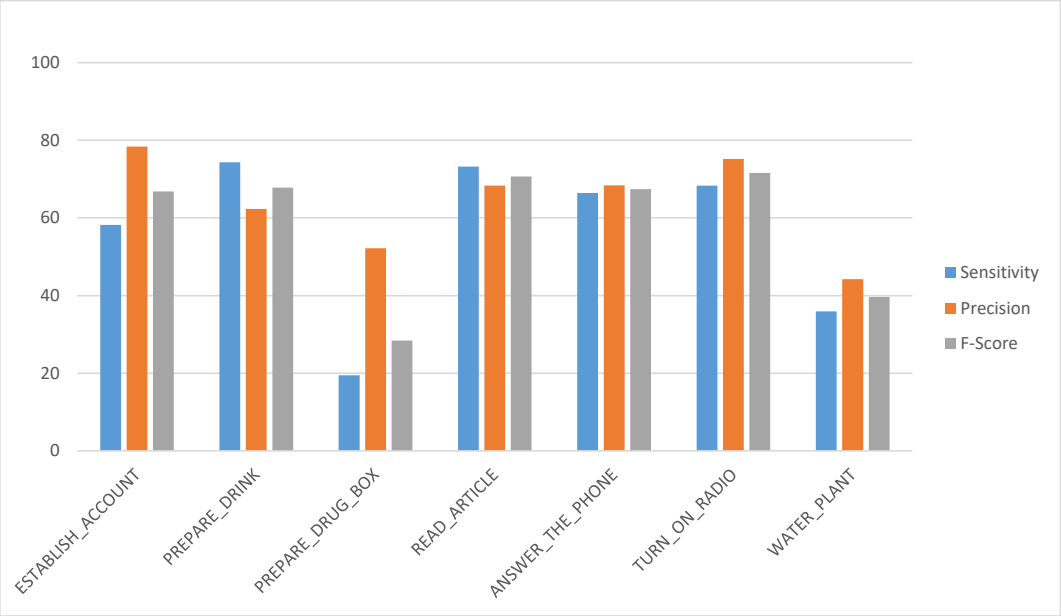
Appendix

Contents

A.1 Hybrid Framework

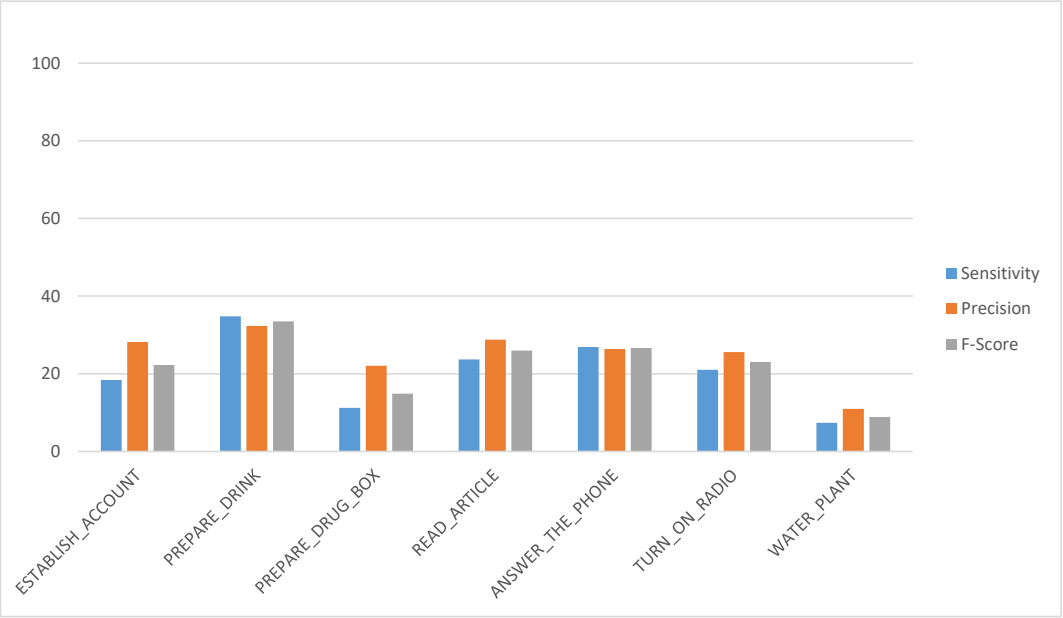
A.1.1 GAADRD Dataset

Here we report detailed results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset. The following tables show class-wise Precision and Recall metrics using different feature types to predict labels by the supervised classifier. The plots on top illustrate the same information of the tables. The gray bins show the F-Score metric. The blue and the orange bins represent Precision and Recall metrics respectively.



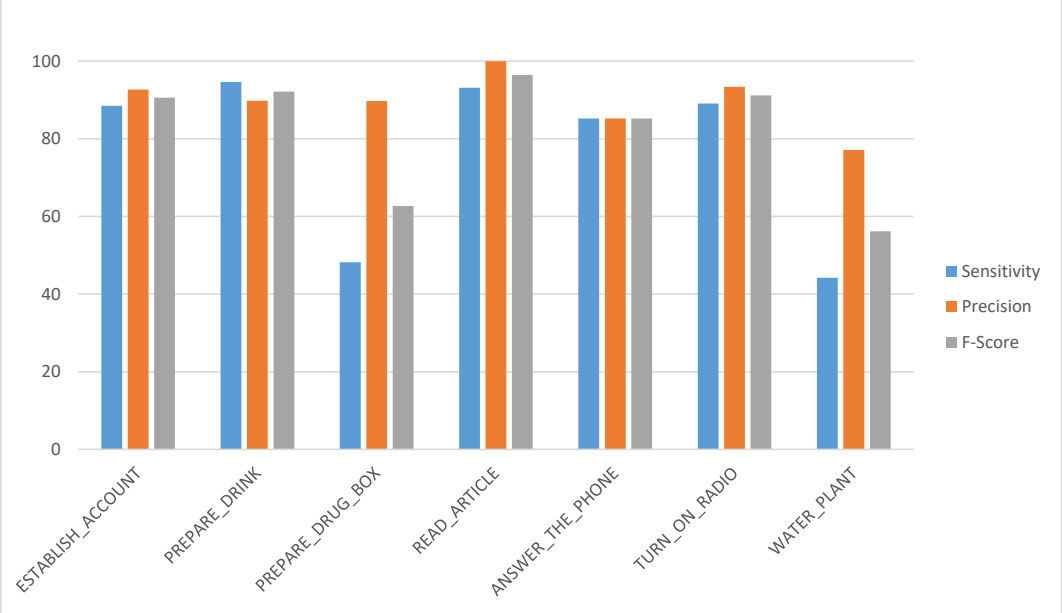
	Angle		
	Precision [%]	Recall [%]	F-Score
Establish Account	78.4	58.2	0.66
Prepare Drink	62.3	74.3	0.67
Prepare DrugBox	52.2	19.5	0.28
Read Article	68.3	73.2	0.70
Answer the Phone	68.4	66.4	0.67
Turn On Radio	75.2	68.3	0.71
Watering Plant	44.2	35.9	0.39
Average	64.14	56.54	0.58

Figure A.1: Illustrate the results of the Angle feature (Calculated from skeleton joints information). Compared to the supervised-only method, the hybrid method performs better as it takes benefits from global information in addition to the supervised cues from the classifier. Supervised method with the best configuration (Encoding method and dictionary size) achieves 0.55 accuracy with this feature type, however, the hybrid framework with the same choices obtains 0.58 in the F-Score measure. The best performance belongs to "Turn On Radio" activity with 0.71. This might be because of this activity's clear and distinguishable pose from the others that helps the method to perform better. The worst performance is on "Prepare DrugBox" class with 0.28 of the F-Score. This is because of low duration of this activity which makes feature extraction challenging (Similar to the "Watering Plant" activity).



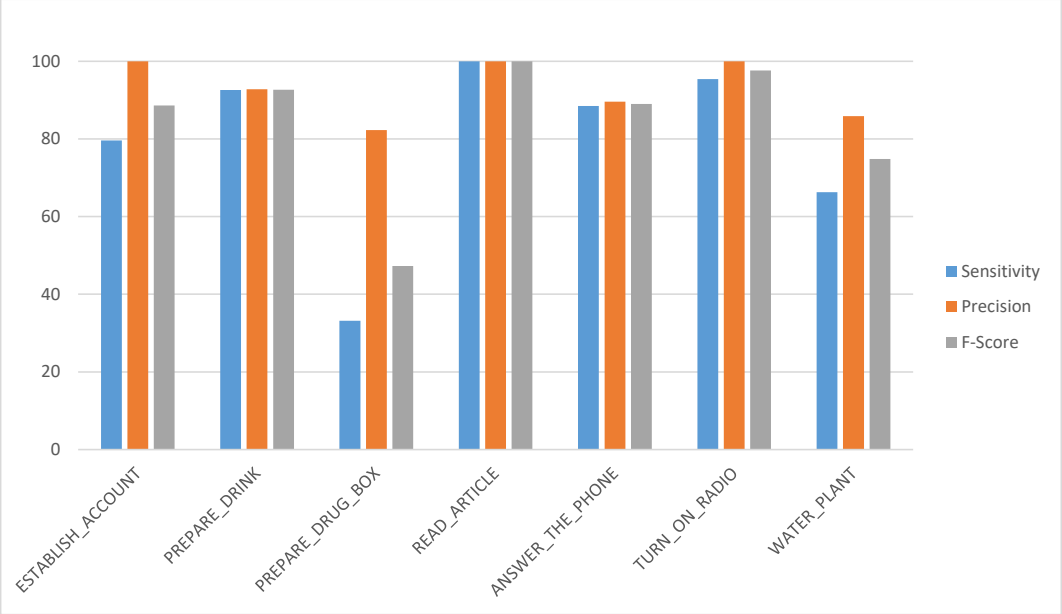
	Distance		
	Precision [%]	Recall [%]	F-Score
Establish Account	28.2	18.4	0.22
Prepare Drink	32.3	34.8	0.33
Prepare DrugBox	22.1	11.2	0.14
Read Article	28.8	23.7	0.26
Answer the Phone	26.4	26.9	0.27
Turn On Radio	25.6	21	0.23
Watering Plant	11	7.4	0.08
Average	24.91	20.49	0.22

Figure A.2: Illustrate the results of the Distance feature (Calculated from skeleton joints information). The hybrid method outperforms the supervised method by 0.06 in F-Score thanks to the extra information it utilizes. The best performance belongs to "Prepare Drink" activity with 0.33 and the worst is "Watering Plant" with 0.08 of the F-Score. The poor performance on the "Watering Plant" activity is due to lack of sufficient information that can be extracted from few frames that this activity contains. Similar to the supervised method, the Distance feature performs poorly in activity recognition task. It is the worst among the features. This is because the distance pose features calculated skeleton are not an strong feature to distinguish daily living activities as they include constant and not so discriminative poses.



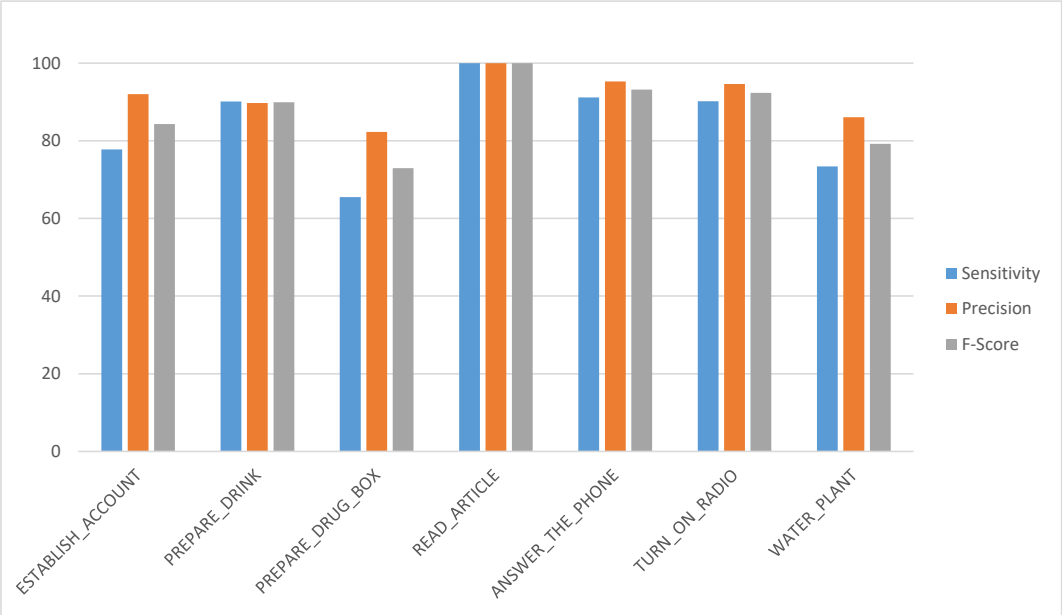
	HOF		
	Precision [%]	Recall [%]	F-Score
Establish Account	92.7	88.5	0.90
Prepare Drink	89.8	94.6	0.92
Prepare DrugBox	89.7	48.2	0.62
Read Article	100	93.1	0.96
Answer the Phone	85.2	85.2	0.85
Turn On Radio	93.4	89.1	0.91
Watering Plant	77.1	44.2	0.56
Average	89.70	77.56	0.82

Figure A.3: Show the results of applying the hybrid framework on the GAADRD dataset using the HOF descriptor. Not impressive as HOG but still it achieves acceptable accuracy with 0.82 of F-Score metric. Using the unsupervised information the hybrid framework outperforms the supervised framework by 0.03 of the F-Score. Using this descriptor with the hierarchical models gives the best performance with the "Read Article" activity with 0.96 in F-Score, whilst, the worst performance belongs to the "Watering Plant" activity with 0.56 of the F-Score.



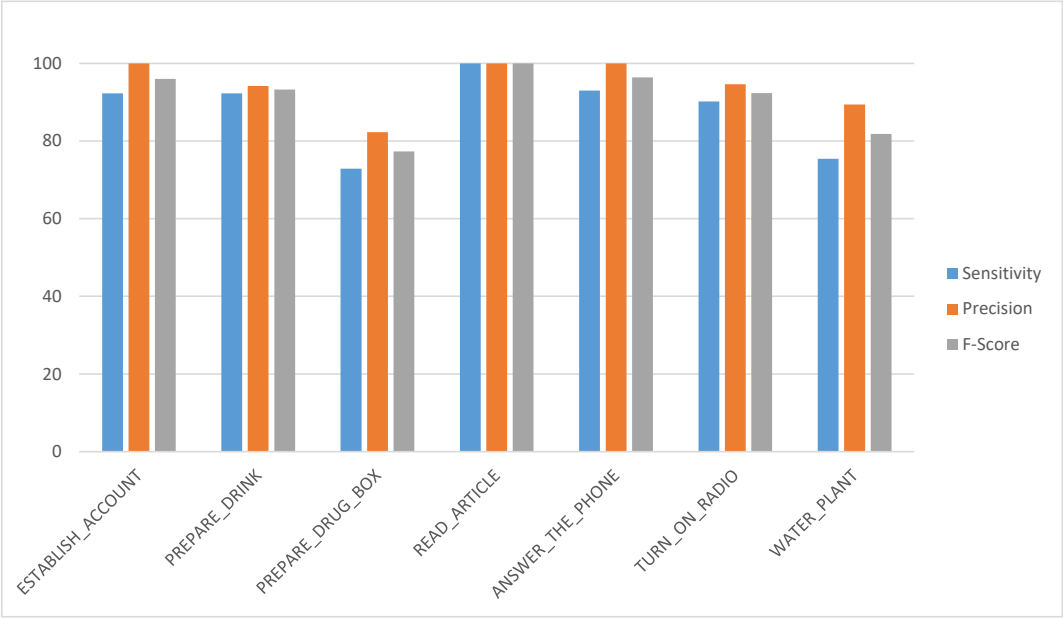
	MBHX		
	Precision [%]	Recall [%]	F-Score
Establish Account	100	79.6	0.88
Prepare Drink	92.8	92.6	0.92
Prepare DrugBox	82.3	33.2	0.47
Read Article	100	100	1
Answer the Phone	89.6	88.5	0.89
Turn On Radio	100	95.4	0.97
Watering Plant	85.9	66.3	0.74
Average	92.94	79.37	0.84

Figure A.4: Results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset with hand-crafted MBHX descriptor for the supervised classifier. This descriptor achieves 0.84 of F-Score metric. Adding the supervised information from this descriptor to the unsupervised models does not improve the accuracy of the recognition. The performance drops with 0.01 in F-Score (-1%). This happens usually because of the conflict in scoring process between the supervised label information and the similarity score coming from the unsupervised HAM models. The best performance belongs to the "Read Article" activity with 1.00 in F-Score, whilst, the worst performance belongs to the "Prepare DrugBox" activity with 0.47 of the F-Score.



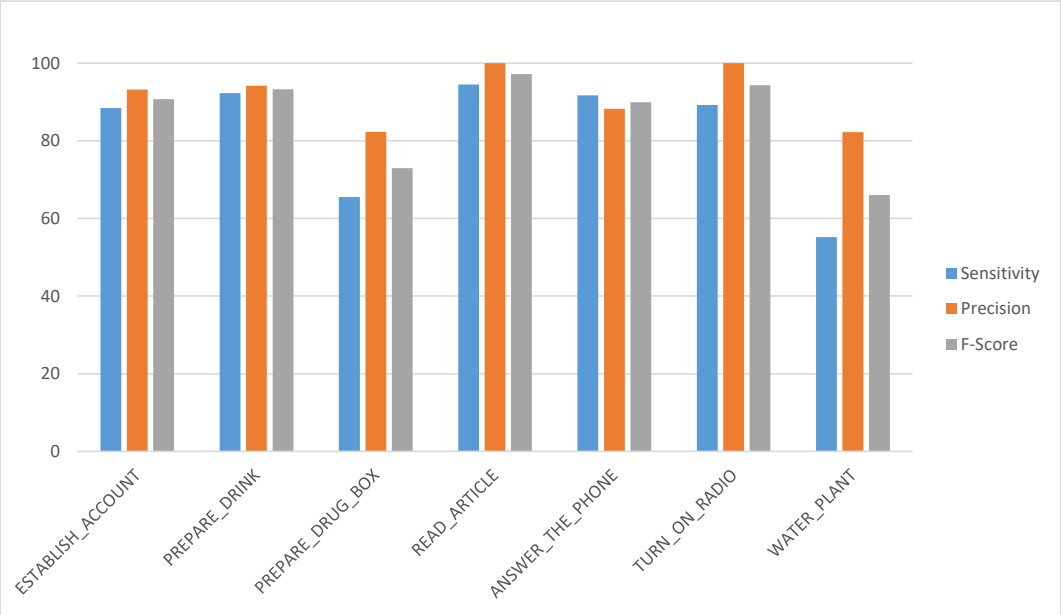
	MBHY		
	Precision [%]	Recall [%]	F-Score
Establish Account	92	77.8	0.84
Prepare Drink	89.7	90.1	0.89
Prepare DrugBox	82.3	65.5	0.72
Read Article	100	100	1
Answer the Phone	95.3	91.2	0.93
Turn On Radio	94.6	90.2	0.92
Watering Plant	86.1	73.4	0.79
Average	91.43	84.03	0.87

Figure A.5: Results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the MBHY descriptor for the supervised classifier. This descriptor achieves 0.87 of F-Score metric. It outperforms significantly the models with geometrical features and performs comparable to the other hand-crafted and deep features. Employing this descriptor helps the activity models to achieve 0.87 accuracy. It outperforms the supervised framework with 0.04 in F-Score metric. The best performance belongs to the "Read Article" activity with 1.00 in F-Score, whilst, the worst performance belongs to the "Prepare DrugBox" activity with 0.72 of the F-Score. This descriptor relies on motion information and poor performance of "Prepare DrugBox" is due to lack of enough motion information.



TDD Spatial			
	Precision [%]	Recall [%]	F-Score
Establish Account	100	92.3	0.95
Prepare Drink	94.2	92.3	0.93
Prepare DrugBox	82.3	72.9	0.77
Read Article	100	100	1
Answer the Phone	100	93	0.96
Turn On Radio	94.6	90.2	0.92
Watering Plant	89.4	75.4	0.81
Average	94.36	88.01	0.91

Figure A.6: Results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the TDD Spatial feature for the supervised classifier. This descriptor achieves 0.91 of F-Score metric. This accuracy rate is better than the highest rate of the supervised framework (0.89). Like most of the utilized feature types, the TDD Spatial deep feature also improve the recognition accuracy when compared to only-supervised method. It outperforms the supervised framework with 0.02 in F-Score metric which is equal to 2% improvement of recognition accuracy. In this dataset, TDD spatial outperforms TDD temporal feature showing that appearance information is more important in long-term activities where no t much motion information is available. The best performance belongs to the "Read Article" activity with 1.00 in F-Score, whilst, the worst performance belongs to the "Prepare DrugBox" activity with 0.77 of the F-Score.

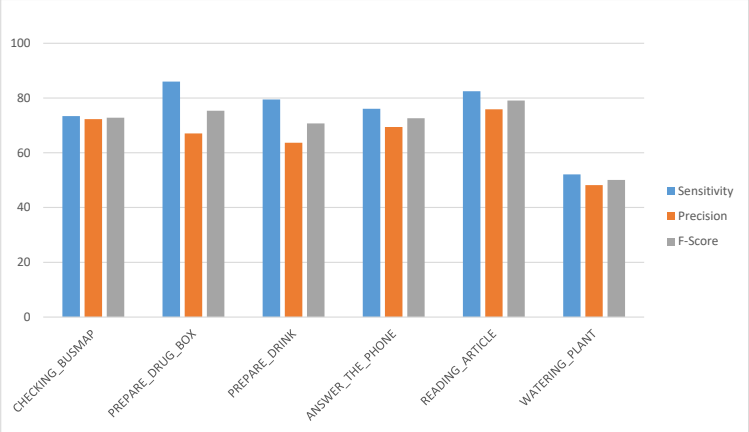


TDD Temporal			
	Precision [%]	Recall [%]	F-Score
Establish Account	93.2	88.4	0.90
Prepare Drink	94.2	92.3	0.93
Prepare DrugBox	82.3	65.5	0.72
Read Article	100	94.5	0.97
Answer the Phone	88.2	91.7	0.89
Turn On Radio	100	89.2	0.94
Watering Plant	82.2	55.2	0.66
Average	91.44	82.40	0.86

Figure A.7: Results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the TDD Temporal feature for the supervised classifier. The plot on top illustrates the same information of the table. This descriptor achieves 0.86 of F-Score metric. This accuracy rate is lower than the highest rate of the supervised framework (0.87). However, the difference is not significant (0.01 lower in F-Score metric). This shows that TDD Spatial deep features performs better than temporal deep feature both on supervised and hybrid methods on the GAADRD dataset which means that appearance features works better than motion features for this dataset. The best performance belongs to the "Read Article" activity with 0.97 in F-Score, whilst, the worst performance belongs to the "Watering Plant" activity with 0.66 of the F-Score. Further analysis shows that activities with similar motion patterns and similar duration are mostly confused with each other.

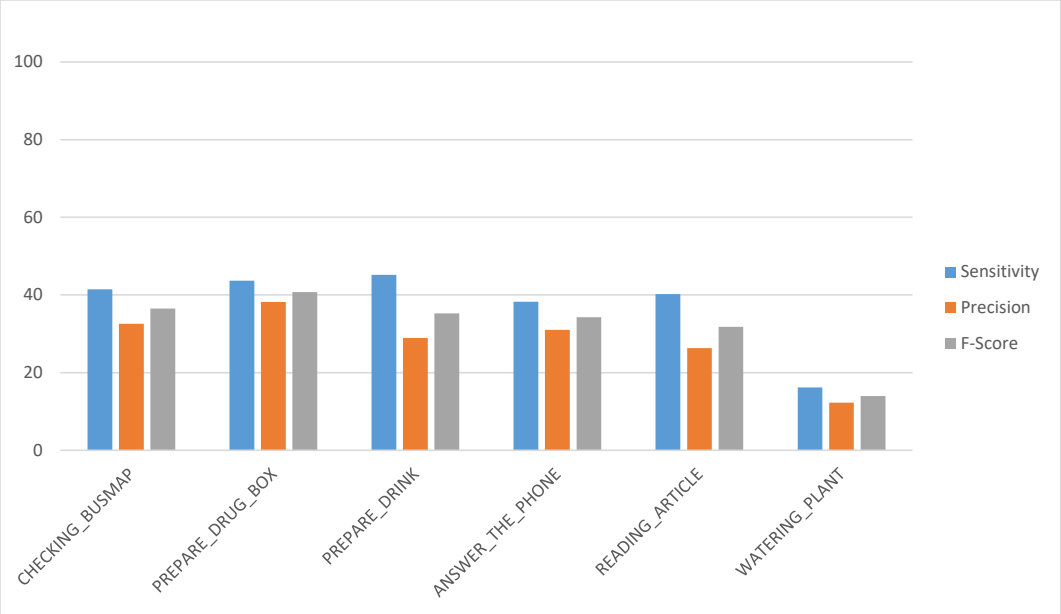
A.1.2 CHU Dataset

The details about this dataset are explained in section 3.7.2 of chapter 3. As explained in chapter 3 the Fisher Vector encoding is used for the supervised classifier where, we have tried different parameters for the SVM classifier and different codebook size for the FV dictionaries (16, 32, 64, 128, 256, and 512). Next, the results of the hybrid framework on the CHU Nice Hospital dataset are reported. The following tables show class-wise Precision and Recall metrics using different feature types to predict labels by the supervised classifier. The plots on top illustrate the same information of the tables. The gray bins show the F-Score metric. The blue and the orange bins represent Precision and Recall metrics respectively.



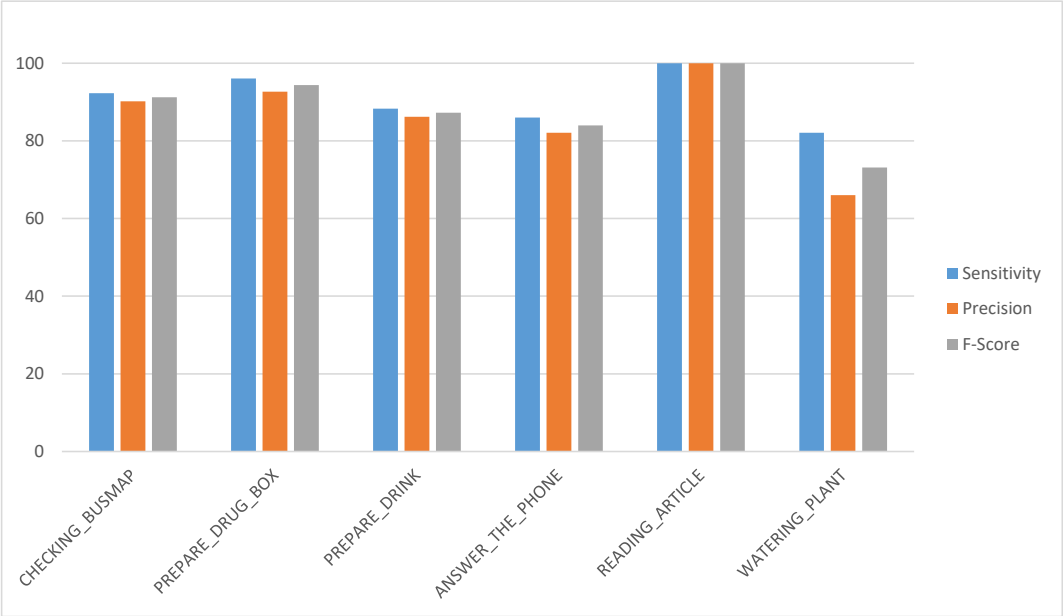
	Angle		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	72.3	73.4	0.72
Prepare DrugBox	67.1	86	0.75
Prepare Drink	63.7	79.5	0.70
Answer the Phone	69.4	76.1	0.72
Reading Article	75.9	82.5	0.79
Watering Plant	48.2	52.1	0.50
Average	66.10	74.93	0.70

Figure A.8: Results of applying the hybrid framework on the CHU dataset with Angle feature calculated from the skeleton information. The hierarchical modeling helps to improve the recognition accuracy. The improvement is not significant (0.01 of F-Score), yet, the overall performance is preserved compared to the supervised framework. The best and worst class-wise performance are similar to the most of the other features and belong to the "Reading Article" and the "Watering Plant" activities with F-score of 0.79 and 0.50 respectively. Detailed analysis of activities shows that similarity of pose in most of the activities cause confusion for the models using geometrical features and results in poor performance.



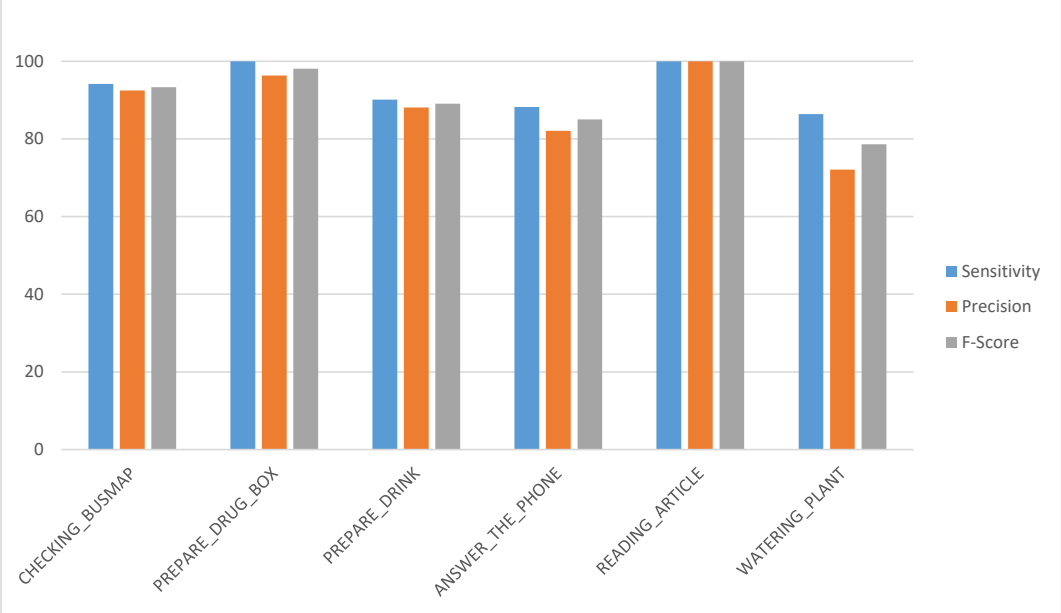
	Distance		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	32.6	41.5	0.36
Prepare DrugBox	38.2	43.7	0.40
Prepare Drink	28.9	45.2	0.35
Answer the Phone	31.0	38.3	0.34
Reading Article	26.3	40.2	0.31
Watering Plant	12.3	16.2	0.13
Average	28.22	37.52	0.32

Figure A.9: Results of applying the hybrid framework on the CHU dataset using the Distance feature for the supervised classifier. Although the raw distance feature is not high-quality and representative feature for the daily activities, the improvement that achieved by combining supervised and unsupervised information is the highest among the evaluated features. The overall rate of F-Score is 0.32 using the hybrid approach, whilst, it was 0.25 with the pure supervised framework (0.07 improvement in the F-Score rate). The best class-wise performance with the distance feature is achieved in the "Prepare DrugBox" activity with F-score of 0.40. This is different from the other feature types that usually they obtain lowe accuracy on this class. The worst belongs to the "Watering Plant" activity with F-Score of 0.13.



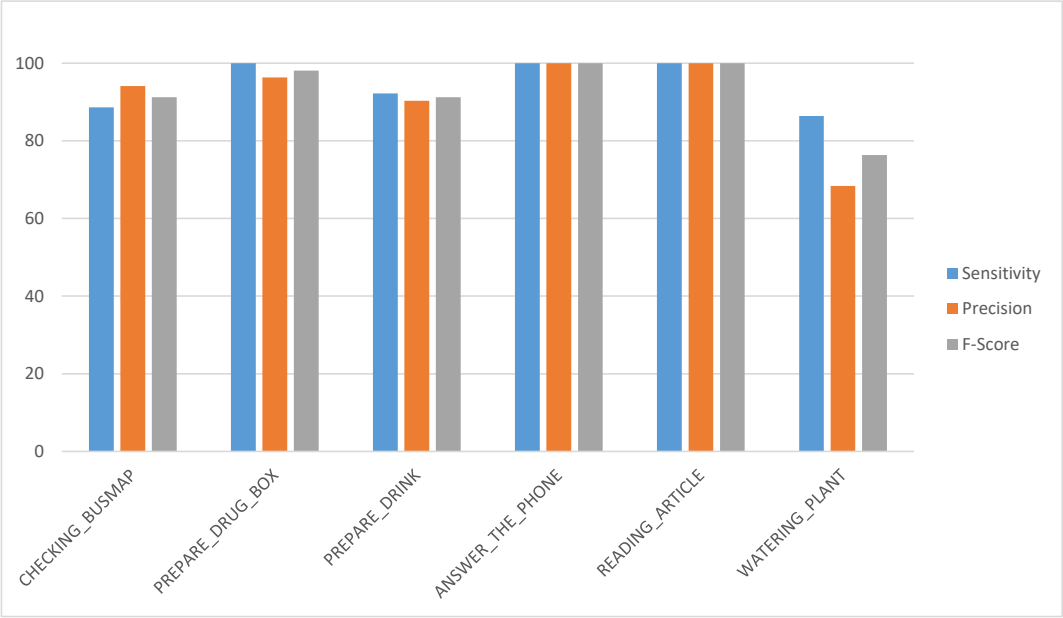
HOF			
	Precision [%]	Recall [%]	F-Score
Checking BusMap	90.2	92.3	0.91
Prepare DrugBox	92.7	96.1	0.94
Prepare Drink	86.2	88.3	0.87
Answer the Phone	82.1	86	0.84
Reading Article	100	100	1.00
Watering Plant	66	82.1	0.73
Average	86.20	90.80	0.88

Figure A.10: Illustrate the results of applying the hybrid framework using HOF descriptor on the CHU dataset. Using HOF descriptors along the HAM models also improve the accuracy of the recognition when compared to the pure supervised classifier. The accuracy improves from 0.84 in the value of F-Score to 0.88 when using the hybrid method with hierarchical models. The best recognition rate is achieved on the "Reading Article" activity and the worst is on the "Watering Plant" with 0.73 of the F-Score.



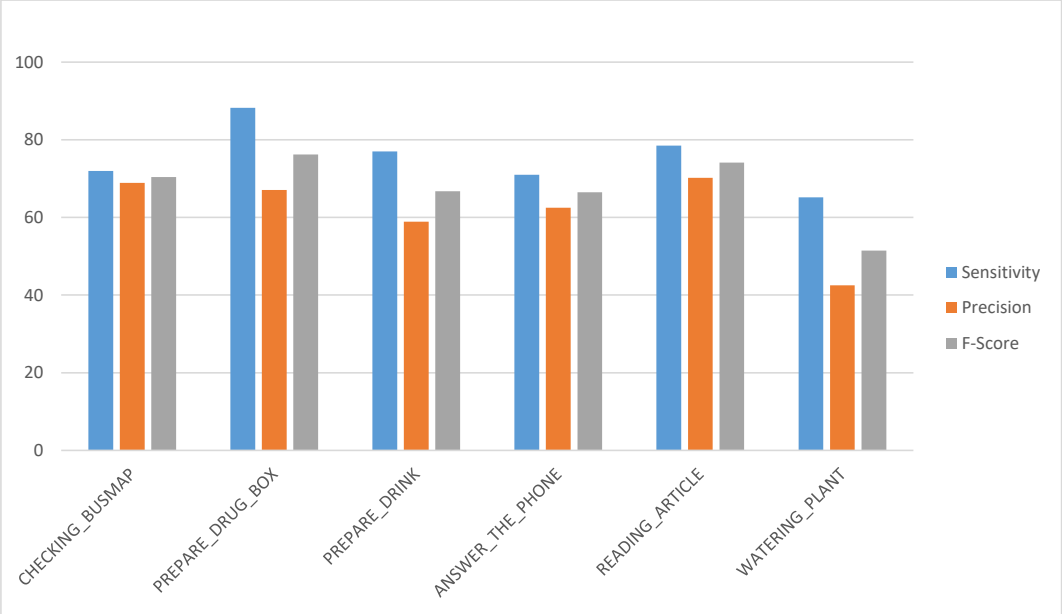
	MBHX		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	92.5	94.2	0.93
Prepare DrugBox	96.3	100	0.98
Prepare Drink	88.1	90.1	0.89
Answer the Phone	82.1	88.2	0.85
Reading Article	100	100	1.00
Watering Plant	72.1	86.4	0.78
Average	86.20	93.15	0.90

Figure A.11: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the MBHX descriptor for the supervised classifier. This descriptor achieves higher recognition rate than the supervised framework. The overal F-Score rate is 0.90 which is 0.02 higher than the supervised approach. The best performance achieved on the "Reading Article" activity and the worst is when the "Watering Plant" activity is performed.



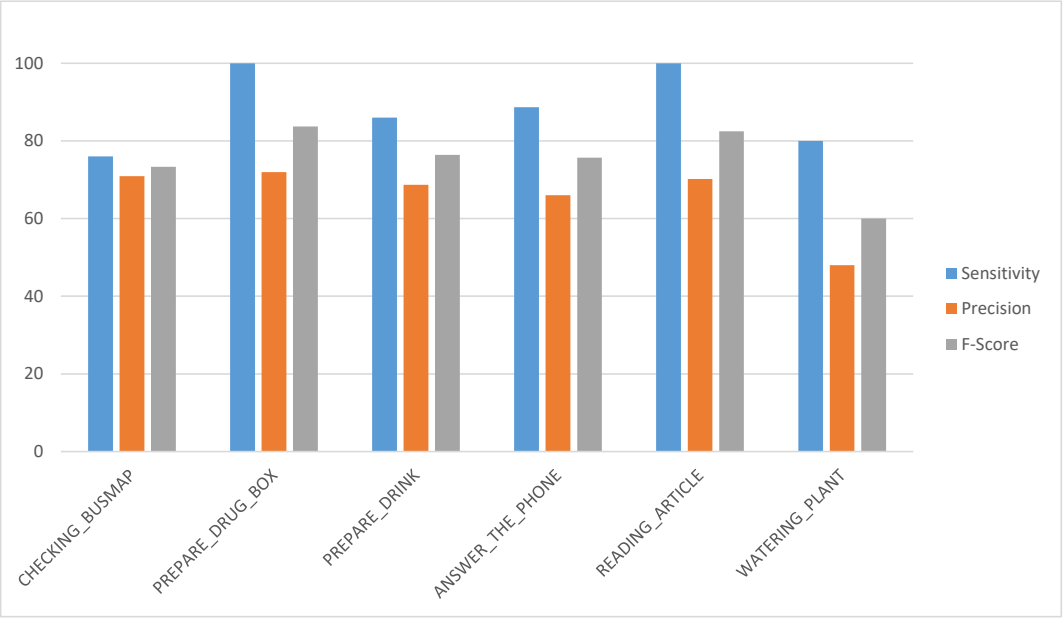
	MBHY		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	94.1	88.6	0.91
Prepare DrugBox	96.3	100	0.98
Prepare Drink	90.3	92.2	0.91
Answer the Phone	100	100	1.00
Reading Article	100	100	1.00
Watering Plant	68.4	86.4	0.77
Average	91.52	94.5	0.92

Figure A.12: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the MBHY descriptor for the supervised classifier. This descriptor achieves the second best recognition rate for the CHU Hospital dataset. It is only 0.03 below the performance of HOG descriptor. Its best performance is on the "Answer the Phone" and "Reading Article" activities since they include more motion and the lowest performance is again on "Watering Plant" activity with 0.77 rate in F-Score.



TDD Spatial			
	Precision [%]	Recall [%]	F-Score
Checking BusMap	68.9	72	0.70
Prepare DrugBox	67.1	88.2	0.76
Prepare Drink	58.9	77.0	0.66
Answer the Phone	62.5	71.0	0.66
Reading Article	70.2	78.5	0.74
Watering Plant	42.5	65.2	0.51
Average	61.68	75.32	0.67

Figure A.13: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the TDD Spatial deep feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Although this feature achieved good performance on the GAADRD dataset and even obtained better accuracy than the other deep feature type (TDD Temporal), it performs poorly on the CHU dataset. The performance of the supervised classifier is not high using this feature (0.65 F-Score) which causes a lower rate in the recognition task. The hierarchical models try to compensate the loss in accuracy but it does not boost the performance significantly (0.67 in F-Score).



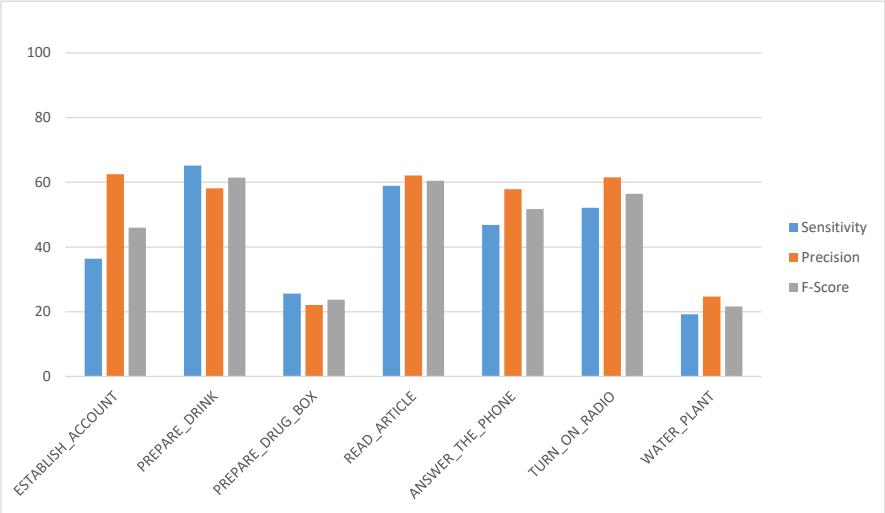
TDD Temporal			
	Precision [%]	Recall [%]	F-Score
Checking BusMap	70.9	76.0	0.73
Prepare DrugBox	72	100	0.83
Prepare Drink	68.7	86.0	0.76
Answer the Phone	66	88.7	0.75
Reading Article	70.2	100	0.82
Watering Plant	48.0	80.0	0.60
Average	65.97	88.45	0.75

Figure A.14: Results on the CHU dataset using temporal deep feature. Although this descriptor performs inferior than the spatial deep features on GAADRD dataset, it outperforms the TDD Spatial deep feature on this dataset. Although the ADL types are similar in both datasets, appearance features in one and motion features in other perform the best. This reveals that the performance of these features are dataset dependant. In CHU dataset the temporal duration of the activities are longer than the GAADRD dataset. It is where the temporality gains more importance and help the models to obtain enough information about each activity. It might be the reason that TDD temporal outperforms TDD spatial on CHU. The best performance is achieved on the "Prepare DrugBox" activity which is performed clearly in front of the camera from a side view (Hand motions are clearly visible). The lowest rate of recognition is on the "Watering Plant" activity which is performed far from the camera with high speed and back of the subject to the camera and.

A.2 Unsupervised Framework

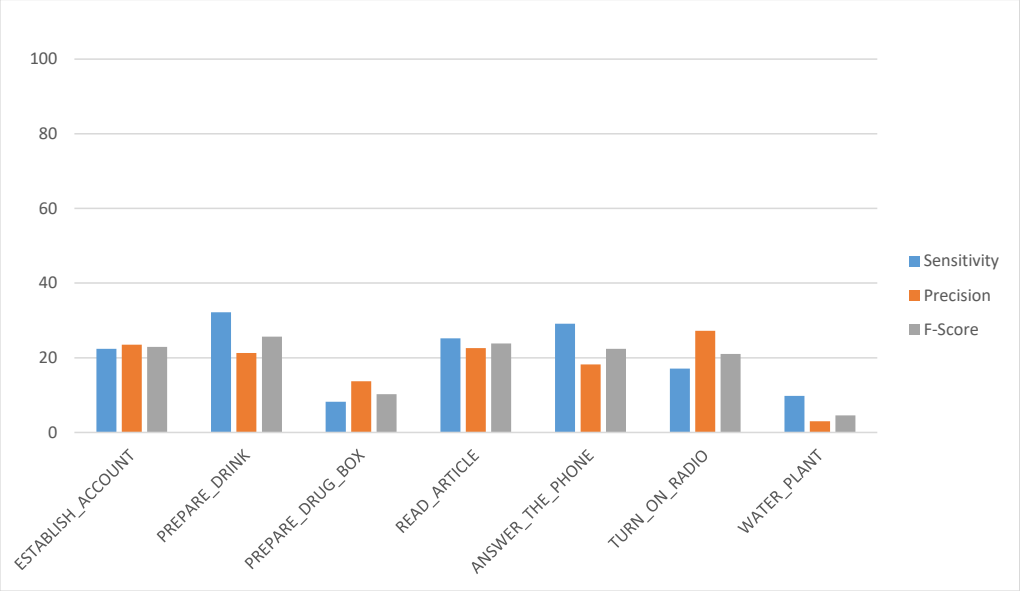
A.2.1 GAADRD Dataset

The details about this dataset are explained in section 3.7.1 of chapter 3. We report the results of experiments on this dataset from table A.15 to table A.21. Description about the results using each feature is included in the captions of the tables.



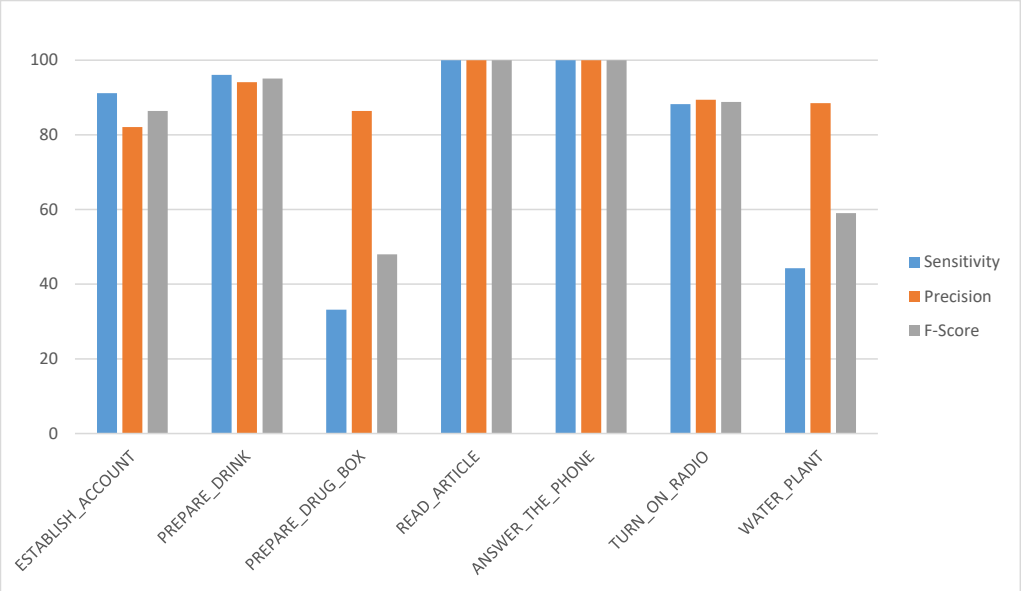
	Angle		
	Precision [%]	Recall [%]	F-Score
Establish Account	62.5	36.4	0.46
Prepare Drink	58.1	65.2	0.61
Prepare DrugBox	22.1	25.6	0.23
Read Article	62.1	58.9	0.60
Answer the Phone	57.9	46.8	0.51
Turn On Radio	61.5	52.1	0.56
Watering Plant	24.7	19.2	0.21
Average	49.84	43.46	0.45

Figure A.15: Results of applying the unsupervised framework on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the Angle feature for descriptor matching procedure. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. The performance of the unsupervised framework is lower than hybrid method using this type of feature. However, similar to the previous approaches, this feature type outperforms the Distance feature type. The best performance belongs to “Prepare Drink” activity with 0.61 and the worst is “Watering Plant” with 0.21 of the F-Score.



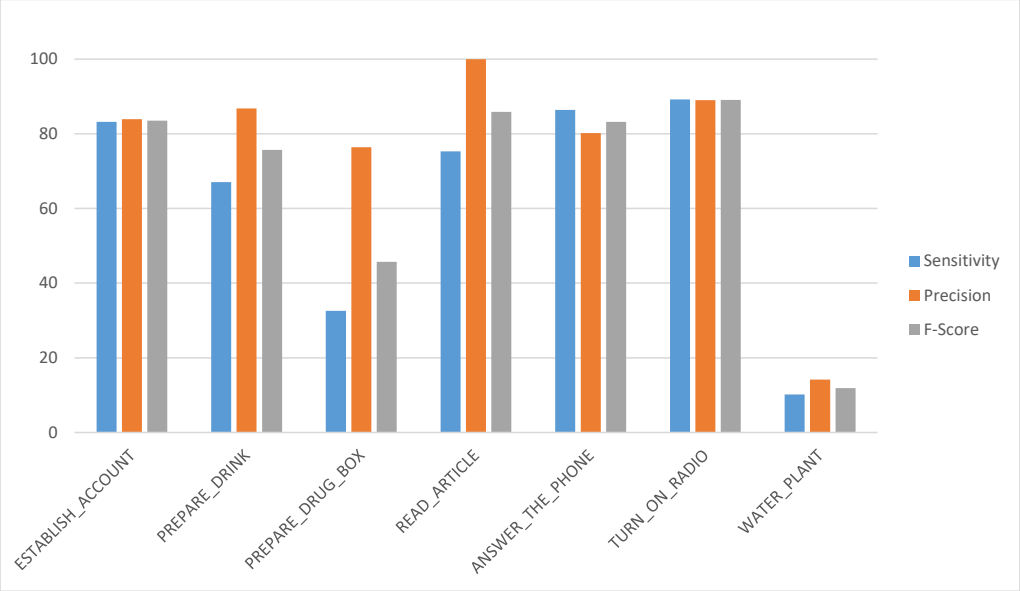
	Distance		
	Precision [%]	Recall [%]	F-Score
Establish Account	23.5	22.4	0.22
Prepare Drink	21.3	32.2	0.25
Prepare DrugBox	13.7	8.2	0.10
Read Article	22.6	25.2	0.23
Answer the Phone	18.2	29.1	0.22
Turn On Radio	27.2	17.1	0.21
Watering Plant	3.0	9.8	0.04
Average	18.50	20.57	0.18

Figure A.16: Results of applying the unsupervised framework on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the Distance feature for descriptor matching procedure. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. The performance of the unsupervised method is below the supervised and the hybrid method. The best performance belongs to "Prepare Drink" activity with 0.25 and the worst is "Watering Plant" with 0.04 of the F-Score. Similar to the supervised and hybrid methods, the Distance feature performs poorly in activity recognition task. It is the worst among the features.



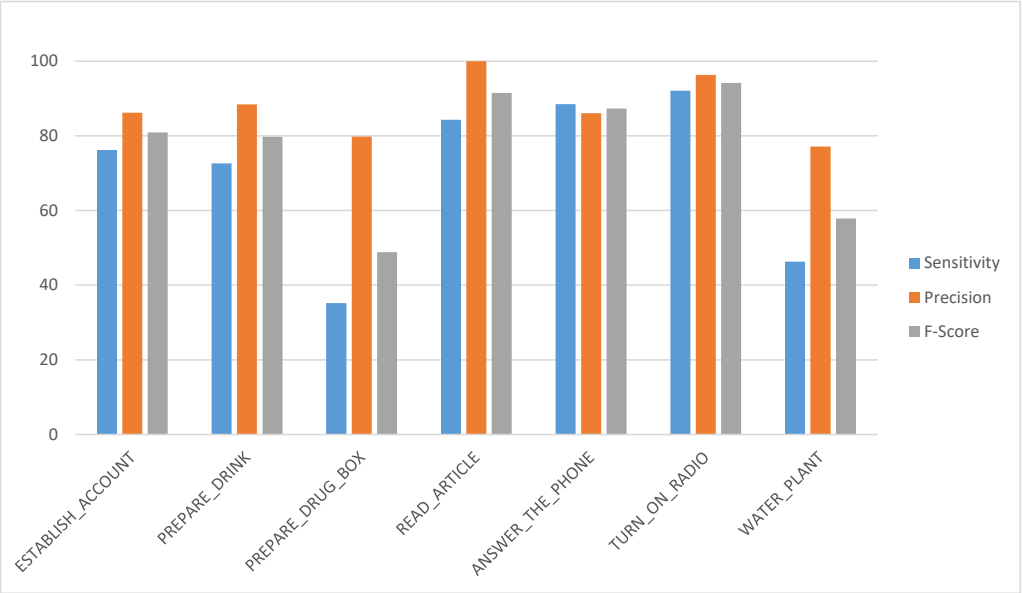
	HOG		
	Precision [%]	Recall [%]	F-Score
Establish Account	82.1	91.2	0.86
Prepare Drink	94.1	96.1	0.95
Prepare DrugBox	86.4	33.2	0.48
Read Article	100	100	1.0
Answer the Phone	100	100	1.0
Turn On Radio	89.4	88.2	0.88
Watering Plant	88.5	44.3	0.59
Average	91.50	79.00	0.82

Figure A.17: Results of applying the unsupervised framework on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the HOG feature for descriptor matching procedure. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. There is a relatively big drop in performance compared to the supervised and hybrid methods. The F-Score of the unsupervised method is almost 10% lower than the hybrid method. The best performance belongs to "Reading Article" and "Answer the Phone" activities with 1.0 and the worst is "Prepare DrugBox" activity with 0.48 of the F-Score.



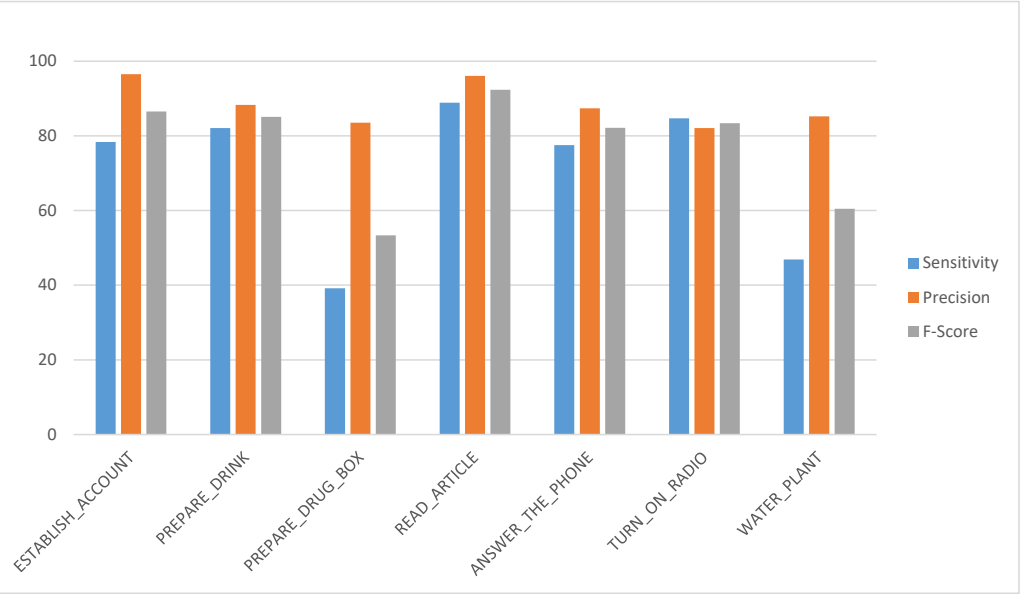
	HOF		
	Precision [%]	Recall [%]	F-Score
Establish Account	83.9	83.2	0.83
Prepare Drink	86.8	67.1	0.75
Prepare DrugBox	76.4	32.6	0.45
Read Article	100	75.3	0.85
Answer the Phone	80.2	86.4	0.83
Turn On Radio	89.0	89.2	0.89
Watering Plant	14.2	10.2	0.11
Average	75.79	63.43	0.67

Figure A.18: Results of applying the unsupervised framework on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the HOF feature for descriptor matching procedure. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Also in this feature type the performance is lower than the hybrid method. The biggest performance drop is on "Watering Plant" activity. The F-Score decreases from 0.56 in the hybrid method to 0.11 in the unsupervised method. "Turn On Radio" activity is recognized with the highest accuracy using the HOF descriptor. However, "Read Article" activity achieved the best performance using this descriptor on the hybrid method. The overall performance is 15% lower than the hybrid method using this descriptor.



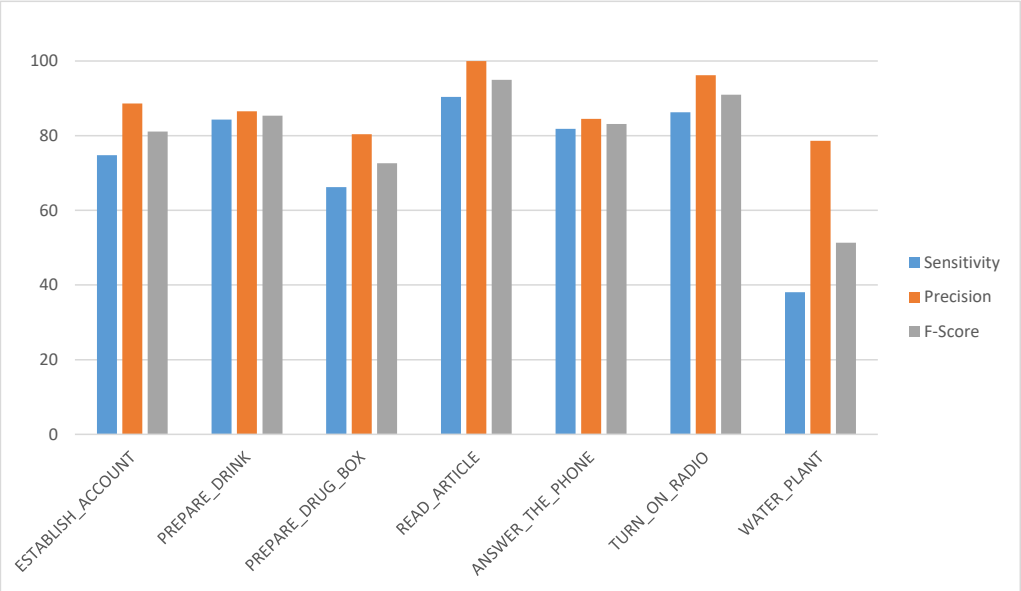
	MBHX		
	Precision [%]	Recall [%]	F-Score
Establish Account	86.2	76.2	0.80
Prepare Drink	88.4	72.6	0.79
Prepare DrugBox	79.8	35.2	0.48
Read Article	100	84.3	0.91
Answer the Phone	86.1	88.5	0.87
Turn On Radio	96.3	92.1	0.94
Watering Plant	77.1	46.3	0.57
Average	87.70	70.74	0.77

Figure A.19: Results of applying the unsupervised framework on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the MBHX feature for descriptor matching procedure. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Also in this feature type the performance is lower than the hybrid method. When it is compared to the hybrid method, the unsupervised method achieves an acceptable accuracy. It achieves 0.77 of F-Score while this score was 0.84 with hybrid method. The lowest performance belongs to the "Prepare DrugBox" activity which is 0.01 higher than the same activity's performance using the hybrid method showing the effectiveness of the unsupervised method. Using this descriptor, the unsupervised models achieve the best performance in recognizing "Turn On Radio" activity.



TDD Spatial			
	Precision [%]	Recall [%]	F-Score
Establish Account	96.5	78.4	0.86
Prepare Drink	88.3	82.1	0.85
Prepare DrugBox	83.5	39.2	0.53
Read Article	96.1	88.9	0.92
Answer the Phone	87.4	77.5	0.82
Turn On Radio	82.1	84.7	0.83
Watering Plant	85.2	46.9	0.60
Average	88.44	71.10	0.77

Figure A.20: Results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the TDD Spatial feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Although the performance of the unsupervised method seems acceptable when it is compared to the other feature types, there is a big drop of accuracy when it is compared to the hybrid method. Using this deep descriptor the hybrid method achieved 0.91 F-Score, while this metric is 0.77 using the unsupervised framework. However, using this descriptor, high Precision is achieved on most of the activities. The best performance belongs to the "Read Article" activity with 0.92 F-Score.

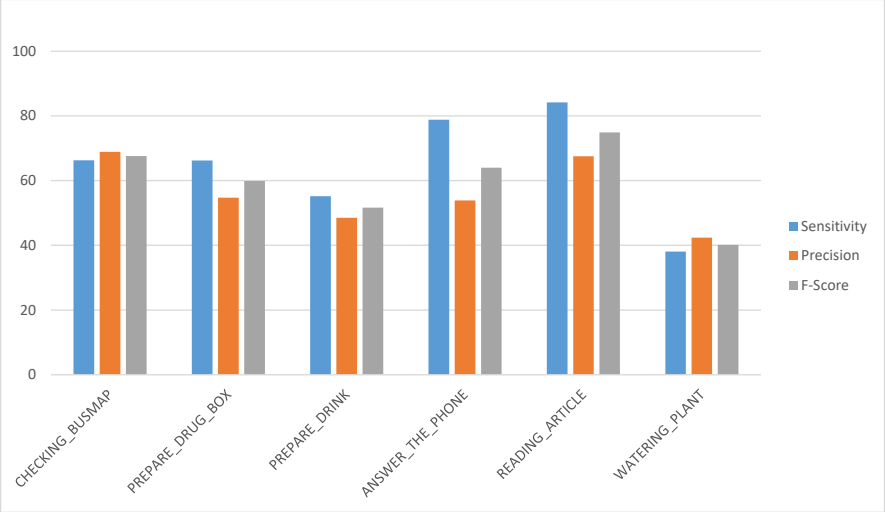


TDD Temporal			
	Precision [%]	Recall [%]	F-Score
Establish Account	88.6	74.8	0.81
Prepare Drink	86.5	84.3	0.85
Prepare DrugBox	80.4	66.2	0.72
Read Article	100	90.4	0.95
Answer the Phone	84.5	81.8	0.83
Turn On Radio	96.2	86.3	0.91
Watering Plant	78.6	38.1	0.51
Average	87.83	74.56	0.79

Figure A.21: Results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the TDD Temporal feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Unlike the hybrid method that the Spatial component of the TDD deep feature outperforms the Temporal component, the TDD Temporal descriptor outperforms the Spatial descriptor using the unsupervised framework. The unsupervised framework achieves acceptable performance in all of the activity classes using this feature. Its best performance is on "Read Article" class with over 0.95 F-Score.

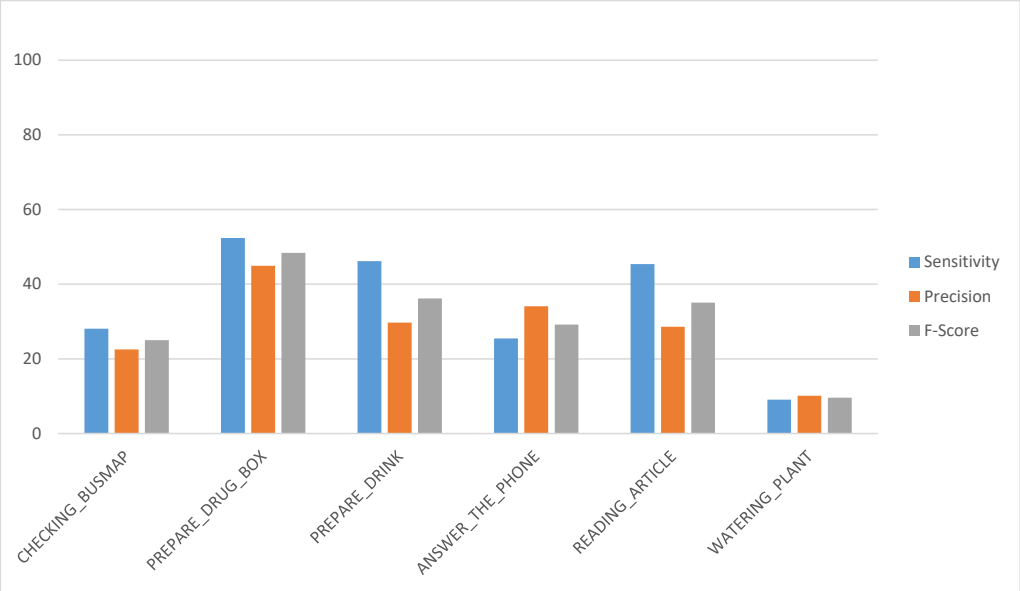
A.2.2 CHU Dataset

The details about this dataset are explained in section 3.7.2 of chapter 3. Next, from table A.22 to table A.28, the results of the unsupervised framework on the CHU Nice Hospital dataset is reported.



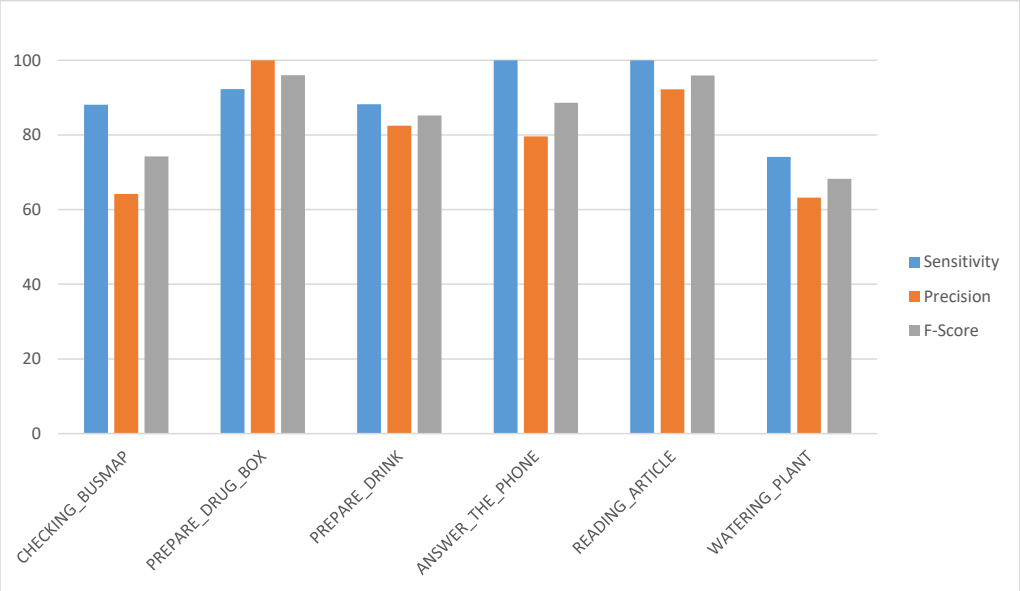
	Angle		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	68.9	66.3	0.67
Preparing DrugBox	54.7	66.2	0.59
Prepare Drink	48.5	55.2	0.51
Answer the Phone	53.9	78.8	0.64
Reading Article	67.5	84.2	0.74
Watering Plant	42.4	38.1	0.40
Average	55.98	64.80	0.59

Figure A.22: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the Angle feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Compared to the hybrid approach, the unsupervised framework obtains a lower F-Score. This feature performs lower than its equivalent in the hybrid and supervised methods. However, it achieves acceptable performance in most of the classes. The highest performance achieved in "Reading Article" activity and the poorest performance is on the "Watering Plant" activity class.



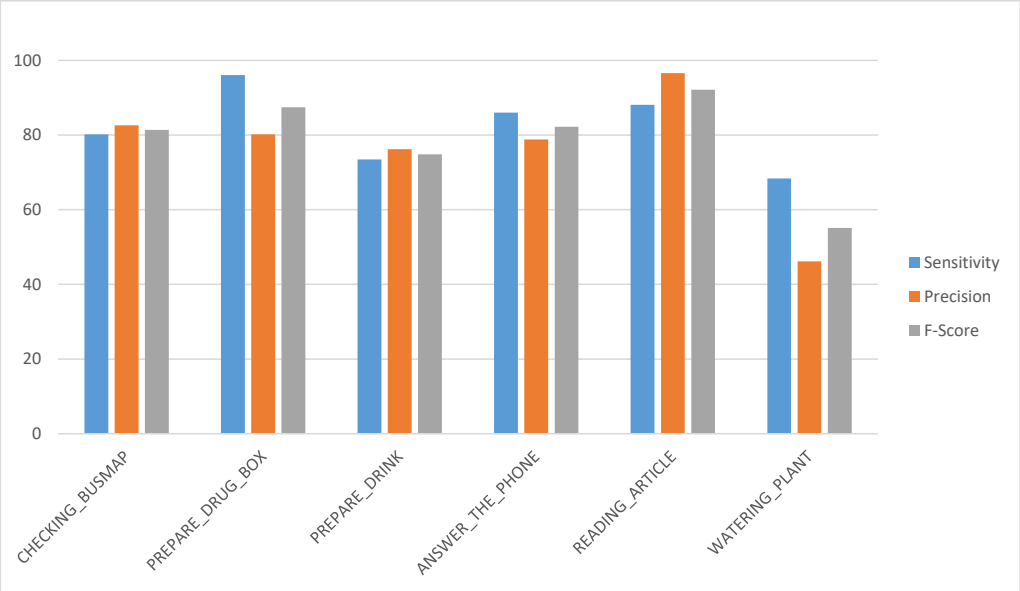
	Distance		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	22.5	28.1	0.25
Preparing DrugBox	44.9	52.4	0.48
Prepare Drink	29.7	46.2	0.36
Answer the Phone	34.1	25.5	0.29
Reading Article	28.6	45.4	0.35
Watering Plant	10.1	9.1	0.9
Average	28.32	34.45	0.30

Figure A.23: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the Distance feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. It is interesting that the distance feature achieves very similar performance to the one in the hybrid method. In hybrid method it achieved 0.32 F-Score while in unsupervised it obtained 0.30. It is the least performance difference among all the feature types. Although this feature's performance is the worst among the others, it demonstrates the most stable performance.



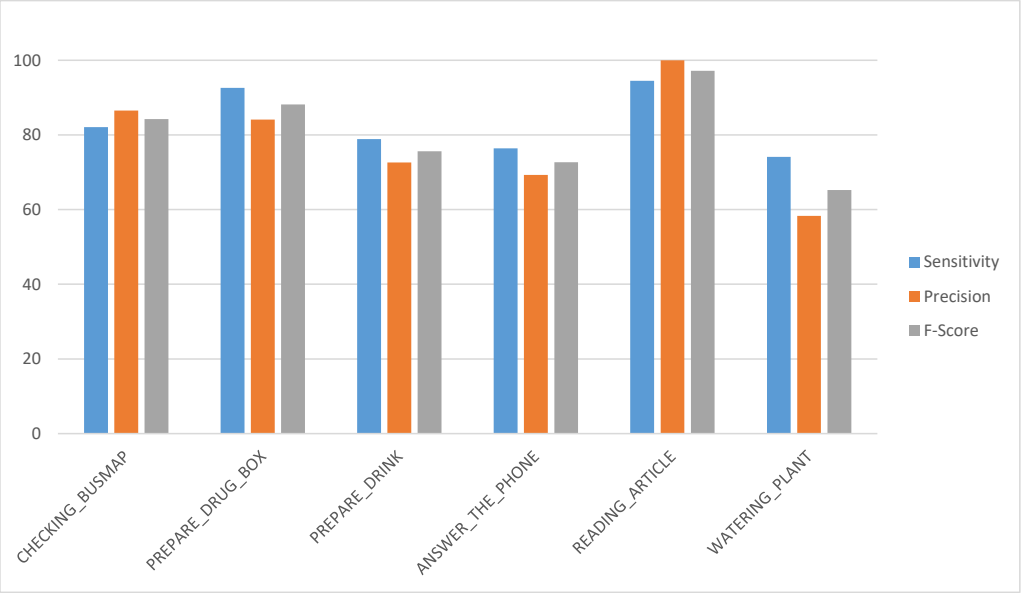
	HOG		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	64.2	88.1	0.74
Preparing DrugBox	100	92.3	0.96
Prepare Drink	82.5	88.2	0.85
Answer the Phone	79.6	100	0.88
Reading Article	92.2	100	0.95
Watering Plant	63.2	74.1	0.68
Average	80.28	90.45	0.84

Figure A.24: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the HOG feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. While there is a big gap between the performance of this feature in hybrid framework and the unsupervised architecture, it achieves competitive performance among the best performing features in the unsupervised method. HOG descriptor combined with the HAM models achieves the second best performance among all of the features with 0.84 F-Score. It is way beyond its performance in the hybrid method where combining the supervised classifier trained on this feature with the HAM models achieved the highest performance among all features as well as all methods (0.96 F-Score).



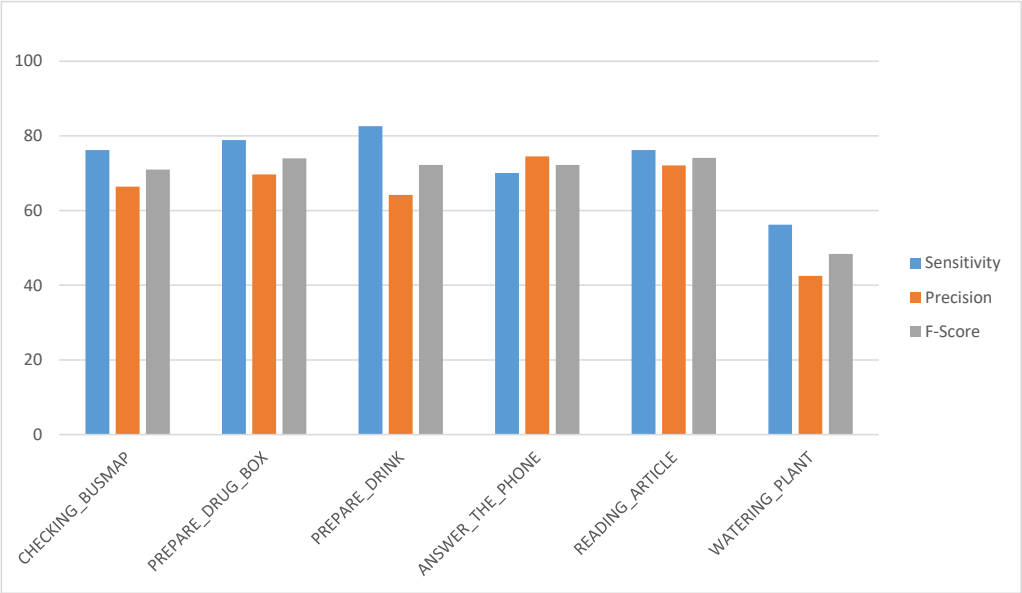
	HOF		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	82.6	80.2	0.81
Preparing DrugBox	80.2	96.1	0.87
Prepare Drink	76.2	73.5	0.74
Answer the Phone	78.8	86.0	0.82
Reading Article	96.6	88.1	0.92
Watering Plant	46.2	68.4	0.55
Average	76.77	82.05	0.78

Figure A.25: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the HOF feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Similar to the HOG feature, this one is also stay behind its counterpart in the hybrid model. With the hybrid it achieved 0.88 F-Score, while with unsupervised it achieves 0.78 F-Score. Knowing that the current method is unsupervised is enough to interpret this level of accuracy as an acceptable one.



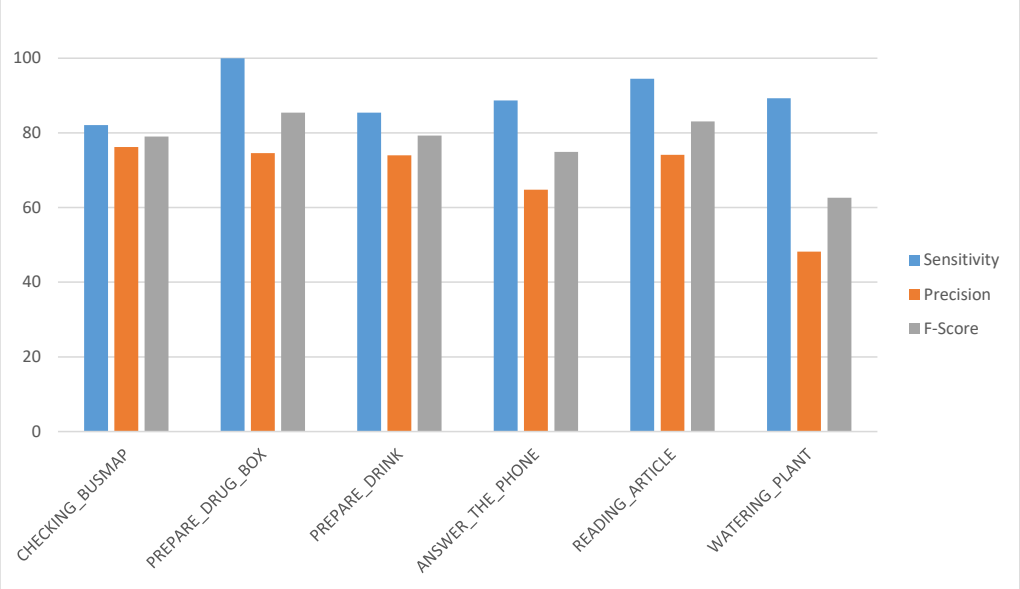
	MBHX		
	Precision [%]	Recall [%]	F-Score
Checking BusMap	86.5	82.1	0.84
Preparing DrugBox	84.1	92.6	0.88
Prepare Drink	72.6	78.9	0.75
Answer the Phone	69.3	76.4	0.72
Reading Article	100	94.5	0.97
Watering Plant	58.3	74.1	0.65
Average	78.47	83.10	0.80

Figure A.26: Results of applying the hybrid framework (supervised+unsupervised) on the CHU dataset. The table shows class-wise Precision and Recall metrics using the MBHX feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. The performance of combining MBHX feature with the hierarchical models results in a lower performance compared to the hybrid method (0.10 difference in F-Score). In overall, we can observe that the performance of each method also depends on the dataset. The MBH features performed better on the GAADRD dataset despite the fact that the GAADRD dataset contains higher number of activities. Nevertheless, we will see in the section 5.5 that the CHU Nice Hospital dataset includes some issues that makes it a challenging dataset.



TDD Spatial			
	Precision [%]	Recall [%]	F-Score
Checking BusMap	66.4	76.2	0.71
Preparing DrugBox	69.7	78.9	0.74
Prepare Drink	64.2	82.6	0.72
Answer the Phone	74.5	70.1	0.72
Reading Article	72.1	76.2	0.74
Watering Plant	42.5	56.2	0.48
Average	64.90	73.37	0.68

Figure A.27: Results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the TDD Spatial deep feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Deep features show stable performance on both hybrid and unsupervised methods. TDD Spatial feature scored 0.67 overall F-Score in hybrid method and 0.68 in unsupervised. It is interesting to see that it outperforms the hybrid approach.



TDD Temporal			
	Precision [%]	Recall [%]	F-Score
Checking BusMap	76.2	82.1	0.79
Preparing DrugBox	74.6	100	0.85
Prepare Drink	74.0	85.4	0.79
Answer the Phone	64.8	88.7	0.74
Reading Article	74.1	94.5	0.83
Watering Plant	48.2	89.3	0.62
Average	68.65	90.0	0.77

Figure A.28: Results of applying the hybrid framework (supervised+unsupervised) on the GAADRD dataset. The table shows class-wise Precision and Recall metrics using the Distance feature for the supervised classifier. The plot on top illustrates the same information of the table. The orange bins show the Recall metric. The blue and the gray bins represent Precision and F-Score metrics respectively. Similar to the other deep feature, this one is also shows performance stability when we change from hybrid to unsupervised framework. It achieves acceptable accuracy in activity recognition by obtaining 0.77 F-Score. TDD Temporal achieves superb performance on Recall metric (0.90). Its overall performance beats the hybrid method's with a small margin (0.02 of F-Score). Together with TDD Spatial feature, deep features are the only feature types that outperform the hybrid method (in one-on-one comparison with same feature type on both methods).

A.3 Conclusions

In this chapter we presented detailed results of applying our proposed frameworks (hybrid and unsupervised) on GAADRD and CHU datasets. Dedicated tables and plots for each feature type show activity-wise accuracy of the two frameworks. The discussion about each feature type are given in the captions of the tables.

Bibliography

[1] Mit summer vision project 1966. http://hdl.handle.net/1721.1/6125. 1966. (Cited on page 2.)

[2] AGAHIAN, S., NEGIN, F., AND KÖSE, C. Improving bag-of-poses with semi-temporal pose descriptors for skeleton-based action recognition. *The Visual Computer* (2018), 1–17. (Cited on page 26.)

[3] AGGARWAL, J., AND RYOO, M. S. Human activity analysis: A review. *ACM Computing Surveys (CSUR) 43*, 3 (2011), 16. (Cited on pages 17, 19, 30 and 38.)

[4] ALEXIADIS, D. S., KELLY, P., DARAS, P., O'CONNOR, N. E., BOUBEKEUR, T., AND MOUSSA, M. B. Evaluating a dancer's performance using kinect-based skeleton tracking. In *Proceedings of the 19th ACM international conference on Multimedia* (2011), ACM, pp. 659–662. (Cited on page 163.)

[5] AMSUSS, S., GOEBEL, P. M., JIANG, N., GRAIMANN, B., PAREDES, L., AND FARINA, D. Self-correcting pattern recognition system of surface emg signals for upper limb prosthesis control. *IEEE Transactions on Biomedical Engineering 61*, 4 (2014), 1167–1176. (Cited on page 163.)

[6] ASSOCIATION, A. P. Diagnostic and statistical manual of mental disorders. *text rev.)* (2000). (Cited on page 160.)

[7] AVGERINAKIS, K., BRIASSOULI, A., AND KOMPATSIARIS, I. Recognition of activities of daily living for smart home environments. In *Intelligent Environments (IE), 2013 9th International Conference on* (2013), IEEE, pp. 173–180. (Cited on page 160.)

[8] AVGERINAKIS, K., BRIASSOULI, A., AND KOMPATSIARIS, I. Activity detection using sequential statistical boundary detection (ssbd). In *to appear in Computer Vision and Image Understanding* (2015), CVIU. (Cited on pages 128, 129, 132 and 146.)

[9] AVGERINAKIS, K., BRIASSOULI, A., AND KOMPATSIARIS, Y. Activity detection using sequential statistical boundary detection (ssbd). *Computer Vision and Image Understanding 144* (2016), 46–61. (Cited on page 165.)

[10] BACCOUCHE, M., MAMALET, F., WOLF, C., GARCIA, C., AND BASKURT, A. Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding* (2011), Springer, pp. 29–39. (Cited on pages 24 and 165.)

[11] BACCOUCHE, M., MAMALET, F., WOLF, C., GARCIA, C., AND BASKURT, A. Spatio-temporal convolutional sparse auto-encoder for sequence classification. In *BMVC* (2012), pp. 1–12. (Cited on page 164.)

[12] BANERJEE, T., KELLER, J. M., POPESCU, M., AND SKUBIC, M. Recognizing complex instrumental activities of daily living using scene information and fuzzy logic. *Computer Vision and Image Understanding 140* (2015), 68–82. (Cited on pages 20, 42, 44 and 160.)

[13] BASHARAT, A., GRITAI, A., AND SHAH, M. Learning object motion patterns for anomaly detection and improved object detection. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8. (Cited on pages 32, 79 and 108.)

[14] BAUMANN, F., EHLERS, A., ROSENHAHN, B., AND LIAO, J. Recognizing human actions using novel space-time volume binary patterns. *Neurocomputing 173* (2016), 54–63. (Cited on pages xiii and 20.)

[15] BILINSKI, P., CORVEE, E., BAK, S., AND BREMOND, F. Relative dense tracklets for human action recognition. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on* (2013), IEEE, pp. 1–7. (Cited on page 28.)

[16] BILINSKI, P., KOPERSKI, M., BAK, S., AND BREMOND, F. Representing visual appearance by video brownian covariance descriptor for human action recognition. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on* (2014), IEEE, pp. 87–92. (Cited on page 28.)

[17] BLEI, D. M., AND LAFFERTY, J. D. Correlated topic models. In *Proceedings of the 18th International Conference on Neural Information Processing Systems* (2005), MIT Press, pp. 147–154. (Cited on page 24.)

[18] BLEI, D. M., NG, A. Y., AND JORDAN, M. I. Latent dirichlet allocation. *Journal of machine Learning research 3*, Jan (2003), 993–1022. (Cited on page 24.)

[19] BOBICK, A. F., AND DAVIS, J. W. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 3 (2001), 257–267. (Cited on pages xiii, 22 and 23.)

[20] BOBICK, A. F., AND WILSON, A. D. A state-based approach to the representation and recognition of gesture. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 19*, 12 (1997), 1325–1337. (Cited on pages 20, 31 and 34.)

[21] BRAND, M., AND KETTNAKER, V. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (2000), 844–851. (Cited on page 34.)

[22] BREGONZIO, M., GONG, S., AND XIANG, T. Recognising action as clouds of space-time interest points. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 1948–1955. (Cited on pages 19, 21 and 27.)

[23] BRENDEL, W., FERN, A., AND TODOROVIC, S. Probabilistic event logic for interval-based event recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 3329–3336. (Cited on page 44.)

[24] Bρίγκας, M. Human activity recognition using conditional random fields and privileged information. (Cited on page 17.)

[25] BRULIN, D., BENEZETH, Y., AND COURTIAL, E. Posture recognition based on fuzzy logic for home monitoring of the elderly. *IEEE transactions on information technology in biomedicine 16*, 5 (2012), 974–982. (Cited on page 160.)

[26] BULAT, A., AND TZIMIROPOULOS, G. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision* (2016), Springer, pp. 717–732. (Cited on page 164.)

[27] C. WONG, C. H. Healthcare in singapore: Challenges and management. *Jpn. Med. Assoc. J. JMAJ 51*, 5 (2008), 343–346. (Cited on page 159.)

[28] CALDERARA, S., CUCCHIARA, R., AND PRATI, A. Detection of abnormal behaviors using a mixture of von mises distributions. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on* (2007), IEEE, pp. 141–146. (Cited on page 165.)

[29] CAO, L., LUO, J., LIANG, F., AND HUANG, T. S. Heterogeneous feature machines for visual recognition. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 1095–1102. (Cited on page 24.)

[30] CAO, Y., TAO, L., AND XU, G. An event-driven context model in elderly health monitoring. In *Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC'09. Symposia and Workshops on* (2009), IEEE, pp. 120–124. (Cited on pages 42 and 44.)

[31] CARREIRA, J., AND ZISSERMAN, A. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on* (2017), IEEE, pp. 4724–4733. (Cited on page 30.)

[32] CATANI, M., ET AL. The rises and falls of disconnection syndromes. *Brain 128*, 10 (2005), 2224–2239. (Cited on page 160.)

[33] CEUSTERS, W., CORSO, J. J., FU, Y., PETROPOULOS, M., AND KROVI, V. N. Introducing ontological realism for semi-supervised detection and annotation of operationally significant activity in surveillance videos. In *STIDS* (2010), pp. 13–20. (Cited on page 44.)

[34] CHAI, X., LI, G., LIN, Y., XU, Z., TANG, Y., CHEN, X., AND ZHOU, M. Sign language recognition and translation with kinect. In *IEEE Conf. on AFGR* (2013). (Cited on page 163.)

[35] CHAKRABORTY, B., HOLTE, M. B., MOESLUND, T. B., GONZALEZ, J., AND ROCA, F. X. A selective spatio-temporal interest point detector for human action recognition in complex scenes. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 1776–1783. (Cited on page 163.)

[36] CHANDRA, S. R., ISSAC, T. G., AND ABBAS, M. M. Apraxias in neurodegenerative dementias. *Indian journal of psychological medicine 37*, 1 (2015), 42. (Cited on page 160.)

[37] CHANG, C.-C., AND LIN, C.-J. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST) 2*, 3 (2011), 27. (Cited on page 59.)

[38] CHAU, D. P., BADIE, J., BREMOND, F., AND THONNAT, M. Online Tracking Parameter Adaptation based on Evaluation. In *IEEE International Conference on Advanced Video and Signal-based Surveillance* (Krakow, Pologne, Aug. 2013). (Cited on page 78.)

[39] CHEN, H., WANG, Q., AND CAO, L. Design of the workstation for hand rehabilitation based on data glove. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2010 IEEE International Conference on* (2010), IEEE, pp. 769–771. (Cited on page 163.)

[40] CHEN, L., HOEY, J., NUGENT, C. D., COOK, D. J., AND YU, Z. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42*, 6 (2012), 790–808. (Cited on page 42.)

[41] CHEN, L., NUGENT, C., AND OKEYO, G. An ontology-based hybrid approach to activity modeling for smart homes. *IEEE Transactions on human-machine systems 44*, 1 (2014), 92–105. (Cited on page 44.)

[42] CHENG, G., WAN, Y., SAUDAGAR, A. N., NAMUDURI, K., AND BUCKLES, B. P. Advances in human action recognition: A survey. *arXiv preprint arXiv:1501.05964* (2015). (Cited on page 30.)

[43] CHÉRON, G., LAPTEV, I., AND SCHMID, C. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 3218–3226. (Cited on pages 29 and 164.)

[44] CISCO, V. Cisco visual networking index: Forecast and methodology 2016–2021.(2017), 2017. (Cited on page 4.)

[45] CRISPIM-JUNIOR, C. F., BATHRINARAYANAN, V., FOSTY, B., ROMDHANE, R., KONIG, A., THONNAT, M., AND BREMOND, F. Evaluation of a Monitoring System for Event Recognition of Older People. In *International Conference on Advanced Video and Signal-Based Surveillance 2013* (Krakow, Poland, Aug. 2013), pp. 165 – 170. (Cited on pages xix, xx, 42, 80, 101, 149, 150 and 151.)

[46] CRISPIM-JUNIOR, C. F., GÓMEZ URÍA, A., STRUMIA, C., KOPERSKI, M., KÖNIG, A., NEGIN, F., COSAR, S., NGHIEM, A. T., CHAU, D. P., CHARPIAT, G., ET AL. Online recognition of daily activities by color-depth sensing and knowledge models. *Sensors 17*, 7 (2017), 1528. (Cited on pages 44 and 80.)

[47] DAI, P., DI, H., DONG, L., TAO, L., AND XU, G. Group interaction analysis in dynamic context. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39*, 1 (2009), 34–42. (Cited on pages 20 and 39.)

[48] DALAL, N., AND TRIGGS, B. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.* (June 2005), vol. 1, pp. 886–893 vol. 1. (Cited on pages 27, 28, 51 and 92.)

[49] DAMEN, D., AND HOGG, D. Recognizing linked events: Searching the space of feasible explanations. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 927–934. (Cited on page 40.)

[50] DARRELL, T., AND PENTLAND, A. Space-time gestures. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on* (1993), IEEE, pp. 335–340. (Cited on pages 20 and 31.)

[51] DATASET, D. Centre Hospitalier Universitaire de Nice (CHU). https://team.inria.fr/stars/demcare-chu-dataset/, 2012. [Online; accessed 19-July-2012]. (Cited on page 15.)

[52] DEE, H. M., COHN, A. G., AND HOGG, D. C. Building semantic scene models from unconstrained video. *Computer Vision and Image Understanding 116*, 3 (2012), 446–456. (Cited on page 165.)

[53] DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K., AND FEI-FEI, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09* (2009). (Cited on pages 54 and 169.)

[54] DOLLÁR, P., RABAUD, V., COTTRELL, G., AND BELONGIE, S. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on* (2005), IEEE, pp. 65–72. (Cited on pages 26 and 51.)

[55] DONAHUE, J., ANNE HENDRICKS, L., GUADARRAMA, S., ROHRBACH, M., VENUGOPALAN, S., SAENKO, K., AND DARRELL, T. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 2625–2634. (Cited on page 165.)

[56] DOWLING, A. V., BARZILAY, O., LOMBROZO, Y., AND WOLF, A. An adaptive home-use robotic rehabilitation system for the upper body. *IEEE journal of translational engineering in health and medicine 2* (2014), 1–10. (Cited on page 163.)

[57] DUCHENNE, O., LAPTEV, I., SIVIC, J., BACH, F., AND PONCE, J. Automatic annotation of human actions in video. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 1491–1498. (Cited on page 165.)

[58] DURAND, T., MORDAN, T., THOME, N., AND CORD, M. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)* (2017), vol. 2. (Cited on page 164.)

[59] DURAND, T., THOME, N., AND CORD, M. Weldon: Weakly supervised learning of deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 4743–4752. (Cited on page 164.)

[60] EARLEY, J. An efficient context-free parsing algorithm. *Communications of the ACM 13*, 2 (1970), 94–102. (Cited on page 41.)

[61] EFROS, A. A., BERG, A. C., MORI, G., AND MALIK, J. Recognizing action at a distance. In *null* (2003), IEEE, p. 726. (Cited on page 31.)

[62] ELLOUMI, S., COŞAR, S., PUSIOL, G., BREMOND, F., AND THONNAT, M. Unsupervised discovery of human activities from long-time videos. *IET Computer Vision* (2014). (Cited on pages 110, 128, 129, 130, 131, 135, 146, 147 and 149.)

[63] EMONET, R., VARADARAJAN, J., AND ODOBEZ, J.-M. Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 3233–3240. (Cited on page 91.)

[64] EMONET, R., VARADARAJAN, J., AND ODOBEZ, J.-M. Temporal Analysis of Motif Mixtures using Dirichlet Processes. *PAMI* (2014). (Cited on pages 28 and 29.)

[65] EMONET, R., VARADARAJAN, J., AND ODOBEZ, J.-M. Temporal analysis of motif mixtures using dirichlet processes. *IEEE transactions on pattern analysis and machine intelligence 36*, 1 (2014), 140–156. (Cited on page 165.)

[66] ESCALERA, S., GONZÀLEZ, J., BARÓ, X., REYES, M., GUYON, I., ATHITSOS, V., ESCALANTE, H., SIGAL, L., ARGYROS, A., SMINCHISESCU, C., ET AL. Chalearn multi-modal gesture recognition 2013: grand challenge and workshop summary. In *Proceedings of the 15th ACM on International conference on multimodal interaction* (2013), ACM, pp. 365–368. (Cited on page 164.)

[67] F. NEGIN, J. BOURGEOIS, P. R. F. B. A gesture recognition framework for cognitive assessment. *Gerontechnology 17* (2018), 169s. (Cited on page 14.)

[68] FEICHTENHOFER, C., PINZ, A., AND ZISSERMAN, A. Convolutional two-stream network fusion for video action recognition. (Cited on page 29.)

[69] GAO, Q., AND SUN, S. Trajectory-based human activity recognition with hierarchical dirichlet process hidden markov models. In *Proceedings of the 1st IEEE China Summit and International Conference on Signal and Information Processing* (2013). (Cited on page 109.)

[70] GAO, Q., AND SUN, S. Trajectory-based human activity recognition with hierarchical dirichlet process hidden markov models. In *Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on* (2013), IEEE, pp. 456–460. (Cited on page 165.)

[71] GARG, S., SINGH, R., AND GROVER, M. India's health workforce: current status and the way forward. *National medical journal of India 25*, 2 (2012), 111. (Cited on page 159.)

[72] GAVRILA, D. M., DAVIS, L. S., ET AL. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *International workshop on automatic face-and gesture-recognition* (1995), Citeseer, pp. 272–277. (Cited on pages 20 and 31.)

[73] GE, L., LIANG, H., YUAN, J., AND THALMANN, D. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 3593–3601. (Cited on page 164.)

[74] GHANEM, N., DEMENTHON, D., DOERMANN, D., AND DAVIS, L. Representation and recognition of events in surveillance video using petri nets. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on* (2004), IEEE, pp. 112–112. (Cited on page 43.)

[75] GILBERT, A., ILLINGWORTH, J., AND BOWDEN, R. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 925–931. (Cited on pages 19 and 27.)

[76] GONG, D., MEDIONI, G., AND ZHAO, X. Structured time series analysis for human action segmentation and recognition. *IEEE transactions on pattern analysis and machine intelligence 36*, 7 (2014), 1414–1427. (Cited on page 164.)

[77] GONG, S., AND XIANG, T. Recognition of group activities using dynamic probabilistic networks. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (2003), IEEE, pp. 742–749. (Cited on pages 20 and 39.)

[78] GUO, K., ISHWAR, P., AND KONRAD, J. Action recognition in video by covariance matching of silhouette tunnels. In *Computer Graphics and Image Processing (SIBGRAPI), 2009 XXII Brazilian Symposium on* (2009), IEEE, pp. 299–306. (Cited on page 24.)

[79] HAMID, R., MADDI, S., JOHNSON, A., BOBICK, A., ESSA, I., AND ISBELL, C. A novel sequence representation for unsupervised analysis of human activities. *Artificial Intelligence 173*, 14 (2009), 1221–1244. (Cited on pages 80 and 91.)

[80] HAN, L., WU, X., LIANG, W., HOU, G., AND JIA, Y. Discriminative human action recognition in the learned hierarchical manifold space. *Image and Vision Computing 28*, 5 (2010), 836–849. (Cited on page 40.)

[81] HEILMAN KM, R. L. Apraxia. *Clinical Neuropsychology 128*, 10 (2003), 215–235. (Cited on page 160.)

[82] HOBBS, J., NEVATIA, R., AND BOLLES, B. An ontology for video event representation. In *IEEE Workshop on Event Detection and Recognition* (2004), p. 119. (Cited on pages 20 and 42.)

[83] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780. (Cited on pages 161 and 172.)

[84] HOLTE, M. B., MOESLUND, T. B., NIKOLAIDIS, N., AND PITAS, I. 3d human action recognition for multi-view camera systems. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on* (2011), IEEE, pp. 342–349. (Cited on pages 19, 21 and 27.)

[85] HU, J.-F., ZHENG, W.-S., LAI, J., GONG, S., AND XIANG, T. Exemplar-based recognition of human–object interactions. *IEEE Transactions on Circuits and Systems for Video Technology 26*, 4 (2016), 647–660. (Cited on page 32.)

[86] HU, W., XIAO, X., FU, Z., XIE, D., TAN, T., AND MAYBANK, S. A system for learning statistical motion patterns. *IEEE transactions on pattern analysis and machine intelligence 28*, 9 (2006), 1450–1464. (Cited on pages 32, 33, 79, 108 and 165.)

[87] HU, W., XIAO, X., FU, Z., XIE, D., TAN, T., AND MAYBANK, S. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 9 (2006), 1450–1464. (Cited on pages 109 and 135.)

[88] HU, Y., CAO, L., LV, F., YAN, S., GONG, Y., AND HUANG, T. S. Action detection in complex scenes with spatial and temporal ambiguities. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 128–135. (Cited on pages 19, 21, 22 and 23.)

[89] IJSSELMUIDEN, J., AND STIEFELHAGEN, R. Towards high-level human activity recognition through computer vision and temporal logic. In *Annual Conference on Artificial Intelligence* (2010), Springer, pp. 426–435. (Cited on page 43.)

[90] IKIZLER, N., AND DUYGULU, P. Histogram of oriented rectangles: A new pose descriptor for human action recognition. *Image and Vision Computing 27*, 10 (2009), 1515–1526. (Cited on page 23.)

[91] IKIZLER-CINBIS, N., AND SCLAROFF, S. Object, scene and actions: Combining multiple features for human action recognition. In *European conference on computer vision* (2010), Springer, pp. 494–507. (Cited on pages 19, 21 and 27.)

[92] INSAFUTDINOV, E., PISHCHULIN, L., ANDRES, B., ANDRILUKA, M., AND SCHIELE, B. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision* (2016), Springer, pp. 34–50. (Cited on pages 19, 21 and 25.)

[93] INTILLE, S. S., AND BOBICK, A. F. A framework for recognizing multi-agent action from visual evidence. *AAAI/IAAI 99*, 518–525 (1999). (Cited on page 43.)

[94] IVANOV, Y. A., AND BOBICK, A. F. Recognition of visual activities and interactions by stochastic parsing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 8 (2000), 852–872. (Cited on pages 20 and 41.)

[95] JAIN, M., VAN GEMERT, J., JÉGOU, H., BOUTHEMY, P., AND SNOEK, C. G. Action localization with tubelets from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 740–747. (Cited on page 165.)

[96] JENKINS, N. 245 million video surveillance cameras installed globally in 2014. *IHS Technology* (2015). (Cited on page 5.)

[97] JI, S., XU, W., YANG, M., AND YU, K. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence 35*, 1 (2013), 221–231. (Cited on pages 29 and 164.)

[98] JIA, Y., SHELHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J., GIRSHICK, R., GUADARRAMA, S., AND DARRELL, T. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093* (2014). (Cited on page 175.)

[99] JOHANSSON, G. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics 14*, 2 (1973), 201–211. (Cited on pages 24 and 76.)

[100] JOO, S.-W., AND CHELLAPPA, R. Attribute grammar-based event recognition and anomaly detection. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on* (2006), IEEE, pp. 107–107. (Cited on pages 20 and 41.)

[101] KADIR, T., AND BRADY, M. Saliency, scale and image description. *International Journal of Computer Vision 45*, 2 (2001), 83–105. (Cited on page 50.)

[102] KARAKOSTAS, A., BRIASSOULI, A., AVGERINAKIS, K., KOMPATSIARIS, I., AND M., T. The dem@care experiments and datasets: a technical report. Tech. rep., 2014. (Cited on pages 15, 48 and 60.)

[103] KATZ, S. Studies of illness in the aged. the index of adl: a standardized measure of biologic and psychologic function. *JaMa 185* (1963), 94–99. (Cited on page 18.)

[104] KE, Y., SUKTHANKAR, R., AND HEBERT, M. Spatio-temporal shape and flow correlation for action recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8. (Cited on pages 19, 20, 21 and 22.)

[105] KELLOKUMPU, V., ZHAO, G., AND PIETIKÄINEN, M. Recognition of human actions using texture descriptors. *Machine Vision and Applications 22*, 5 (2011), 767–780. (Cited on page 34.)

[106] KHADEMI, M., MOUSAVI HONDORI, H., MCKENZIE, A., DODAKIAN, L., LOPES, C. V., AND CRAMER, S. C. Free-hand interaction with leap motion controller for stroke rehabilitation. In *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems* (2014), ACM, pp. 1663–1668. (Cited on page 161.)

[107] KIM, T.-K., AND CIPOLLA, R. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 8 (2009), 1415–1428. (Cited on page 24.)

[108] KIM, W., LEE, J., KIM, M., OH, D., AND KIM, C. Human action recognition using ordinal measure of accumulated motion. *EURASIP Journal on Advances in Signal Processing 2010*, 1 (2010), 219190. (Cited on page 23.)

[109] KINGMA, D., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). (Cited on page 176.)

[110] KITANI, K. M., SATO, Y., AND SUGIMOTO, A. Recovering the basic structure of human activities from noisy video-based symbol strings. *International Journal of Pattern Recognition and Artificial Intelligence 22*, 08 (2008), 1621–1646. (Cited on pages 20 and 42.)

[111] KLASER, A., MARSZAŁEK, M., AND SCHMID, C. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference* (2008), British Machine Vision Association, pp. 275–1. (Cited on pages 51 and 163.)

[112] KOLLER, O., NEY, H., AND BOWDEN, R. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV, USA, June 2016), pp. 3793–3802. (Cited on page 175.)

[113] KÖNIG, A., CRISPIM-JUNIOR, C. F., URIA, A. G., COVELLA, F. B., DERREUMAUX, A., BENSADOUN, G., DAVID, R., VERHEY, F., AALTEN, P., AND ROBERT, P. Ecological assessment of autonomy in instrumental activities of daily living in dementia patients by the means of an automatic video monitoring system. *ICT for assessment and rehabilitation in Alzheimer's disease and related disorders* (2016), 29. (Cited on page 160.)

[114] KOPERSKI, M., BILINSKI, P., AND BREMOND, F. 3D Trajectories for Action Recognition. In *ICIP - The 21st IEEE International Conference on Image Processing* (Paris, France, Oct. 2014), IEEE. (Cited on page 28.)

[115] KOPERSKI, M., AND BREMOND, F. Modeling Spatial Layout of Features for Real World Scenario RGB-D Action Recognition. In *AVSS 2016* (Colorado Springs, United States, Aug. 2016), pp. 44 – 50. (Cited on page 28.)

[116] KOPPULA, H. S., GUPTA, R., AND SAXENA, A. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research 32*, 8 (2013), 951–970. (Cited on page 26.)

[117] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105. (Cited on page 164.)

[118] KUEHNE, H., JHUANG, H., GARROTE, E., POGGIO, T., AND SERRE, T. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 2556–2563. (Cited on page 165.)

[119] LAPTEV, I. On space-time interest points. *International journal of computer vision 64*, 2-3 (2005), 107–123. (Cited on pages 26, 27, 50, 163 and 165.)

[120] LAPTEV, I., MARSZALEK, M., SCHMID, C., AND ROZENFELD, B. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.* (2008), IEEE, pp. 1–8. (Cited on pages 48 and 92.)

[121] LAPTEV, I., AND PÉREZ, P. Retrieving actions in movies. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), IEEE, pp. 1–8. (Cited on page 165.)

[122] LAVEE, G., RIVLIN, E., AND RUDZSKY, M. Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 39*, 5 (2009), 489–504. (Cited on page 87.)

[123] LAWTON, M. P., AND BRODY, E. M. Assessment of older people: self-maintaining and instrumental activities of daily living. *The gerontologist 9*, 3_Part_1 (1969), 179–186. (Cited on page 18.)

[124] LAZEBNIK, S., SCHMID, C., AND PONCE, J. A sparse texture representation using affine-invariant regions. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 2, IEEE, pp. II–II. (Cited on page 55.)

[125] LE, Q. V., ZOU, W. Y., YEUNG, S. Y., AND NG, A. Y. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 3361–3368. (Cited on page 164.)

[126] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324. (Cited on page 161.)

[127] LEI, J., LI, G., ZHANG, J., GUO, Q., AND TU, D. Continuous action segmentation and recognition using hybrid convolutional neural network-hidden markov model model. *IET Computer Vision 10*, 6 (2016), 537–544. (Cited on page 36.)

[128] LEIGHTLEY, D., LI, B., MCPHEE, J. S., YAP, M. H., AND DARBY, J. Exemplar-based human action recognition with template matching from a stream of motion capture. In *International Conference Image Analysis and Recognition* (2014), Springer, pp. 12–20. (Cited on page 31.)

[129] LI, W., ZHANG, Z., AND LIU, Z. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on* (2010), IEEE, pp. 9–14. (Cited on pages 52 and 167.)

[130] LIN, Z., JIANG, Z., AND DAVIS, L. S. Recognizing actions by shape-motion prototype trees. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 444–451. (Cited on page 31.)

[131] LIU, C., AND YUEN, P. C. Human action recognition using boosted eigenactions. *Image and vision computing 28*, 5 (2010), 825–835. (Cited on page 24.)

[132] LIU, J., SHAHROUDY, A., XU, D., AND WANG, G. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision* (2016), Springer, pp. 816–833. (Cited on page 37.)

[133] LIU, L., SHAO, L., ZHENG, F., AND LI, X. Realistic action recognition via sparsely-constructed gaussian processes. *Pattern Recognition 47*, 12 (2014), 3819–3827. (Cited on page 163.)

[134] LOPES, O., REYES, M., ESCALERA, S., AND GONZÀLEZ, J. Spherical blurred shape model for 3-d object and pose recognition: Quantitative analysis and hci applications in smart environments. *IEEE Transactions on Cybernetics* (2014), 1–1. (Cited on page 163.)

[135] LUBLINERMAN, R., OZAY, N., ZARPALAS, D., AND CAMPS, O. Activity recognition from silhouettes using linear systems and model (in) validation techniques. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on Pattern Analysis* (2006), vol. 1, IEEE, pp. 347–350. (Cited on page 31.)

[136] LUCAS, B. D., KANADE, T., ET AL. An iterative image registration technique with an application to stereo vision. (Cited on page 25.)

[137] LUO, J., WANG, W., AND QI, H. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 1809–1816. (Cited on page 164.)

[138] LV, F., AND NEVATIA, R. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *European conference on computer vision* (2006), Springer, pp. 359–372. (Cited on page 26.)

[139] LV, F., AND NEVATIA, R. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. *Computer Vision–ECCV 2006* (2006), 359–372. (Cited on page 164.)

[140] LV, F., AND NEVATIA, R. Single view human action recognition using key pose matching and viterbi path searching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8. (Cited on pages xiii and 35.)

[141] MA, S., SIGAL, L., AND SCLAROFF, S. Learning activity progression in lstms for activity detection and early detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 1942–1950. (Cited on page 29.)

[142] MA, S., ZHANG, J., IKIZLER-CINBIS, N., AND SCLAROFF, S. Action recognition and localization by hierarchical space-time segments. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 2744–2751. (Cited on page 165.)

[143] MA, S., ZHANG, J., SCLAROFF, S., IKIZLER-CINBIS, N., AND SIGAL, L. Space-time tree ensemble for action recognition and localization. *International Journal of Computer Vision 126*, 2-4 (2018), 314–332. (Cited on page 26.)

[144] MACQUEEN, J., ET AL. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (1967), vol. 1, California, USA, p. 14. (Cited on page 82.)

[145] MAKRIS, D., AND ELLIS, T. Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 35*, 3 (2005), 397–408. (Cited on pages 32, 79, 80 and 108.)

[146] MARSZALEK, M., LAPTEV, I., AND SCHMID, C. Actions in context. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 2929–2936. (Cited on page 165.)

[147] MAUTHNER, T., ROTH, P. M., AND BISCHOF, H. Temporal feature weighting for prototype-based action recognition. In *Asian Conference on Computer Vision* (2010), Springer, pp. 566–579. (Cited on page 40.)

[148] MESSING, R., PAL, C., AND KAUTZ, H. Activity recognition using the velocity histories of tracked keypoints. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 104–111. (Cited on page 25.)

[149] MINNEN, D., ESSA, I., AND STARNER, T. Expectation grammars: Leveraging high-level expectations for activity recognition. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 2, IEEE, pp. II–626. (Cited on pages 20 and 41.)

[150] MOESLUND, T. B., HILTON, A., AND KRÜGER, V. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding 104*, 2 (2006), 90–126. (Cited on page 8.)

[151] MOORE, D., AND ESSA, I. Recognizing multitasked activities from video using stochastic context-free grammar. In *Proceedings of the National Conference on Artificial Intelligence* (2002), Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, pp. 770–776. (Cited on pages 20, 41, 42 and 80.)

[152] MORARIU, V. I., AND DAVIS, L. S. Multi-agent event recognition in structured scenarios. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 3289–3296. (Cited on pages 20 and 43.)

[153] MORI, G., BELONGIE, S., AND MALIK, J. Shape contexts enable efficient retrieval of similar shapes. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (2001), vol. 1, IEEE, pp. I–I. (Cited on page 55.)

[154] MORRIS, B., AND TRIVEDI, M. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 11 (Nov 2011), 2287–2301. (Cited on pages 109 and 135.)

[155] MORRIS, B. T., AND TRIVEDI, M. M. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *IEEE transactions on pattern analysis and machine intelligence 33*, 11 (2011), 2287–2301. (Cited on pages 86 and 165.)

[156] MURRAY, C. D., PETTIFER, S., HOWARD, T., PATCHICK, E. L., CAILLETTE, F., KULKARNI, J., AND BAMFORD, C. The treatment of phantom limb pain using immersive virtual reality: three case studies. *Disability and rehabilitation 29*, 18 (2007), 1465–1469. (Cited on page 163.)

[157] NATARAJAN, P., AND NEVATIA, R. Coupled hidden semi markov models for activity recognition. In *Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on* (2007), IEEE, pp. 10–10. (Cited on page 35.)

[158] NATIONS, U. World population ageing: 1950-2050. *New York: Department of Economic and Social Affairs* (2002). (Cited on page 6.)

[159] NEGIN, F., AKGÜL, C. B., YÜKSEL, K. A., AND ERÇIL, A. An rdf-based action recognition framework with feature selection capability, considering therapy exercises utilizing depth cameras. *Journal of Theoretical and Applied Computer Science 8*, 3 (2014), 3–22. (Cited on page 26.)

[160] NEGIN, F., BOURGEOIS, J., CHAPOULIE, E., ROBERT, P., AND BREMOND, F. Praxis and gesture recognition. In *The 10th World Conference of Gerontechnology (ISG 2016)* (2016). (Cited on page 14.)

[161] NEGIN, F., AND BREMOND, F. Human action recognition in videos: A survey. Tech. rep., INRIA Technical Report, 2016. (Cited on page 17.)

[162] NEGIN, F., COGAR, S., BREMOND, F., AND KOPERSKI, M. Generating unsupervised models for online long-term daily living activity recognition. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on* (2015), IEEE, pp. 186–190. (Cited on pages 12, 13 and 160.)

[163] NEGIN, F., KOPERSKI, M., CRISPIM, C. F., BREMOND, F., COŞAR, S., AND AVGERINAKIS, K. A hybrid framework for online recognition of activities of daily living in real-world settings. In *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on* (2016), IEEE, pp. 37–43. (Cited on pages 12 and 13.)

[164] NEGIN, F., ÖZDEMIR, F., AKGÜL, C. B., YÜKSEL, K. A., AND ERÇIL, A. A decision forest based feature selection framework for action recognition from rgb-depth cameras. In *International Conference Image Analysis and Recognition* (2013), Springer, pp. 648–657. (Cited on pages 26 and 164.)

[165] NEGIN, F., RODRIGUEZ, P., KOPERSKI, M., KERBOUA, A., GONZÀLEZ, J., BOURGEOIS, J., CHAPOULIE, E., ROBERT, P., AND BREMOND, F. Praxis: Towards automatic cognitive assessment using gesture recognition. *Expert Systems with Applications 106* (2018), 21–35. (Cited on pages 13 and 29.)

[166] NGHIEM, A.-T., AUVINET, E., AND MEUNIER, J. Head detection using kinect camera and its application to fall detection. In *ISSPA* (2012), pp. 164–169. (Cited on page 78.)

[167] OBERWEGER, M., WOHLHART, P., AND LEPETIT, V. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807* (2015). (Cited on page 164.)

[168] OFLI, F., CHAUDHRY, R., KURILLO, G., VIDAL, R., AND BAJCSY, R. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation 25*, 1 (2014), 24–38. (Cited on page 26.)

[169] OIKONOMOPOULOS, A., PANTIC, M., AND PATRAS, I. Sparse b-spline polynomial descriptors for human activity recognition. *Image and vision computing 27*, 12 (2009), 1814–1825. (Cited on pages 19 and 27.)

[170] OLIVER, N., HORVITZ, E., AND GARG, A. Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on* (2002), IEEE, pp. 3–8. (Cited on pages 20 and 39.)

[171] OLIVER, N. M., ROSARIO, B., AND PENTLAND, A. P. A bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern analysis and machine intelligence 22*, 8 (2000), 831–843. (Cited on page 34.)

[172] ONEATA, D., VERBEEK, J., AND SCHMID, C. Action and event recognition with fisher vectors on a compact feature set. In *Computer Vision (ICCV), 2013 IEEE International Conference on* (Dec 2013), pp. 1817–1824. (Cited on page 131.)

[173] OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on systems, man, and cybernetics 9*, 1 (1979), 62–66. (Cited on page 169.)

[174] OZBULAK, G., AYTAR, Y., AND EKENEL, H. K. How transferable are cnn-based features for age and gender classification? In *Biometrics Special Interest Group (BIOSIG), 2016 International Conference of the* (2016), IEEE, pp. 1–6. (Cited on page 174.)

[175] PARK, S., AND AGGARWAL, J. K. A hierarchical bayesian network for event recognition of human actions and interactions. *Multimedia systems 10*, 2 (2004), 164–179. (Cited on pages 20, 21, 34 and 35.)

[176] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., ET AL. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research 12*, Oct (2011), 2825–2830. (Cited on page 59.)

[177] PEIGNEUX, P., VAN DER LINDEN, M., AND LE GALL, D. Evaluation des apraxies gestuelles. *L'apraxie, 2* (2003), 133–138. (Cited on page 160.)

[178] PELLEG, D., MOORE, A. W., ET AL. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml* (2000), vol. 1, pp. 727–734. (Cited on page 84.)

[179] PEREIRA, C. R., PEREIRA, D. R., SILVA, F. A., MASIEIRO, J. P., WEBER, S. A., HOOK, C., AND PAPA, J. P. A new computer vision-based approach to aid the diagnosis of parkinson's disease. *Computer Methods and Programs in Biomedicine 136* (2016), 79–88. (Cited on page 161.)

[180] PERRONNIN, F., LIU, Y., SÁNCHEZ, J., AND POIRIER, H. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (2010), IEEE, pp. 3384–3391. (Cited on pages 56, 58 and 171.)

[181] PERRONNIN, F., SÁNCHEZ, J., AND MENSINK, T. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision* (2010), Springer, pp. 143–156. (Cited on pages 56, 58 and 171.)

[182] PICIARELLI, C., AND FORESTI, G. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters 27*, 15 (2006), 1835–1842. (Cited on pages 32, 79 and 108.)

[183] PIGOU, L., DIELEMAN, S., KINDERMANS, P.-J., AND SCHRAUWEN, B. Sign language recognition using convolutional neural networks. In *Workshop at the European Conference on Computer Vision* (2014), Springer, pp. 572–578. (Cited on page 163.)

[184] PINHANEZ, C. S., AND BOBICK, A. F. Human action detection using pnf propagation of temporal constraints. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on* (1998), IEEE, pp. 898–904. (Cited on page 43.)

[185] PIRSIAVASH, H., AND RAMANAN, D. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (2012), IEEE, pp. 2847–2854. (Cited on page 160.)

[186] PISHCHULIN, L., INSAFUTDINOV, E., TANG, S., ANDRES, B., ANDRILUKA, M., GEHLER, P., AND SCHIELE, B. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016). (Cited on pages 19, 21, 25, 76 and 167.)

[187] PUSIOL, G. *Discovery of human activities in video*. Phd, Institut National de Recherche en Informatique et en Automatique (INRIA), May 2012. (Cited on pages 17 and 89.)

[188] QIAN, H., MAO, Y., XIANG, W., AND WANG, Z. Recognition of human activities using svm multi-class classifier. *Pattern Recognition Letters 31*, 2 (2010), 100–111. (Cited on pages 19, 21 and 22.)

[189] RANTZ, M. J., BANERJEE, T. S., CATTOOR, E., SCOTT, S. D., SKUBIC, M., AND POPESCU, M. Automated fall detection with quality improvement ârewindâ to reduce falls in hospital rooms. *Journal of gerontological nursing 40*, 1 (2013), 13–17. (Cited on pages 20 and 44.)

[190] RAPTIS, M., KIROVSKI, D., AND HOPPE, H. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation* (2011), ACM, pp. 147–156. (Cited on page 163.)

[191] RODRIGUEZ, M. D., AHMED, J., AND SHAH, M. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer*

Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (2008), IEEE, pp. 1–8. (Cited on page 22.)

[192] RODRIGUEZ, P., CUCURULL, G., GONZÀLEZ, J., GONFAUS, J. M., NASROLLAHI, K., MOESLUND, T. B., AND ROCA, F. X. Deep pain: Exploiting long short-term memory networks for facial expression classification. *IEEE Transactions on Cybernetics* (2017). (Cited on pages 165, 172, 174 and 176.)

[193] ROH, M.-C., SHIN, H.-K., AND LEE, S.-W. View-independent human action recognition with volume motion template on single stereo camera. *Pattern Recognition Letters 31*, 7 (2010), 639–647. (Cited on pages 19, 21, 22 and 23.)

[194] RYOO, M. S., AND AGGARWAL, J. K. Recognition of composite human activities through context-free grammar based representation. In *Computer vision and pattern recognition, 2006 ieee computer society conference on* (2006), vol. 2, IEEE, pp. 1709–1718. (Cited on pages xiii, 20, 42 and 43.)

[195] SADEK, S., AL-HAMADI, A., MICHAELIS, B., AND SAYED, U. An action recognition scheme using fuzzy log-polar histogram and temporal self-similarity. *EURASIP Journal on Advances in Signal Processing 2011*, 1 (2011), 540375. (Cited on page 27.)

[196] SALTON, G., AND MCGILL, M. J. Introduction to modern information retrieval. (Cited on page 56.)

[197] SALTON, G., AND MICHAEL, J. Mcgill. *Introduction to modern information retrieval* (1983), 24–51. (Cited on page 55.)

[198] SCHÖNAUER, C., PINTARIC, T., KAUFMANN, H., JANSEN-KOSTERINK, S., AND VOLLENBROEK-HUTTEN, M. Chronic pain rehabilitation with a serious game using multimodal input. In *Virtual Rehabilitation (ICVR), 2011 International Conference on* (2011), IEEE, pp. 1–8. (Cited on page 163.)

[199] SCHULTZ, M., GILL, J., ZUBAIRI, S., HUBER, R., AND GORDIN, F. Bacterial contamination of computer keyboards in a teaching hospital. *Infection Control & Hospital Epidemiology 24*, 04 (2003), 302–303. (Cited on page 163.)

[200] SCOVANNER, P., ALI, S., AND SHAH, M. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia* (2007), ACM, pp. 357–360. (Cited on page 27.)

[201] SEMPENA, S., MAULIDEVI, N. U., AND ARYAN, P. R. Human action recognition using dynamic time warping. In *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on* (2011), IEEE, pp. 1–5. (Cited on page 26.)

[202] SHAHROUDY, A., LIU, J., NG, T.-T., AND WANG, G. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 1010–1019. (Cited on pages 20, 21 and 36.)

[203] SHAO, L., ZHEN, X., TAO, D., AND LI, X. Spatio-temporal laplacian pyramid coding for action recognition. *IEEE Transactions on Cybernetics 44*, 6 (2014), 817–827. (Cited on page 163.)

[204] SHECHTMAN, E., AND IRANI, M. Space-time behavior based correlation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (2005), vol. 1, IEEE, pp. 405–412. (Cited on pages 19, 20, 21 and 22.)

[205] SHEIKH, Y., SHEIKH, M., AND SHAH, M. Exploring the space of a human action. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 1, IEEE, pp. 144–149. (Cited on pages 19, 21 and 25.)

[206] SHI, J., AND TOMASI, C. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on* (1994), IEEE, pp. 593–600. (Cited on page 50.)

[207] SHI, Q., CHENG, L., WANG, L., AND SMOLA, A. Human action segmentation and recognition using discriminative semi-markov models. *International journal of computer vision 93*, 1 (2011), 22–32. (Cited on page 36.)

[208] SHI, Y., HUANG, Y., MINNEN, D., BOBICK, A., AND ESSA, I. Propagation networks for recognition of partially ordered sequential action. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on* (2004), vol. 2, IEEE, pp. II–II. (Cited on page 40.)

[209] SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), Ieee, pp. 1297–1304. (Cited on pages 25 and 76.)

[210] SIMONYAN, K., AND ZISSERMAN, A. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems* (2014), pp. 568–576. (Cited on pages 29 and 54.)

[211] SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014). (Cited on pages 54, 169 and 174.)

[212] SISKIND, J. M. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of artificial intelligence research 15* (2001), 31–90. (Cited on page 43.)

[213] SMEULDERS, A. W. M., WORRING, M., SANTINI, S., GUPTA, A., AND JAIN, R. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 12 (2000), 1349–1380. (Cited on page 10.)

[214] SMINCHISESCU, C., KANAUJIA, A., AND METAXAS, D. Conditional models for contextual human motion recognition. *Computer Vision and Image Understanding 104*, 2-3 (2006), 210–220. (Cited on pages 20, 21, 34 and 36.)

[215] SUCAR, L. E., LUIS, R., LEDER, R., HERNÁNDEZ, J., AND SÁNCHEZ, I. Gesture therapy: A vision-based system for upper extremity stroke rehabilitation. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE* (2010), IEEE, pp. 3690–3693. (Cited on pages 161 and 165.)

[216] SUN, M., KOHLI, P., AND SHOTTON, J. Conditional regression forests for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (2012), IEEE, pp. 3394–3401. (Cited on pages 52 and 167.)

[217] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)* (2015). (Cited on page 174.)

[218] TAN, C. W., CHIN, S. W., AND LIM, W. X. Game-based human computer interaction using gesture recognition for rehabilitation. In *Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on* (2013), IEEE, pp. 344–349. (Cited on pages 161 and 165.)

[219] THERIAULT, C., THOME, N., AND CORD, M. Dynamic scene classification: Learning motion descriptors with slow features analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2013). (Cited on page 29.)

[220] THI, T. H., ZHANG, J., CHENG, L., WANG, L., AND SATOH, S. Human action recognition and localization in video using structured learning of local space-time features. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on* (2010), IEEE, pp. 204–211. (Cited on page 27.)

[221] TOMPSON, J., STEIN, M., LECUN, Y., AND PERLIN, K. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG) 33*, 5 (2014), 169. (Cited on page 164.)

[222] TOWN, C. Ontological inference for image and video analysis. *Machine Vision and Applications 17*, 2 (2006), 94. (Cited on pages 20 and 44.)

[223] TRAN, S., AND DAVIS, L. Event modeling and recognition using markov logic networks. *Computer Vision–ECCV 2008* (2008), 610–623. (Cited on pages 43 and 44.)

[224] UIJLINGS, J., DUTA, I., SANGINETO, E., AND SEBE, N. Video classification with densely extracted hog/hof/mbh features: an evaluation of the accuracy/computational efficiency trade-off. *International Journal of Multimedia Information Retrieval 4*, 1 (2015), 33–44. (Cited on page 163.)

[225] VAMSIKRISHNA, K., DOGRA, D. P., AND DESARKAR, M. S. Computer-vision-assisted palm rehabilitation with supervised learning. *IEEE Transactions on Biomedical Engineering 63*, 5 (2016), 991–1001. (Cited on pages 161 and 163.)

[226] VAPNIK, V. N., AND KOTZ, S. *Estimation of dependences based on empirical data*, vol. 40. Springer-Verlag New York, 1982. (Cited on page 58.)

[227] VAQUETTE, G., ORCESI, A., LUCAT, L., AND ACHARD, C. The daily home life activity dataset: A high semantic activity dataset for online recognition. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on* (2017), IEEE, pp. 497–504. (Cited on page 200.)

[228] VEERARAGHAVAN, A., CHELLAPPA, R., AND ROY-CHOWDHURY, A. K. The function space of an activity. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 1, IEEE, pp. 959–968. (Cited on pages 20, 21 and 31.)

[229] VEERARAGHAVAN, H., PAPANIKOLOPOULOS, N., AND SCHRATER, P. Learning dynamic event descriptions in image sequences. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–6. (Cited on page 80.)

[230] VEMULAPALLI, R., ARRATE, F., AND CHELLAPPA, R. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 588–595. (Cited on page 164.)

[231] VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory 13*, 2 (1967), 260–269. (Cited on pages 20, 21 and 34.)

[232] VU, V.-T., BREMOND, F., AND THONNAT, M. Automatic video interpretation: A novel algorithm for temporal scenario recognition. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (San Francisco, CA, USA, 2003), IJCAI'03, Morgan Kaufmann Publishers Inc., pp. 1295–1300. (Cited on page 80.)

[233] WANG, H., KLÄSER, A., SCHMID, C., AND LIU, C.-L. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition* (Colorado Springs, United States, June 2011), pp. 3169–3176. (Cited on pages 19, 21, 27, 28, 48, 50, 51, 58, 92, 127, 128, 129, 130, 131, 146, 147, 148 and 149.)

[234] WANG, H., KLÄSER, A., SCHMID, C., AND LIU, C.-L. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision 103*, 1 (2013), 60–79. (Cited on pages 163 and 164.)

[235] WANG, H., ONEATA, D., VERBEEK, J., AND SCHMID, C. A robust and efficient video representation for action recognition. *International Journal of Computer Vision 119*, 3 (2016), 219–238. (Cited on page 163.)

[236] WANG, H., AND SCHMID, C. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 3551–3558. (Cited on pages 163, 166 and 171.)

[237] WANG, H., ULLAH, M. M., KLASER, A., LAPTEV, I., AND SCHMID, C. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference* (2009), BMVA Press, pp. 124–1. (Cited on pages 52 and 164.)

[238] WANG, J., LIU, Z., CHOROWSKI, J., CHEN, Z., AND WU, Y. Robust 3d action recognition with random occupancy patterns. In *Computer vision–ECCV 2012*. Springer, 2012, pp. 872–885. (Cited on page 25.)

[239] WANG, L., QIAO, Y., AND TANG, X. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 4305–4314. (Cited on pages 28, 48 and 53.)

[240] WANG, L., AND SUTER, D. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8. (Cited on pages 20, 21, 34 and 36.)

[241] WANG, L., WANG, Y., AND GAO, W. Mining layered grammar rules for action recognition. *International journal of computer vision 93*, 2 (2011), 162–182. (Cited on pages 20 and 42.)

[242] WANG, L., XIONG, Y., WANG, Z., QIAO, Y., LIN, D., TANG, X., AND VAN GOOL, L. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision* (2016), Springer, pp. 20–36. (Cited on page 29.)

[243] WANG, Y., AND MORI, G. Human action recognition by semilatent topic models. *IEEE transactions on pattern analysis and machine intelligence 31*, 10 (2009), 1762–1774. (Cited on pages 23 and 28.)

[244] WEI, S.-E., RAMAKRISHNA, V., KANADE, T., AND SHEIKH, Y. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 4724–4732. (Cited on pages 19, 21, 25 and 76.)

[245] WERBOS, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE 78*, 10 (1990), 1550–1560. (Cited on page 172.)

[246] WILLEMS, G., BECKER, J. H., TUYTELAARS, T., AND VAN GOOL, L. J. Exemplar-based action recognition in video. In *BMVC* (2009), vol. 2, Citeseer, p. 3. (Cited on page 165.)

[247] WILLEMS, G., TUYTELAARS, T., AND VAN GOOL, L. An efficient dense and scale-invariant spatio-temporal interest point detector. In *European conference on computer vision* (2008), Springer, pp. 650–663. (Cited on pages 27 and 163.)

[248] WU, C., ZHANG, J., SAVARESE, S., AND SAXENA, A. Watch-n-patch: Unsupervised understanding of actions and relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 4362–4370. (Cited on pages 28, 86 and 164.)

[249] WU, D., PIGOU, L., KINDERMANS, P.-J., LE, N. D.-H., SHAO, L., DAMBRE, J., AND ODOBEZ, J.-M. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE transactions on pattern analysis and machine intelligence 38*, 8 (2016), 1583–1597. (Cited on page 164.)

[250] WU, D., AND SHAO, L. Silhouette analysis-based action recognition via exploiting human poses. *IEEE Transactions on Circuits and Systems for Video Technology 23*, 2 (2013), 236–243. (Cited on page 163.)

[251] WU, D., AND SHAO, L. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 724–731. (Cited on page 164.)

[252] WU, Z., WANG, X., JIANG, Y.-G., YE, H., AND XUE, X. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM international conference on Multimedia* (2015), ACM, pp. 461–470. (Cited on page 29.)

[253] XIA, L., CHEN, C.-C., AND AGGARWAL, J. K. View invariant human action recognition using histograms of 3d joints. In *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on* (2012), IEEE, pp. 20–27. (Cited on page 26.)

[254] YACOOB, Y., AND BLACK, M. J. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding 73*, 2 (1999), 232–247. (Cited on pages 20, 21 and 31.)

[255] YAMATO, J., OHYA, J., AND ISHII, K. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on* (1992), IEEE, pp. 379–385. (Cited on pages 20 and 34.)

[256] YAMAURA, H., MATSUSHITA, K., KATO, R., AND YOKOI, H. Development of hand rehabilitation system for paralysis patient–universal design using wire-driven mechanism–. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE* (2009), IEEE, pp. 7122–7125. (Cited on page 163.)

[257] YANG, X., AND TIAN, Y. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation 25*, 1 (2014), 2–11. (Cited on pages 52, 166 and 167.)

[258] YANG, X., AND TIAN, Y. L. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on* (2012), IEEE, pp. 14–19. (Cited on page 25.)

[259] YILMA, A., AND SHAH, M. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 1, IEEE, pp. 150–157. (Cited on pages 19 and 25.)

[260] YIN, J., AND MENG, Y. Human activity recognition in video using a hierarchical probabilistic latent model. In *Computer Vision and Pattern Recognition Work-*

shops (CVPRW), 2010 IEEE Computer Society Conference on (2010), IEEE, pp. 15–20. (Cited on page 40.)

[261] YU, E., AND AGGARWAL, J. K. Detection of fence climbing from monocular video. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (2006), vol. 1, IEEE, pp. 375–378. (Cited on pages 20 and 39.)

[262] YU, E., AND AGGARWAL, J. K. Human action recognition with extremities as semantic posture representation. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on* (2009), IEEE, pp. 1–8. (Cited on page 34.)

[263] YUAN, J., LIU, Z., AND WU, Y. Discriminative subvolume search for efficient action detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 2442–2449. (Cited on page 165.)

[264] ZANFIR, M., LEORDEANU, M., AND SMINCHISESCU, C. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 2752–2759. (Cited on pages 53 and 168.)

[265] ZARIFFA, J., AND STEEVES, J. D. Computer vision-based classification of hand grip variations in neurorehabilitation. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on* (2011), IEEE, pp. 1–4. (Cited on page 159.)

[266] ZENG, Z., AND JI, Q. Knowledge based activity recognition with dynamic bayesian network. In *European Conference on Computer Vision* (2010), Springer, pp. 532–546. (Cited on page 40.)

[267] ZHANG, D., GATICA-PEREZ, D., BENGIO, S., MCCOWAN, I., AND LATHOUD, G. Modeling individual and group actions in meetings: a two-layer hmm framework. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on* (2004), IEEE, pp. 117–117. (Cited on pages 20 and 39.)

[268] ZHANG, S., LIU, X., AND XIAO, J. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2017), IEEE, pp. 148–157. (Cited on pages 20, 21 and 37.)

[269] ZHU, W., HU, J., SUN, G., CAO, X., AND QIAO, Y. A key volume mining deep framework for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* (2016), IEEE, pp. 1991–1999. (Cited on page 24.)