Action Detection for Untrimmed Videos based on Deep Neural Networks

Rui Dai

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Détection d'Action pour les Vidéos par les Réseaux de Neurones Profonds

Rui Dai

Inria, Équipe STARS

Présentée en vue de l'obtention du grade de docteur en Informatique **d'**Université Côte d'Azur
Dirigée par : François BRÉMOND
Soutenue le 13 Septembre 2022

Devant le jury, composé de :
Président du jury :
Ivan LAPTEV, Directeur de Recherche, Inria Paris, École Normale Supérieure, France

Rapporteurs :
Karteek ALAHARI, Chargé de Recherche, Inria Grenoble-Rhône-Alpes, France

Dima DAMEN, Professeure, University of Bristol, UK

Examinateur :
Ming-Hsuan YANG, Professeur, University of California Merced, Google, USA

Une thèse préparée à Inria

Inria

Détection d'Action pour les Vidéos par les Réseaux de Neurones Profonds

Action Detection for Untrimmed Videos based on Deep Neural Networks

Jury :

Président du jury :

Ivan LAPTEV	- Directeur de Recherche, Inria Paris, École Normale Supérieure, France

Rapporteurs :

Karteek ALAHARI	- HDR, Inria Grenoble-Rhône-Alpes, France
Dima DAMEN	- Professeure, University of Bristol, UK

Examinateurs :

Ming-Hsuan YANG	- Professeur, University of California Merced, Google, USA
François BRÉMOND	- Directeur de Recherche, Inria Sophia Antipolis, France

DÉTECTION D'ACTION POUR LES VIDÉOS PAR LES RÉSEAUX DE NEURONES PROFONDS

Rui Dai
Directeur de thèse: François Brémond
STARS, Inria Sophia Antipolis, France

RÉSUMÉ

La compréhension du comportement humain et de ses activités facilite l'avancement de nombreuses applications dans le monde réel et est essentielle pour l'analyse vidéo. Malgré les progrès des algorithmes de reconnaissance d'actions dans les vidéos découpées, la majorité des vidéos du monde réel sont longues et non découpées avec des régions d'intérêt denses. Un système efficace de compréhension d'actions dans le monde réel devrait être capable de détecter des actions multiples dans de longues vidéos non découpées. Dans cette thèse, nous nous concentrons principalement sur la détection d'actions temporelles dans les vidéos non découpées, qui vise à trouver les occurrences d'actions dans le temps dans la vidéo. Plus précisément, les méthodes de détection d'actions temporelles font face à trois défis principaux : (a) modéliser dans une vidéo les dépendances temporelles entre les actions, y compris les actions composites et co-occurrentes, (b) apprendre la représentation d'actions à grain fin ainsi que (c) apprendre une représentation à partir de modalités multiples.

Dans cette thèse, nous présentons tout d'abord un important benchmark de détection d'actions en intérieur : Toyota Smarthome Untrimmed, qui fournit des activités spontanées avec des annotations riches et denses pour aborder la détection d'activités complexes dans des scénarios du monde réel. Ensuite, nous proposons plusieurs nouvelles approches pour la détection d'actions dans les vidéos non découpées. Ces approches visent à relever les trois défis susmentionnés : Premièrement, nous étudions la modélisation temporelle pour la détection d'actions. Plus précisément, nous étudions comment améliorer la représentation temporelle en utilisant des mécanismes d'auto-attention. Les méthodes que nous proposons permettent de traiter des vidéos à long terme et de raisonner sur les dépendances temporelles entre les images vidéo à plusieurs échelles de temps. Deuxièmement, nous explorons comment reconnaître et détecter des actions à grain fin en utilisant la sémantique de l'objet et de l'action contenus dans la vidéo. Dans ce travail, nous proposons un cadre général de raisonnement sémantique. Ce cadre se compose principalement de deux étapes : (1) l'extraction de la sémantique de la vidéo pour former une représentation vidéo structurelle ; (2) l'amélioration de la représentation vidéo par le raisonnement sur la sémantique extraite. La stratégie de raisonnement sémantique proposée améliore la détection d'actions à grain fin et montre son efficacité dans les tâches de reconnaissance et de détection d'actions. Troisièmement, nous nous attaquons au problème de la représentation d'une vidéo non découpée en utilisant plusieurs modalités pour la détection d'actions. Nous proposons deux lignes de base multimodales basées soit sur le mécanisme d'attention, soit sur la distillation des connaissances. Les deux méthodes tirent parti des modalités supplémentaires pour améliorer la représentation de la vidéo RVB, ce qui se traduit par de meilleures performances en matière de détection d'action.

Nos méthodes ont été évaluées de manière approfondie sur des repères de détection d'action difficiles. Les méthodes proposées sont plus performantes que les méthodes précédentes, ce qui fait progresser de manière significative la détection d'actions temporelles dans les déploiements du monde réel.

Mots clés: Reconnaisance d'action, Traitement de video, Vision par ordinateur

ACTION DETECTION FOR UNTRIMMED VIDEOS BASED ON DEEP NEURAL NETWORKS

by
Rui Dai
Supervisor: François Brémond
STARS, Inria Sophia Antipolis, France

ABSTRACT

Understanding human behaviour and its activities facilitate the advancement of numerous real-world applications and is critical for video analysis. Despite the progress of action recognition algorithms in trimmed videos, the majority of real-world videos are lengthy and untrimmed with dense regions of interest. An effective real-world action understanding system should be able to detect multiple actions in long untrimmed videos. In this thesis, we focus mainly on temporal action detection in untrimmed videos, which aims at finding the action occurrences along time in the video. Specifically, temporal action detection methods face three main challenges: (a) modelling in a video the temporal dependencies between actions, including composite and co-occurring actions, (b) learning the representation of fine-grained actions as well as (c) learning a representation from multiple modalities.

In this thesis, we first introduce a large indoor action detection benchmark: Toyota Smarthome Untrimmed, which provides spontaneous activities with rich and dense annotations to address the detection of complex activities in real-world scenarios. After that, we propose multiple novel approaches towards action detection in untrimmed videos. These approaches are targeting the aforementioned three challenges: Firstly, we study temporal modelling for action detection. Specifically, we study how to enhance temporal representation using self-attention mechanisms. Our proposed methods allow for processing long-term video and for reasoning about temporal dependencies between video frames at multiple time scales. Secondly, we explore how to recognize and detect fine-grained actions using semantics of object and action contained in the video. In this work, we propose a general semantic reasoning framework. This framework consists of mainly two steps: (1) extracting the semantics from the video to form a structural video representation; (2) enhancing the video representation by reasoning about the extracted semantics. The proposed semantic reasoning strategy improves the detection of fine-grained actions and shows its effectiveness in action recognition and detection tasks. Thirdly, we tackle the problem on how to represent untrimmed video using multiple modalities for action detection. We propose two cross-modality baselines based either on attention mechanism or on knowledge distillation. Both methods leverage the additional modalities to enhance RGB video representation resulting in better action detection performance.

Our methods have been extensively evaluated on challenging action detection benchmarks. The proposed methods outperform previous methods, significantly pushing temporal action detection to real-world deployments.

Keywords: video understanding, action detection, temporal modelling, semantic reasoning, multiple modalities.

ACKNOWLEDGMENTS

I have received countless help and encourage during my Ph.D. study at Inria.

- First of all, I would like to thank my supervisor Francois Bremond. Thank you for giving me the chance for pursuing my PhD. I also appreciate your patience in the numerous discussions. Your guidance really helps me in making critical decisions. In the last three years, I learn many things from you. The most important one is to be calm and positive in any situation.
- Secondly, I would like to acknowledge my PhD jury members. Thanks to my thesis reviewers, Dima Damen and Karteek Alahari, who kindly agreed to review my PhD manuscript. Also, thanks to Ming-Hsuan Yang and Ivan Laptev for serving as members of my thesis committee.
- Thirdly, I would like to thank Toyota Motors Europe, especially Gianpiero Francesca for the support and advice. Thank you to give me the opportunity to join the Toyota Smarthome project. This project gives me a very good starter for my PhD study and I indeed obtain numerous inspirations from this project. I am also grateful to Luca and Lorenzo for their suggestions and help.
- I would like to give special thanks to Srijan Das. Your initial suggestions led me to be a curious man and take up academic research seriously. I really miss our sessions of brainstorming and the sleepless nights we were working together before deadlines.
- I like to thank Université Côte d'Azur for funding my PhD thesis and for providing resources and technical support for my research. I would also like to thank all my colleagues from the STARS team at Inria. I have received a lot of help during my onboarding days. I also want to thank my friends who support me during my Ph.D., especially Hao, Zihao, Di and Yaohui. Without you, my life at Sophia would not be that colourful and interesting. Thank you all for the discussions and encouragement in the last four years.
- Last but not the least, I would like to thank my family, especially my mom who advised me to pursue a doctoral degree after my master's studies. My parents have always supported my life decisions. Without their support, I would not be able to study in France and complete my PhD research. I am also appreciate to my girlfriend Feiyang. Thank you for accompanying me during my thesis.

Many thanks!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer vision is a field of artificial intelligence that focuses on mimicking parts of the human visual system and enabling computers to derive information from images, videos, and other inputs. Nowadays, smartphone and various cameras continually produce tremendous video and media content from individuals every day. Therefore, video understanding and analysis has become one of the essential research subjects in computer vision. Video analysis can be defined as a combination of understanding the scene, objects, actions, events, attributes, and concepts [8] from a series of frames (i.e., a video). Although deep learning techniques have accomplished remarkable performance in many computer vision tasks (e.g., image classification, object detection), video understanding is still far from ideal. Among the topics in video understanding, analysis of action in the video is one of the most critical and challenging tasks. In fact, human beings play an prominent role in the video. Statistics show that 35%, 34%, and 40% of pixels in movies, TV and YouTube videos are related to humans [9]. Hence, studying the human actions and behaviour in a video can help to understand its contents. As an important element of video analysis, action understanding facilitates the progress of numerous real-world applications, such as smarthome, sport analysis system, or human-robot interaction.

In the action understanding domain, **action recognition** is the fundamental task, it aims at classifying the action categories of trimmed video. In this thesis, we define trimmed videos as "pre-segmented" video clips, where each video contains only a single action instance. In other words, the context of the action, i.e., moments before or after the action are not included in the trimmed video. Therefore, action recognition only needs to classify the action categories without the need to detect starting and ending timestamps. However, the majority of videos in the wild (i.e., recorded in unconstrained environments), are naturally untrimmed. Untrimmed videos are long unsegmented videos which may contain several action instances along with the moments before or after each action (i.e., temporal background). The action instances in one video can belong to sev-

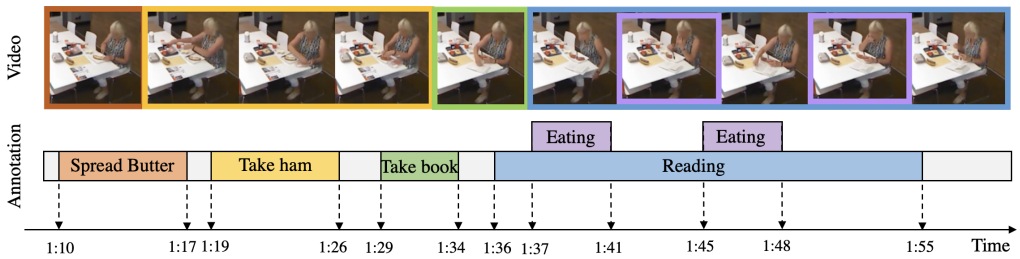


Figure 1.1: Temporal action detection aims to localize action instances in time and to recognize their categories. Here is an example of an untrimmed video that includes multiple action instances of interest with various lengths and categories. Moreover, action instances can overlap.

eral action classes. Besides, action instances may occur at any time of the video and may have various duration. Moreover, overlapping can exist between action instances (i.e., co-occurring actions). An example of the real-world untrimmed video is shown in figure 1.1. The task of detecting actions in untrimmed videos is called **temporal action detection**.

Temporal action detection can be defined as the ability to localize the action instances in time and to recognize their categories. This task has received a lot of attention recently, as it can provide information on: what are the actions and when do the actions happen? The moments right before or right after an action may be very similar in appearance to the start or end of the action, which makes the localization of action intervals very challenging. Previous studies on temporal action detection mainly focused on actions of high-level semantics and videos with a sparse set of actions [10, 11, 12]. However, the action may occur "densely" in real-world scenarios. Besides, low-level "fine-grained" actions can also be important for many applications. For instance, collaborative robots need to recognize how a human partner completes the job in sub-steps to cope with the variations in the task, and sport analysis systems must comprehend fine-grained game actions to report commentaries of live activities, etc. In this thesis, we focus on the temporal action detection that targets **fine-grained actions** and **videos with dense occurrence of actions**. While the temporal action detection task has been studied in both full and limited supervision settings, for the action detection in video with dense action occurrence, the methods still highly rely on full supervision. This is because of the complex temporal relations among action instances, dense action regions, and numerous action categories in the videos. Hence, in this thesis, we only studied fully supervised action detection methods. The goal is to predict action labels at every frame of the video [2, 13, 7].

Temporal action detection has drawn much attention in recent years and has broad applications in video analysis tasks. With the cameras, an automatic detection system may help the deployment of an indoor vision intelligence system system, such as *human less store* and *smarthome*. Take smarthome as an example: a temporal action detection

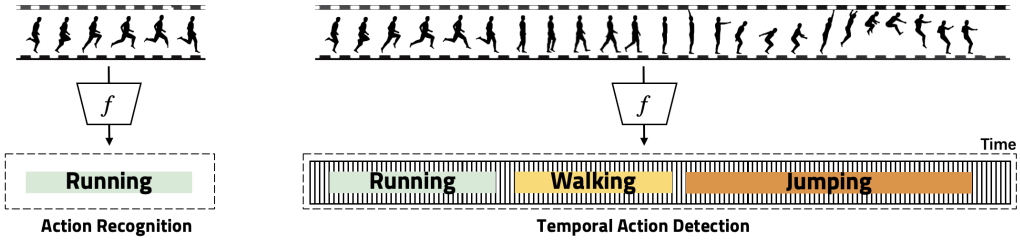


Figure 1.2: The difference between action recognition and temporal action detection tasks. f indicates the network of each task. For action recognition task, its network maps a video clip into an action category label. For temporal action detection task, its network maps a series of frames of untrimmed video into a series of frame-level predictions.

module can detect the behavior of the human subject in real-time and send this information to the support robot (e.g., Partner Robot-HSR [14]), to better interact with the user in the smarthome. Also, suspicious events (e.g., *falling down*) can be detected automatically by the action detection module and reported to the caregivers. Another application is instructional videos. With the growing popularity of social media, many people follow tutorials online to learn cooking or assembling furniture. The instructional videos are usually untrimmed and they include several steps for a main task. Temporal action detection may help detect the main action steps to facilitate the learning process.

The rest of the introduction is organized as follow: firstly, we define the problem statement of action recognition and detection in Sec. 1.1. Secondly, we introduce the applications for action understanding in Sec. 1.2. Then, we describe the scientific challenges in the task of action recognition and detection in Sec. 1.3. Finally, we summarize our contribution in Sec. 1.4. The structure of the thesis is outlined in Sec. 1.5.

1.1 Problem Statement

This thesis involves mainly two tasks: Action Recognition and Temporal Action Detection. Action recognition aims at the classification of clipped action instances. Compared to the vanilla action recognition task, temporal action detection is more challenging. This is because locating the instances in the video is also required in action detection task. An overall difference between action recognition and temporal action detection is shown in Fig. 1.2[1]. Below, we provide the problem statements and their definition for these two tasks.

[1]Action figures are taken from: https://www.pinterest.fr/pin/1688918602463243/

1.1.1 Action Recognition

Action recognition, also known as action classification, is a specific task in video classification, which aims at recognizing the actions in trimmed video sequences. Normally, the video clips are short (e.g around 10 sec./clip) and a video clip contains only a single action without context. The action category (i.e., label) is composed of verb, noun and adverb.

Task Definition: Given a set of videos $\mathbb{V}$ and a set of the corresponding action categories $\mathbb{C}$. Each video $V \in \mathbb{V}$ contains one label $c_V \in \mathbb{C}$. Hence the objective of action recognition is to predict the label c_V based on a video representation of video V. This statement could be extended for multiple action instances in a video clip, where $c_V \in \mathbb{C}$ is a set of action categories (i.e., multi-label video classification).

1.1.2 Temporal Action Detection

Temporal action detection is an extension of the action recognition problem. Besides recognizing the action categories, temporal action detection task consists in localizing the action instance in the untrimmed video as well. The untrimmed video can be complex: a video may contain one or more action instances and the action instances can overlap (i.e., concurrent actions). Normally, (1) the task designed for videos with a sparse set of actions [11, 12] is referred to as "temporal action localization" [15]. (2) the task that focuses on fine-grained actions can be referred to as temporal action detection [16, 13, 17, 18] or segmentation [5, 19]. In this thesis, we focus on the fine-grained action detection for videos with dense occurrence of actions. As a result, we follow the naming convention in [13], using the term "temporal action detection".

Task Definition: Formally, for a video sequence of length T, each time-step t is associated with a ground-truth action label $y_{t,c} \in \{0, 1\}$, where $c \in \{1, ..., C\}$ indicates an action class. For every time-step, an action detection network predicts class probabilities $\tilde{y}_{t,c} \in [0, 1]$.

Similar Tasks: there are many similar tasks to temporal action detection. We present below their definitions along with the differences compared to temporal action detection.

- **Temporal Action Proposal Generation**: This task evaluates the ability of algorithms to generate high-quality action proposals. The goal is to produce a set of candidate temporal segments that are likely to contain a human action. Unlike temporal action detection, temporal action proposal generation is a class-agnostic task. This task usually acts as a sub-task of event-level action detection.

- **Untrimmed Video Classification**: This task aims at recognizing all the actions that appear in an untrimmed video. Unlike temporal action detection, untrimmed ac-

tion classification does not need to predict the temporal boundaries of the action instances.

- **Spatio-temporal Action Localization**: This task is intended to evaluate the ability of algorithms to localize human actions in both space and time. Unlike temporal action detection, this task also needs to detect the spatial location of the subject who performs the action.
- **Online Action Detection**: Unlike previous tasks, the goal of online action detection is to detect an action as it happens and ideally even before the action is fully completed. Being able to detect an action at the time of the occurrence can be useful in many real-world practical applications.

We claim that for an effective and efficient real-world action understanding system, the system should be able to detect multiple actions in long untrimmed videos. In this thesis, we focus mainly on temporal action detection in untrimmed videos, which aims at finding the action occurrences along time in the videos. Nevertheless, we also evaluate some proposed methods on trimmed action recognition task for validating the model generalization. For convenience and with a slight abuse of terms, hereafter in this thesis, we often refer to action detection as the problem of temporal action detection. Also, we may utilise "activity" to indicate "action" in this thesis.

1.2 Applications

Temporal action detection bears a significant potential for numerous real-world applications. In the following, we introduce four representative applications.

Smarthome: With the ageing population issue, a Smarthome system could relieve the dramatic need of caregiver workforce. With such a smart indoor camera-based system, the elderlies can live better alone at home. Such a system can work with a partner robot. The real-time information is sent to the robot to better interact with the older adult and helps them. Moreover, such a system can help detect and report potentially dangerous situations to caregivers if necessary. It can also analyse daily living actions to improve life quality. For example, how much water is *drunk* a day, or how much time is spent *reading* or *using electronic devices*. In such a system, the temporal action detection module can be seen as the core component for detecting actions from those streaming videos.

Video Summarization: The goal of video summarization is to produce a compact visual summary that encapsulates the key components of a video (i.e. highlight moments). Its main value is to turn hours of video into a short summary that can be interpreted by a

human viewer in seconds. Creating visual diaries and video highlights are popular usages of video summarization. As humans play an important role in human recorded videos, action detection can help to retrieve the key points from the untrimmed videos. The video summarization has already been used in real-world applications, such as video highlights in smartphone albums.

Skill Assessment: Many domains now require to analyze the quality of a person's activities and to assess whether the action is being performed correctly. Such skill assessment is, for example, relevant for progress assessment in physical rehabilitation, or for coaching in some sports (e.g., basketball, tennis). Skill Assessment allows athletes to take a critical look at their performance in order to improve their skills and prevent injuries. As an action can composed of several atomic actions, learning the occurrences of such atomic action relies on temporal action detection techniques.

Human-Robot Interaction: Ability to perceive human actions plays a key role in many human-robot interaction scenarios. With the development of action recognition and detection, a robot is able to recognize actions or gestures done by a user and to trigger appropriate feedback. Such solutions have been already introduced to some human assistant robots or autonomous vehicles. For example, smart car systems rely on action detection systems to understand human's gestures, such as *change the music* or *answer a call*.

1.3 Scientific Challenges

Over the past years, deep learning has led to huge success in image and video analysis. Among video analysis tasks, temporal action detection is a novel but important research problem. Many unanswered questions keep this problem challenging.

1.3.1 Video Representation Learning

The input of action detection model is an untrimmed video, which is a sequence of frames of a scene at a given frame rate. Processing long-term videos is challenging as the input data can be very large. For example, a 5 minutes video with 24 fps and VGA resolution (i.e., 640 $\times$ 480) contains more than 2.2 billion pixels. Detecting actions in a video relies on the capacity of the model that extracts the action related information from the video. Therefore, how to effectively and efficiently model the representation of the video is challenging and crucial in temporal action detection. To tackle this challenge, previous methods utilized a two-stage framework [20, 17, 21]: Firstly, extracting the features of the local video snippets using a visual encoder (e.g., 3D Convolutional Network [22, 23] or

Transformer [24, 25]). The visual encoder is used to model the spatio-temporal relations in the short-term video snippet. Secondly, after stacking the extracted features along time, temporal models (e.g., LSTM [26] or TCN [5]) are used to explore the long-term temporal dependencies. This framework effectively reduces the computation cost. In this thesis, similarly to previous work, we follow a two-stage framework. Our focus for video representation learning lies in the second stage: how to effectively model long-term temporal information?

Recently, some researchers have found that the temporal model performance is limited by the visual encoder. Because of the dissociation with the visual encoder, the temporal model can not take full advantage of the spatial information of the video to model temporal dependencies. To tackle this issue, some researchers firstly improve visual encoder by leveraging the attention mechanism [27, 28] to filter more salient spatial information from the video. Secondly, few recent methods [29, 30] utilize the momentum updated memory bank to connect the visual encoder and the temporal model, so that they can train jointly the two stages in a latent manner. As the training process is costly and not all the data is necessary used to train the model, we keep the framework in a two-stage fashion in this thesis. Studying long-term spatial-temporal modelling will be one of our future works.

1.3.2 Supervision-level

Supervision-level indicates how much ground truth labels we need for learning the discriminative representation for a specific task. The challenge consists in using as few annotations as possible. The annotation process is complex and costly, especially for large datasets. Therefore, the model that relies on less annotations is essential but challenging. There are some propositions in the community for designing algorithms with less annotations. For example, weakly-supervised action detection learns the model using only video-level labels to detect actions from untrimmed videos. However, current methods for this task highly rely on a filter that can distinguish the foreground and background actions [31, 32]. Therefore, those models are evaluated on simple videos with sparse action regions. For the videos with a dense occurrence of actions, current methods still highly rely on full supervised annotation. The situation is even more severe for a model that has lesser supervision (e.g., unsupervised action detection).

As we focus on analysing videos with dense action occurrences, we provide full ground truth to our model in this thesis. However, with supervised setting, it is still challenging to get high performance. How to learn the representation of untrimmed video with less supervision is our future work.

1.3.3 Cross-dataset Generalization

The ultimate goal for temporal action detection is to detect action instances from arbitrary real-world videos. This objective requires the model to be general enough to cope with various environments and scenarios. To tackle this point, a new task is proposed in the community. This task requires the model to be trained dataset-by-dataset and prevent the model performance from decreasing on the previous dataset. In the training phase, the new dataset may introduce new action classes, which makes the task more challenging [33]. In other words, if a dataset represents the source domain and another dataset represents the target domain, we want the knowledge learned by the model from the source domain can be generalized to the target domain [34, 35]. Such an action detection system requires the model to learn a robust representation of different scenarios and environments. Limited by the annotation issue, current methods [34, 33, 35] have been tried only in trimmed video clip for video classification. As the action detection task input is the untrimmed videos for which action instances are not pre-segmented, generalising the representation for action detection across different domains is more challenging. In this thesis, we keep the same domain for the training and testing sets of our method. However, we evaluate these methods on multiple large datasets to show the generalization and robustness of our methods.

1.3.4 Class-imbalance

In the real world, action distribution is in general highly imbalanced with a long-tail distribution [36], with a few categories covering most of the data (so-called head of the distribution), and the rest having only a few samples per category (so-called tail). Training the model with imbalanced data is challenging, as it is difficult to learn the representation of tail action classes while training along with the head action classes. To train on imbalanced data, some methods firstly introduced focal loss [37] to reassign the importance of different samples. Nevertheless, giving more focus on the rare samples led the model to focus more on the outliers and may cause over-fitting. Secondly, some methods [38, 39, 40] used the sampling and data augmentation techniques to re-balance the number of samples for each class. However, limited by the imperfect synthetic samples (i.e., data generation), the few sampled class detection is not getting effectively improved. In this thesis, we pre-trained our visual encoder on large balanced action datasets (i.e., kinetics [41]) and fine-tuned the visual encoder on the training set of the target dataset. This strategy helps the model better learn the few-sampled action representation and alleviates the class imbalance issues to some extent.

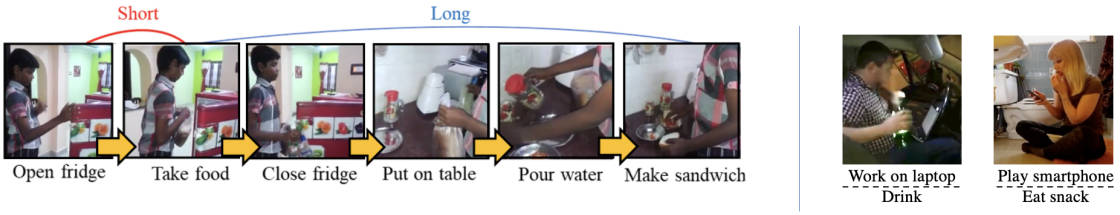


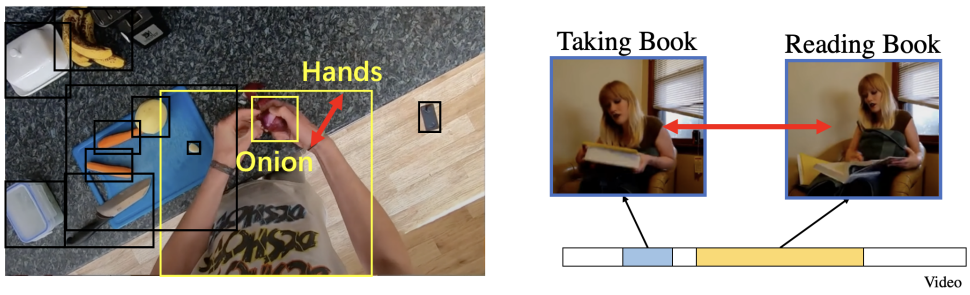
Figure 1.3: **Complex temporal relations in untrimmed videos.** On the left, we show a set of actions performed in a sequential manner in a video. On the right, we present two examples of co-occurring actions. The sampled frames are taken from Charades dataset.

1.3.5 Filming Setting

Videos are captured in the open world with variant filming settings. The subject can be far away from the camera or close to it, resulting in various sizes of the subject in the video. Furthermore, the video may not be recorded by a cameraman but by a fixed camera, thus the subject can be partially not in the centre of the frame and can occur anywhere in the frame even outside. Moreover, the subject can be captured in diverse camera views (not frontal) and can be partially occluded by objects in the environment. How to make the representation robust to different filming settings is crucial for designing a model for the real world. In this thesis, we introduce a large indoor dataset: Toyota Smarthome, which comprises multiple camera views along with a real-world filming setting. We evaluated our models in both settings: We utilize the popular benchmarks which have Lab settings to compare our models with SoTA methods. In addition, we experiment with Toyota Smarthome which features a new setting to evaluate our models in real-world conditions.

1.3.6 Complex Temporal Relations

By contrast to trimmed videos, untrimmed ones contain rich semantics with complex temporal relations. As mentioned earlier, in the standard two-stage temporal action detection framework, the untrimmed video needs to be partitioned into shorter clips for feature extraction. Processing these shorter clips independently can lead to loss of information corresponding to the temporal or semantic dependencies between video segments. Therefore, temporal modelling is critical to capture these dependencies, in order to benefit from the video context to refine the clip representation. Effective reasoning across time can help predict the action category of the clip and predict the action completeness. Modelling temporal dependencies is essential especially for videos with complex temporal relations. For example, a set of actions that occur together often follow a well defined temporal pattern. As shown in figure 1.3, when "*making breakfast*", the sub-actions "*opening fridge*", "*taking*



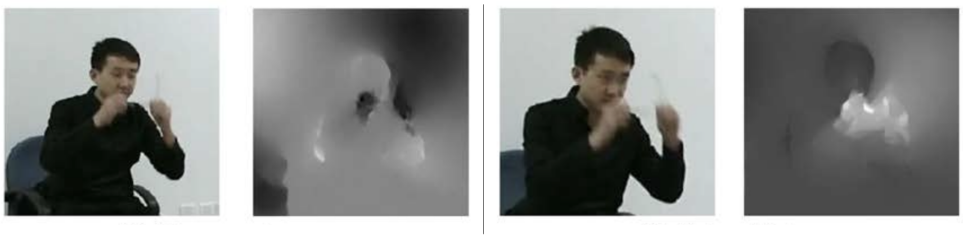
(1) Object (2) Action

Figure 1.4: Semantic reasoning helps action detection. (1) Object semantics: Knowing the existence of the action relevant **objects** (onion, hands) and the "motion" between them can help determine action *peeling onion*. (2) Action semantics: knowing the existence of action *taking a book* can help predicting its relevant action *reading book*. The red arrows represent the relation between the semantics. Sampled frames are taken from EPIC-KITCHENS [1] and Charades [2] datasets.

food" and "*make sandwich*" can be performed in a sequential manner. Besides, both short-term and long-term actions may occur at the same time in the same video. For example, performing the short action *eating snack* while *playing smartphone* (i.e., long action). To detect the actions in such complex videos, it is important to model both short-term and long-term temporal dependencies of the actions. Therefore, temporal modelling is one of the focus in this dissertation.

1.3.7 Fine-grained Actions

Knowing fine-grained details of the action is critical for some context-aware scenarios. For example, while cooking, knowing *cutting* either "*beef*" or "*onion*" can provide clues for learning a better model for cooking instructional videos. However, recognizing and detecting fine-grained actions from videos is challenging, as there are subtle inter-class variations among the fine-grained action categories (e.g., *drink from bottle* or *can*). Hence, detecting such actions needs to capture both the relevant semantic information and the cross-semantic relationships in the video. For example, as shown in Figure. 1.4, modelling the relation between different "object" semantics such as *hand* and *onion* can help detect the fine-grained action instance (e.g., *peeling onion*). Also, one untrimmed video may contain multiple action instances. Detecting an action in the video may rely on the representation of other relevant action instances in this video. For example, knowing "*taking the book*" action can help detect "*reading book*" action in the same video, and vice versa. In this thesis, we propose a semantic reasoning framework for detecting actions in the aforementioned challenging scenarios.



(1) Take off glasses (2) Take on glasses

Figure 1.5: Complementary nature of different modalities. Here we take an example of actions *taking on* and *off glasses* with RGB and optical flow. The sampled frames are taken from PKU-MMD [3] dataset.

1.3.8 Multi-modalities

While recording a scene, data can be captured by different sensors to have different modalities, such as RGB, depth, audio, etc. Those modalities can be complementary for recognizing complex actions. For example, *taking on* and *taking off glasses* are similar in RGB frames while the difference can be salient in the optical flow frames (see figure 1.5). Thanks to the complementary nature between modalities, learning from multiple modalities can improve the action detection performance. While two-stream networks [42] have become a conventional setting in the action detection framework, how to fuse the different modalities remains a challenging and worthy research problem.
The sensors for capturing the modalities are costly. With the development of recent algorithms, additional modalities can be extracted from RGB videos in a "post-processing" step (e.g., optical flow, skeletons). However, extracting those modalities from RGB is quite computationally expensive. The difficulty remains in effectively leveraging multiple modalities of an untrimmed video with reasonable computation cost. In this thesis, we introduce two methods for leveraging additional modalities to enhance the RGB representation, in a light-weighted manner.

1.4 Contributions

Our contributions are motivated by the complex challenges involved in real-world videos. To address these challenges, we made four key contributions. The first contribution includes a real-world indoor dataset containing action videos performed in a spontaneous manner. Our second contribution are three temporal models that leverage attention mechanisms for enhancing the temporal representation. Our third contribution is a semantic reasoning framework that can learn the relationships among video semantics for fine-grained action understanding. Our fourth contribution is two multi-modal strategies to

take the benefits of multiple modalities into account for detecting actions. Below, we briefly describe these contributions.

1.4.1 Smarthome Dataset

In this thesis, we introduced a real-world indoor dataset: Toyota Smarthome Untrimmed (TSU), which contains daily-living activities performed in a natural manner and captured from multiple non-optimal viewpoints. Activities performed in a spontaneous manner lead to many real-world challenges that are often ignored by the vision community. This includes composite action detection, concurrent action detection, low camera framing, action-class imbalance, and occlusions. We provide rich and dense annotations of TSU dataset to address the detection of complex actions in real-world scenarios.

1.4.2 Temporal Relational Reasoning

Temporal modelling is important for processing sequential data, including videos. As mentioned earlier, it is essential to model the temporal dependencies between different time steps (i.e., snippet) in a video. In this dissertation, we proposed three effective temporal modelling networks for action detection:

Self-Attention Temporal Convolutional Network (SA-TCN): SA-TCN is an attention-based model which features an encoder-decoder structure to shrink the temporal resolution. Between the encoder and decoder, a self-attention block is used to capture the non-local dependencies between different time-steps in the video. We argue that such an architecture design can help model long-term temporal dependencies in untrimmed videos.

Pyramid Dilated Attention Network (PDAN): similar to SA-TCN, PDAN also relies on attention mechanism. The basic component `Dilated Attention Layer (DAL)` allocates attentional weights to neighbouring features in the kernel, which enables it to learn better local representation across time. PDAN is built upon DALs, which can model short-term and long-term temporal relations simultaneously by focusing on local segments at the level of low and high temporal receptive fields. This property enables PDAN to handle complex temporal relations between different action snippets in long untrimmed videos.

Multi-Scale Temporal ConvTransformer (MS-TCT): Benefiting from the Transformer and Temporal Convolution architecture, we proposed a ConvTransformer based architecture for the action detection task. This network comprises a Temporal Encoder module which extensively explores global and local temporal relations at multiple temporal resolutions.

Then, a Classification module is used to learn the instance center-relative position and to predict the frame-level classification scores.

1.4.3 Semantic Relational Reasoning for Action Detection

Videos may contain rich semantic information such as objects, actions, and scenes. Relationships among different semantics are high-level knowledge which is critical for understanding the video content. Therefore, semantic relational reasoning can help determine the action instance occurrences and locate the actions in the video. To leverage semantic for action understanding, we proposed two semantic modelling networks: CTRN is designed to capture the "inter-action" relations for action detection and THORN aims at modelling the "intra-action" relations for action recognition.

Class-Temporal Relational Network (CTRN) is a network for the action detection task. This network targets modelling the complex action relations in a video and refining action detection precision based on the learned global action relationships. In other words, CTRN enhances the action detection performance by exploring the inter-action class relationships. In practice, CTRN filters the action class-specific representation from the mixed representations and then models the action class and temporal relations alternatively. With CTRN, we can effectively detect the actions in complex videos with dense action regions. CTRN has achieved competitive state-of-the-art performance in challenging action detection datasets.

Temporal Human-Object Relational Network (THORN) is a network for action recognition, which can be seen as a continuation of CTRN, and which explores the intra-instance semantic relationship. Unlike CTRN which aims to model the action relations across time, THORN focuses on exploring the object semantics in the spatio-temporal space. In practice, we extract the object representation from the spatio-temporal representation and we model the cross-object relations to predict the action. THORN has achieved competitive state-of-the-art performance in egocentric-view action recognition datasets.

1.4.4 Multi-Modal Learning

Recognizing and detecting actions in videos involves understanding different cues. The cues that are computed from different modalities are complementary in their feature space. Thus, fusing them in a common feature space enables a classifier to learn even more discriminative features compared to their classification in an individual feature space. Therefore, to incorporate the effectiveness of each modality, we propose two strategies to learn a multi-modal representation for action detection.

Attention Guided Network (AGNet): For leveraging multiple modalities, we firstly propose to utilize the additional modality to guide the RGB stream based on the attention mechanism. The goal of this method is to leverage the complementary nature of the different modalities (e.g., Optical Flow, 3D Poses) to guide the RGB stream for better action detection. The main contribution is the attention module, which utilizes additional modalities to generate the attention weights at multiple temporal scales and which indicates the region of interest of the action in the video. AGNet is the baseline proposed in Toyota Smarthome Untrimmed for the action detection task.

Knowledge distillation for action detection: The two-stream structure is effective for action detection. However, using such a setting is contingent upon the availability of multiple modalities and of expensive processing resources. To handle this, we propose a knowledge distillation framework that can encourage the RGB stream to learn both local and global video information from additional modalities. With this new framework, the distillation is realized at both atomic and sequence levels. The result is an Augmented-RGB stream that achieves competitive performance as the two-stream network while using only RGB at inference time.

1.5 Thesis Structure

In the following chapters: Firstly, we review methods related to this thesis, especially the state-of-the-art methods in action detection in chapter 2. Then, we introduce our proposed dataset and methods:

- In chapter 3, we introduce a challenging indoor dataset Toyota Smarthome Untrimmed [43] for action detection. We also compare this dataset with other related datasets to highlight the challenges featured by Toyota Smarthome Untrimmed dataset.
- In chapter 4, the methods aiming at modelling temporal relations in the video are introduced. In this chapter, we describe three networks: SA-TCN [44], PDAN [45] and MS-TCT [46], which combine self-attention with temporal convolution to capture both local and global temporal dependencies in untrimmed videos.
- In chapter 5, we present a semantic reasoning framework for action understanding. To study that this framework can model both spatial and temporal semantics, experiments are conducted on both action detection [47] and recognition [48] tasks.
- In chapter 6, we focus on the multi-modal framework. We introduce two methods that leverage the attention mechanism [43] and knowledge distillation [49] for multi-modal action detection.

Finally, we summarize the thesis contributions and we describe several perspectives as future work in chapter 7.

1.5.1 Publication List

We list all publication contributions in the course of this thesis.

[Chapter 3] Toyota Smarthome Dataset:

– **R. Dai**, S. Das, S. Sharma, L. Minciullo, L. Garattoni, F. Bremond, and G. Francesca. *Toyota Smarthome Untrimmed: Real-World Untrimmed Videos for Activity Detection*, TPAMI 2022. [43]

– S. Das, **R. Dai**, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond and G. Francesca. *Toyota Smarthome: Real World Activities of Daily Living*, ICCV 2019. [50, 51]

– D. Yang, **R. Dai**, Y. Wang, R. Mallick, L. Minciullo, G. Francesca and F. Bremond. *Selective Spatio-Temporal Aggregation Based Pose Refinement System: Towards Understanding Human Activities in Real-World Videos*, WACV 2021. [52]

[Chapter 4] Temporal Relational Reasoning:

– **R. Dai**, S. Das, K. Kahatapitiya, M. Ryoo, F. Bremond. *MS-TCT: Multi-Scale Temporal ConvTransformer for Action Detection*, CVPR 2022. [46]

– **R. Dai**, S. Das, L. Minciullo, L. Garattoni, G. Francesca and F. Bremond. *PDAN: Pyramid Dilated Attention Network for Action Detection*, WACV 2021. [45, 53]

– **R. Dai**, L. Minciullo, L. Garattoni, G. Francesca and F. Bremond. *Self-Attention Temporal Convolutional Network for Long-Term Daily Living Activity Detection*, AVSS 2019. [44]

[Chapter 5] Semantic Relational Reasoning:

– **R. Dai**, S. Das, and F. Bremond. *Class Temporal Relational Network for Action Detection*, BMVC 2021. [47]

– G. Mohammed, **R. Dai**, and F. Bremond. *THORN : Temporal Human Object Relation Network for Action Recognition*, ICPR 2022. [48]

[Chapter 6] Multi-Modal Representation Learning:

– **R. Dai**, S. Das, and F. Bremond. *Learning a compact RGB representation with cross-modal knowledge distillation for action detection*, ICCV 2021. [49]

– S. Das, **R. Dai**, D. Yang and F. Bremond. *VPN++: Rethinking Video-Pose embeddings for understanding Activities of Daily Living*, TPAMI 2021. [54]

– S. Das, S. Sharma, **R. Dai**, F. Bremond and M. Thonnat. *VPN: Learning Video-Pose Embedding for Activities of Daily Living*, ECCV 2020. [55]

Chapter 2

Literature Review

In this chapter, we overview how previous methods address action detection task. Firstly, we discuss the framework and relevant methods for the temporal action detection task. We then introduce the principle loss function, evaluation metrics and datasets that we utilized in this task. More fine-grained related work to each contribution will be discussed in the corresponding chapters respectively.

2.1 Action Detection Methods

Action detection involves representing an untrimmed video. To model the video representation, most recent action detection frameworks involve two steps: (1) Extracting frame or snippet level features using a model trained for the action classification task, we call this step visual encoding or video encoding; (2) Modelling the temporal relation across the snippet-level features. After that, the prediction heads detect the action instance with the obtained video representation. Below we revise the relevant methods for each of these two steps.

2.1.1 Visual Encoding

Learning representations for video has been popular over the years [42, 56, 22, 23, 57, 58]. As mentioned earlier, an untrimmed video consists of a huge number of pixels, thus an effective and efficient manner is required to light-weight the computation cost. Modelling a video into a sequence of snippet-level features is the conventional way of processing long untrimmed videos [5, 59, 60, 21, 15, 17, 18, 19]. More specifically, in this step, frame or snippet level features are extracted using a model which is trained on action clips. These features are further input to a model which is trained for the task of action detection. Thus, the efficiency of the detection task highly relies on the

quality of the extracted features or, in other words, on the learned representations of the action classification models. These classification models vary based on the input data modality. For instance, 3D human poses are generally processed by sequential networks or graph convolutional networks, whereas RGB images and optical flow images are generally treated by 3D convolutional networks.

3D human pose is a popular modality which provides the location of the key joints of a subject for every frame [61]. This modality is also dubbed the human skeleton data. Skeleton data attracted considerable attention due to their strong adaptability to dynamic motion and complicated background [62, 63, 64]. Conventional deep learning based methods manually structure the skeletons as a sequence of joint-coordinate vectors [65, 62]. However, representing skeleton data as a vector sequence can not fully express the dependency between correlated joints. Recently, graph convolutional networks (GCNs) have been applied to model the skeleton data [66, 67, 52, 68, 69]. Yan et al. [66] have constructed a spatial graph based on the natural connections of joints in the human body. Inspired by [66], Shi et al. [67] have proposed a two-stream GCN to better model the spatial information within a short period of time. Most recently, Duan et al. [70] proposed to utilise 3D heatmap volume instead of a graph sequence as the base representation of human skeletons. Compared to GCN-based methods, their method is more effective in learning spatio-temporal features and more robust against pose estimation noises. Skeletons can be effective for representing the pose of the person performing an action and capturing human-centred motion. But what about contextual information like environmental details (e.g. sink for *clean dishes with water*), encoding object information (e.g. glasses)? For that, we need RGB frames.

RGB images are the commonest modality which is utilized by many effective methods in order to model the appearance information. Few works [56, 71] learn appearance features from the frame-level classification of actions, using 2D CNNs [72, 73]. 3D CNNs are the natural evolution of their 2D counterparts [23, 22, 57, 74]. Tran et al. [23] have proposed 3D CNNs (C3D) to capture spatio-temporal patterns from a sequence of 8 RGB frames. In the same vein, I3D [22] inflates the kernels of ImageNet pre-trained 2D CNN to jumpstart the training of 3D CNNs. While these methods are effective for the recognition of fine-grained and object-based actions with a short temporal extent, they are too rigid and computationally expensive to handle minute-long videos [28, 75]. In order to effectively learn temporal localization of actions in long videos, the existing action detection methods process the videos on top of the aforementioned 2D or 3D CNNs. As **Optical Flow** and **Depth** videos have the similar data structure as RGB (i.e., Height $\times$ Width $\times$ Time $\times$ Channel), these modalities follow an RGB-like fashion for visual encoding.

Recently, many transformer-based models [25, 24, 76] are proposed in video classification task that outperformed the state-of-the-art 3D CNNs performances. Those video

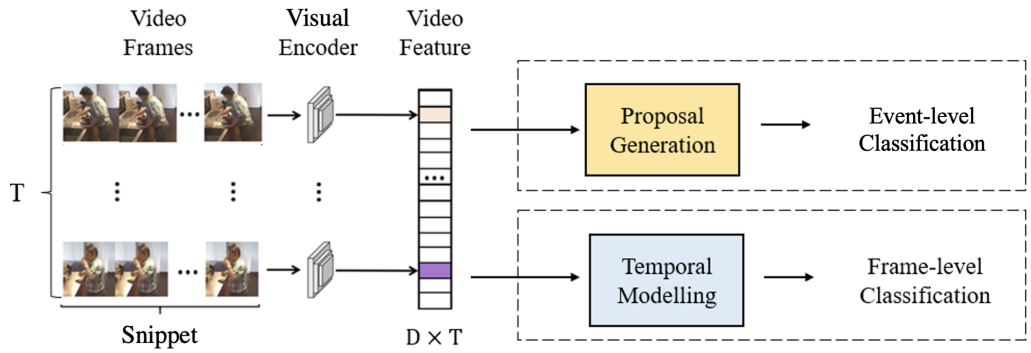


Figure 2.1: A typical action detection framework. There are two types of predictions: (1) event-level and (2) frame-level prediction.

transformers can model the spatio-temporal information from the video clips in an effective manner, and hence can be utilized as the visual encoder in future work.

2.1.2 Temporal Action Detection

The second step is to utilise the encoded visual features for temporally detecting the actions. As shown in Fig. 2.1, there are principally two types of prediction [77]: event-level (i.e., instance-level) and frame-level predictions. The prediction head is relevant to the prediction type. For video with sparse action regions [78, 11, 12], extracting the foreground events from the background is straightforward. As a result, after the temporal modelling, a typical framework is composed of two main components as prediction head: an event proposal generator to generate the potential event regions and a classifier to recognize the action labels for the proposed event. Actionness [59, 79, 80] and anchor [81, 15, 82] based proposal generation are wildly used for generating the action proposals. When processing densely labelled videos with fine-grained actions [83, 84, 2, 13], the prediction head is formulated as learning a mapping function that maps a series of temporal features to a series of predictions. For the feature at every time step (i.e., frame or snippet), the prediction head predicts its action labels. In this dissertation, we focus on the frame-level prediction in videos with dense action occurrences. Below, we present the aforementioned techniques in detail.

2.1.2.1 Proposal-based Methods

Inspired by the proposal-based object detection [85], recent action detection methods with sparse action occurrences are proposal-based methods which possess a proposal generator in their framework. There are principally two types of proposal generators: actionness-based and anchor-based generators.

Actionness-based methods [59, 86, 87, 88] composed of two classifiers. In these methods, one binary classifier provides the actionness score (i.e. foreground action or background) for each frame to generate the action proposals, while another classifier predicts the action class of these proposals. The actionness detection can also be decomposed of the detection of the action-start and action-end [87] or the composition of action-start and action end[89, 60, 90]. Actionness detection is similar to the object location heat-map in anchor-free object detection methods [91]. However, different from the class-specific heat-map in object detection, the actionness detector is class-agnostic. Therefore, an additional classifier is required to classify the proposals. As the standard actionness-based methods rely on the binary classifier to filter the foreground actions from the background, this strategy can not handle the videos with dense foreground action regions.

Anchor-based architecture [81, 15, 82, 92, 21] is inspired by two-stage object detection framework [85], which leverages a set of predefined anchors to generate action proposals along with another classification stage. The proposals are the multi-sized windows centred at the anchor. An NMS post-processing is normally used to filter the high overlapped action proposals. However, anchor-based methods require a large number of anchors for generating the proposals and the computation increases exponentially with the increase of the number of anchors. Moreover, the anchor-based methods with NMS may fail on detecting the co-occurring action pairs. Hence, anchor-based methods normally fail in densely labelled datasets [81, 92].

Recently, with the advances of Transformer architectures, proposal-based methods can handle more complex videos. RTD-Net [93] is built based on transformer decoder DETR [94] for generating action proposals in videos. However, this network relies on boundary attentive representations to detect the action boundaries. Similar to the previous mentioned actionness strategy, such module can not work on a densely annotated dataset which does not have clear foreground and background boundaries. Similarly, Nawhal et al. [77] propose an encoder-decoder transformer: Activity Graph Transformer. This model also follows a structure similar to DETR. Activity Graph Transformer represents video in a graph structure and utilises graph attention to learn the video's global context. After that, with the learned global context, this model transforms a set of queries into contextual embedding. These embeddings are then used to provide predictions of action instances. However, limited by the fixed number of queries (i.e., proposals), the model struggles in densely labelled videos where there is a large variation in the number of action instances in the video.

2.1.2.2 Sequence-to-Sequence Model

To handle the issues in anchor-based techniques for processing densely labelled videos, some methods [5, 18, 95, 96, 19] borrow the Seq2Seq framework from Natural Language Processing (NLP) [97] to apply it to frame-level action detection. This process is also similar to image semantic segmentation as both aim to classify every single instance, i.e., frames in the temporal domain versus pixels in the spatial domain.

After the visual encoding, Seq2Seq methods feature an efficient temporal module to model the temporal information and a classifier to perform the frame-level action detection. This framework "interprets" the image sequence into a sequence of prediction scores. In other words, frame-level action detection can be seen as a class-specific actionness detector. To compare with the proposal-based methods and evaluate with sparsely labelled datasets, by referring to advances in "actionness" detection, the action proposals (i.e., discrete detection instances) can be further generated from the frame-level detection results via a post-processing manner [13, 18, 96]. Below, we briefly introduce some representative sequence-to-sequence models; a more detailed representation is provided in the related work of chapter 4.

Recurrent Neural Networks (RNNs) [13, 98, 99, 96] have been popularly used to model the temporal relations between frames. In this network, connections between nodes form a directed or undirected graph along a temporal sequence. This allows it to exhibit temporal dynamic behaviour. However, RNNs only implicitly capture relationships between certain actions with high motion. Furthermore, due to the vanishing gradient problem, RNN-based models can only capture a limited amount of temporal information and short-term dependencies.

Temporal Convolutional Networks (TCNs) [5, 100, 101, 19, 102] are another group of temporal processing methods, which is a one dimensional convolutional network [103]. Contrary to RNN-based methods, TCNs can process long videos thanks to the fact that kernels share weights for all the time steps. However, the translation invariance and Pooling layers may lead the convolution network to ignore the relationship between the part and the whole [104]. Moreover, the shared kernel allocates the same weights to each local feature in the kernel. This property prevents TCN from extracting the key information efficiently from videos. As a result, current methods can only process datasets with videos characterized by simple temporal relations [84, 83].

Transformers: inspired by the advanced methods in the NLP domain, some researchers [7, 71, 105, 106] leverage the Transformer architecture and the self-attention mechanism [107] for temporal modelling in action detection. Transformer architectures feature general modelling capability, which can model all token-to-token relationships in sequence-to-sequence tasks. Besides, unlike the aforementioned local operations, the

transformer can model the temporal dependencies by attending to the entire sequence information, which is a global operation. Moreover, this architecture is scalable to large models and large data. For the above reason, Transformers are getting more and more popular in the computer vision domain and become dominant in many sequence-to-sequence tasks.

2.2 Loss Function

In this thesis, we focus on fully supervised frame-level action detection. Full supervision is a process to train a network (i.e., algorithm) to map the input data into prediction labels, where each training data has its corresponding ground truth label. In the task of temporal action detection, full supervision employs the labels of the training set that contains the action category labels and the corresponding temporal annotation information, i.e., action occurrences for each category at each time step.

Videos with dense action occurrence contain co-occurring actions, i.e., multiple instances occurring at the same time. Since the video has been embedded into a sequence of frame-level or snippet-level features by the visual encoder, detecting actions from such temporal features can be seen as multi-label classification task on top of these features. Hence, sequence-to-sequence action detection frameworks utilize the binary cross entropy loss ($\mathcal{L}_{BCE}$) [108]:

$$\mathcal{L}_{BCE} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C} y_{tc} log(P_{tc}) \tag{2.1}$$

Where T is the number of the frames or snippets, C is the number of action classes and P is the predicted score. In other words, after temporal modelling, we perform a binary classification for each frame or snippet feature and for every action class. This loss term is the main loss for frame-level action detection in this thesis.

2.3 Evaluation Metrics

In this section, we revise the common action detection evaluation metrics. We first revise the basic concepts for the evaluation of the detection task, and then we revisit the main evaluation metrics in the event-level and frame-level action detection.

2.3.1 Basic Concepts

For each class $c \in 1, 2, ..., C$ in the dataset, we denote TP^c, FP^c, TN^c and FN^c the number of True-Positive, False-Positive, True-Negative and False-Negative frames, respectively.

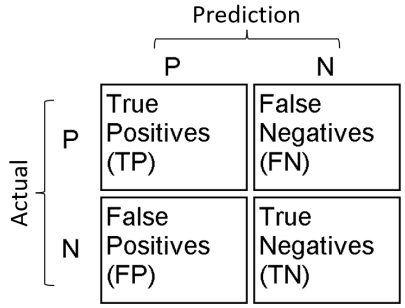


Figure 2.2: Confusion matrix. "P" indicates the Positive (i.e., with action) and "N" indicates the Negative (i.e., without action).

These four parameters are used to calculate many kinds of performance evaluation metrics. The logical details of these four parameters are shown in Fig. 2.2. By utilizing these parameters we can compute the following two measures:
(1) **Precision** (P^c) is the percentage of the predicted real positive samples in predicted results. The formula is as follows:

$$P^c = \frac{TP^c}{TP^c + FP^c} \tag{2.2}$$

(2) **Recall** (R^c) is the coverage of predicting correctly. Specifically, recall corresponds to how many real positive samples in the testing set were identified. The formula is as follows:

$$R^c = \frac{TP^c}{TP^c + FN^c} \tag{2.3}$$

(3) **Intersection over Union** ($t - IoU$) can be understood as the overlap between the predicted instance by the model and the ground truth instance for the action detection in an untrimmed video. The calculation formula is the intersection of Detection Result and Ground Truth compared to their union. IoU is used to check whether the IoU between the predicted results and the ground truth is greater than a predicted threshold.

2.3.2 Event-level Evaluation

The datasets that contain videos with sparse action occurrences leverage the event-level (i.e., instance-level) evaluation. There are two main evaluation metrics: (1) **Average Recall** (AR) summarises the distribution of recall across a range of overlap thresholds, which is the principle evaluation metric for temporal action proposals generation. Because proposal generation is irrelevant to category classification, i.e., it only focuses on finding the action instance boundaries. AR are often used to measure the completeness of generated temporal proposals, which is critical for proposal generation models. Normally, AR is defined as the mean of all recall values using tIoU thresholds between 0.5 and 0.95 with a

step size of 0.05. (2) **Mean Average Precision** (mAP) [109] is the evaluation metrics which is the most commonly used in the community. This metrics is similar with metrics used in object localization task. The general definition for the Average Precision (AP) is finding the area under the precision-recall curve. mAP is the average of AP over all action class categories.

2.3.3 Standard Frame-level Evaluation

In order to evaluate videos with dense action occurrences, previous methods chose frame-wise metrics [110, 13, 83, 2]. One of the common metrics is the **Frame-wise Accuracy** (FA_1), which represents the ratio of correctly classified frames to all frames in the dataset. Frame-wise accuracy is defined as:

$$FA_1 = \frac{\sum_c TP^c}{\sum_c N_c} \tag{2.4}$$

where N_c is the number of the frames in the dataset for class c. Note that this metrics is sensitive to the class distribution but provides an intuitive measure of the algorithm ability to recognize actions. A second metrics is the **F-Score**, which combines precision (P^c) and recall (R^c) for each class c and is defined as the harmonic mean of these two values:

$$F - Score = \frac{2}{|C|} \sum_c \frac{P^c \times R^c}{P^c + R^c} \tag{2.5}$$

where P^c and R^c are precision and recall metrics of class c respectively. As the focus of our work is to evaluate the model performance with real-world dense labeling videos, we evaluate our models using **Average Precision** (AP) measured on our frame-level labels [13]. For every action class c, we compute its own average precision AP^c. **Frame-level mean average precision** ($F - mAP$) is the mean value of AP^c for all the classes (C).

$$F - mAP = \frac{1}{C} \sum_{c \varepsilon C} AP^c \tag{2.6}$$

This metric can handle the case where one frame contains multiple labels.

2.3.4 Action Dependency Metrics

As mentioned earlier, the problem of temporal action detection consists in predicting the action, or actions, occurring at each time-step of a video. The standard metrics for evaluating temporal action detection performances, such as $F - mAP$, treats each time-step as an individual sample. More specifically, $F - mAP$ measures the performance of each class independently, and averages their scores, but it overlooks the measure if models learn

the relationships between these classes. This issue is not unique to $F-mAP$. Other per-frame action detection metrics or multi-label classification metrics [111, 112, 113] also do not consider the relationships between different action classes or time-steps, which makes them unsuitable to evaluate how well action dependencies are modeled. Such property is important to evaluate if the model can handle the complex temporal relation among different action instances in a video. To this end, Tirupattur et al. [7] propose a set of action detection metrics which measure the ability to model both co-occurrence dependencies and temporal dependencies of the proposed method.

We first introduce the Action Conditional Precision and Recall. As presented in the basic concepts, the standard precision and recall measure a model performance in individual classes. However, they do not take into account the relationships and dependencies between these classes. Action-conditional precision and recall can solve this issue. For an action class c_i, we measure its action conditional precision and recall when another action (c_j) is present within a temporal window τ [7] following:

$$P_{AC}(c_i|c_j,\tau) = \frac{N_{correct}(c_i|c_j)}{N_{predict}(c_i|c_j)} \tag{2.7}$$

$$R_{AC}(c_i|c_j,\tau) = \frac{N_{correct}(c_i|c_j)}{N_{gt}(c_i|c_j)} \tag{2.8}$$

These metrics measure the precision and recall of an action c_i, given that action c_j was present within the last τ time-steps. We also want c_j not to be present within the current time-step as this condition ensures that it measures only temporal dependencies and not co-occurrence dependencies. For a given video k and at time-step t, we formulate such condition as:

$$y^{(k)}_{t,c_j} = 0 \bigcap \exists y^{(k)}_{t^*,c_j} = 1, t^* \in [t-\tau, t) \tag{2.9}$$

Therefore, the action-conditional precision (P_{AC}) and recall (R_{AC}) can be computed with the following equations:

$$N_{correct}(c_i|c_j,\tau) = \sum_{k,t} \mathbb{1}[y^{(k)}_{t,c_i} = \tilde{y}^{(k)}_{t,c_i} = 1]\mathbb{1}[\mathcal{X}] \tag{2.10}$$

$$N_{predict}(c_i|c_j,\tau) = \sum_{k,t} \mathbb{1}[\tilde{y}^{(k)}_{t,c_i} = 1]\mathbb{1}[\mathcal{X}] \tag{2.11}$$

$$N_{gt}(c_i|c_j,\tau) = \sum_{k,t} \mathbb{1}[y^{(k)}_{t,c_i} = 1]\mathbb{1}[\mathcal{X}] \tag{2.12}$$

Here, $\mathcal{X}$ is the condition in equation 2.9. P_{AC} and R_{AC} can measure the precision and recall of an action class c_i when c_j is present within the given time-step and these metrics

are not symmetric.

For measuring the capacity for handling the co-occurrence relationship, we compute the formulation with $\tau = 0$. In practice, when $\tau = 0$, we replace the $\mathbb{1}[\mathcal{X}]$ with $\mathbb{1}[\tilde{y}^{(k)}_{t,c_j} = 1]$ in Eq. 2.10, Eq. 2.11 and Eq. 2.12. After that we can measure the capacity of the proposed model for detecting two actions occur within the same time-step.

Since some actions never co-occur or follow each other, the overall metrics is computed by averaging all action pairs $(c_i, c_j), i \neq j$, such that $N_{gt}(c_i|c_j, \tau) > 0$. More complex performance metrics like F1-score and mAP can also be computed using the action-conditional precision and recall metrics.

2.3.5 Conclusion

Frame-level metrics are robust to annotation ambiguity [13, 2]. However, event-level evaluation metrics enable us to get a better insight into action detection as this metric is not biased by action duration. Moreover, event-level evaluation measures the continuity and completeness of the action prediction, which are overlooked in frame-level metrics. For this reason, although most action detection methods on densely annotated datasets [13, 2, 83] still rely on frame-level metrics [16, 17, 18, 7, 19], we believe that future action detection algorithms should focus more on event-level evaluation.

2.4 Datasets

In this section, we describe the datasets that are used to evaluate our method in this dissertation. More discussion about action detection datasets in the community is provided in chapter 3.

2.4.1 Untrimmed datasets

Depending on the density of annotations, there are two kinds of datasets for action detection: (1) Sparsely labelled [11, 78, 64, 12] and (2) Densely labelled [13, 2, 83, 84] datasets. Densely labelled datasets contain more foreground action instances and may include fine-grained actions occurring concurrently. As they are more challenging and closer to real-world scenarios [13], more and more attention is given to densely labelled datasets. In this thesis, we focus on the densely annotated datasets.

Charades [114] was recorded by hundreds of people in their private homes. This dataset consists of 9848 videos across 157 actions. The actions are mainly object-based daily living actions performed at home. Each video is about 30 seconds containing complex co-occurring actions. In our experiments, we follow the original Charades settings for action

detection [114] (i.e. Charades v1 localize evaluation). The performances are measured in terms of mAP by evaluating per-frame prediction.

DAHLIA [110] is a large ADL dataset for detection. Contrary to some widely used datasets, in which labelled actions are very short and with low-semantic level, DAHLIA focuses on high-semantic level longer actions. It contains 8 ADL action classes performed by 51 subjects on 3 camera views. The duration of videos ranges from 24 *mins* to 64 *mins*. In each video, an average of 6.7 actions are performed. The mean duration of actions is 6 *mins*. By default, we performed experiments using the cross-subject protocol. The final result is obtained as the average of the results on the 3 camera views.

Breakfast [83] features over $1.7k$ video sequences of cooking in a kitchen environment. The overall duration is $66.7h$. The dataset contains 48 action classes. In each video, an average of 4.9 actions are performed. The mean duration of actions is about $30s$. Actions are thus shorter than those in DAHLIA, but they are more diverse. We performed our experiments using the protocol described in [100].

PKU-MMD [64] covers a wide range of complex human actions with well annotated information. This dataset contains 1076 long video sequences in 51 action categories, performed by 66 subjects. PKU-MMD provides multi-modality data sources, including RGB, depth, Infrared Radiation, and Skeleton. Following the original paper of PKU-MMD, the performances are evaluated in terms of event-based mAP in Cross-Subject protocol (CS).

THUMOS14 [11] and **MultiTHUMOS** [13]: Different from the aforementioned daily living action datasets, THUMOS datasets contain sport videos form YOUTUBE. There are two version of THUMOS datasets in the community (see Fig. 2.3), we choose MultiTHUMOS as the main dataset in this thesis, which is an enhanced version of the THUMOS14 dataset with dense annotations. This dataset consists of 65 action classes, compared to 20 in THUMOS14, and contains on average 10.5 action classes per video and 1.5 labels per frame and up to 25 different action labels in each video. THUMOS14 and MultiTHUMOS consists of YouTube videos of various sport actions like baseball games or cliff diving.

2.4.2 Trimmed datasets

In this thesis, it involves several datasets for action recognition tasks. Note that, there are multiple tasks for EPIC-Kitchen [1] and EGTEA Gaze+ [115] datasets. In this work, we utilizes only the trimmed version of these datasets for action recognition task.
NTU RGB+D [62] is acquired with a Kinect V2 camera and consists of 56880 video samples with 60 action classes. The actions were performed by 40 subjects and recorded from 80 viewpoints. For each frame, the dataset provides RGB, depth, and a 25-joint skeleton

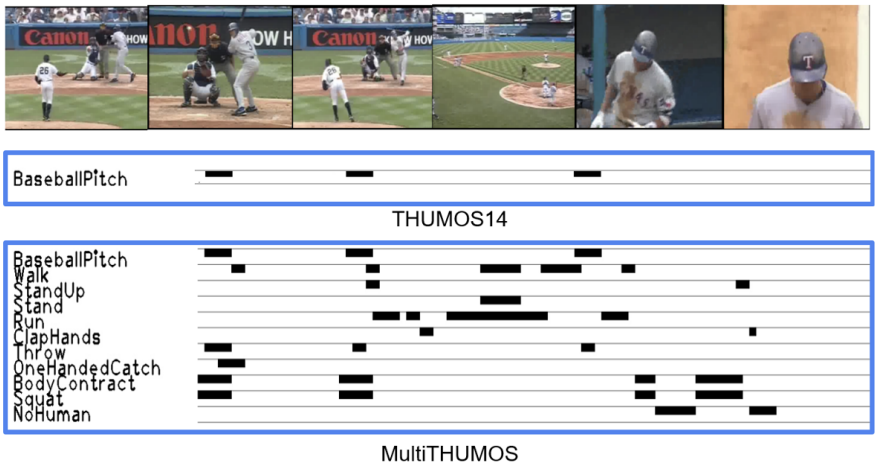


Figure 2.3: THUMOS dataset. (1) THUMOS14: the sparsely annotated version and (2) MultiTHUMOS: the densely annotated version.

of each subject in the frame.

EPIC-KITCHENS 55[1] is an egocentric dataset which contains 55 hours of recording of 32 different kitchens in 4 cities. This dataset focuses on fine-grained cooking actions with a total of 125 verbs and 352 nouns.

EGTEA Gaze+ [115] is an egocentric dataset which contains 28 hours of cooking actions from 86 unique sessions of 32 subjects, with over 10k video clips of 106 fine-grained egocentric actions.

2.5 Conclusion

In our literature survey we have observed that current action detection methods rely on an effective visual backbone to extract the spatio-temporal features and on a temporal module to model the temporal dependencies among different temporal features. The main focus for current action detection methods lies in the second step, i.e., how to effectively model the temporal relations in the long-term video. Recently, 1D temporal convolutional networks have become an obvious choice for temporal modelling in videos but with additional functionalities to address the complex temporal relations in densely annotated videos. However, limited by the local operation of convolution, how to model the complex temporal relations and how to model both local and global dependencies remain challenging for temporal convolutional networks. To tackle this issue, we study different directions to enhance the vanilla temporal convolutional network in the following chapters by: (1) self-attention mechanism in chapter 4, (2) semantic reasoning in chapter 5,

and (3) additional modalities in chapter 6.

In the next chapter, we propose a novel dataset Toyota Smarthome Untrimmed to include the challenges in real-world indoor action detection. We further compare this new dataset with current benchmarks in the next chapter.

Chapter 3

Toyota Smarthome Untrimmed: Real-World Untrimmed Videos

In this chapter, we introduce a challenging indoor dataset: Toyota Smarthome Untrimmed (TSU) that features many real-world challenges for action (i.e., activity) detection. Constructing this dataset is part of a Toyota project that aims at developing a "Smarthome" indoor video understanding system for the elderly living alone at home. This system is intended to work with a partner robot - Toyota HSR. The real-time information will be sent to the robot to better interact with the older adult and to facilitate their life. With TSU, researchers can develop novel approaches to promote Smarthome activity detection systems in the wild. This work has been accepted by IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) [43].

3.1 Introduction

According to a recent report of the United Nations [116], the global population aged 60+ is projected to grow from 0.9 billion in 2015 to 1.4 billion in 2030. This demographic trend results in a dramatic need for an increase of the workforce in healthcare. A great support to the healthcare workforce could come from activity detection systems, which help monitor the health state of older patients and could early detect potential physical or mental disorders. For instance, monitoring patients' eating habits allows doctors to track the state of a patient and to react before serious health conditions arise. Thanks to such systems, seniors could stay longer at home without the need of being hospitalized, which would greatly improve their comfort and quality of life. Building such activity detection systems requires fine-grained understanding of long untrimmed videos.

In recent years, numerous datasets for activity classification in trimmed videos have been proposed [62, 63, 41], whereas very little has been done for activity detection in

untrimmed videos. By activity detection, we mean predicting the activity labels as well as their temporal boundaries within an input video. This detection task has to cope with important open challenges: i) handling the combinatorial explosion of activity proposals while detecting accurate temporal boundaries in long video sequences, ii) managing concurrent activities, and iii) distinguishing between background and foreground activities (e.g. *standing still/using telephone*). In this work, we focus on untrimmed videos of Activities of Daily Living (ADLs). These videos contain activities that usually occur in the daily lives of older people. Typically ADLs feature activities with similar motion (e.g. *eating/drinking*), activities with high temporal variance (e.g. *putting on glasses* in 5 sec./ *reading* for 10 min.), or subtle motions (e.g. *stirring the coffee*).

Most of the untrimmed video datasets that are widely adopted in the literature do not focus on ADLs. These datasets are often collections of videos from the web [78, 11, 117, 118, 13, 12]. For instance, ActivityNet [78] and MultiTHUMOS [13] are collections of a large number of videos encompassing sports and outdoor activities. These activities are often characterized by high inter-class variation due to large and distinctive motions. Other datasets contain movie excerpts or instructional videos [119, 120]. The videos in these datasets retain only the key part of the activity and are mostly recorded by a cameraman from a frontal viewpoint, with nearly no occlusions.

Some ADL datasets have been proposed in the past few years [2, 64, 110]. These datasets share common characteristics: i) Subjects usually follow a rigid script, which results into unnatural movements; ii) Videos and thus activities are usually short; iii) Subjects are usually centered in the middle of the frame and perform activities facing the camera (i.e. high camera framing). These characteristics do not reflect the spontaneity of human activities in real-world scenarios.

Motivated by the shortcomings of current datasets, we introduce Toyota Smarthome Untrimmed (TSU). TSU provides realistic untrimmed videos with diverse spontaneous human activities and real-world settings. We invited 18 volunteers to a recording session in a smart home. The volunteers are senior people in the age range of 60 to 80 years. Their daily lives were recorded by multiple cameras in the apartment. The resulting data consists of 536 long RGB+D videos with 51 annotated activity classes. This dataset is an extension of our previously published dataset [50], which was designed for the classification task of clipped videos. Unlike most previous datasets in the community, the TSU videos are unscripted. Activities are annotated with both coarse and fine-grained labels. This dataset poses several challenges: high intra-class temporal variance, high class imbalance, composite and elementary activities, and activities with similar motion. In our data acquisition process, each participant was recorded continuously for 8 hours. We believe that this setup reduced camera awareness in the participants, leading to increased spontaneity. Consequently, in TSU, the participants may commit errors, search for items,

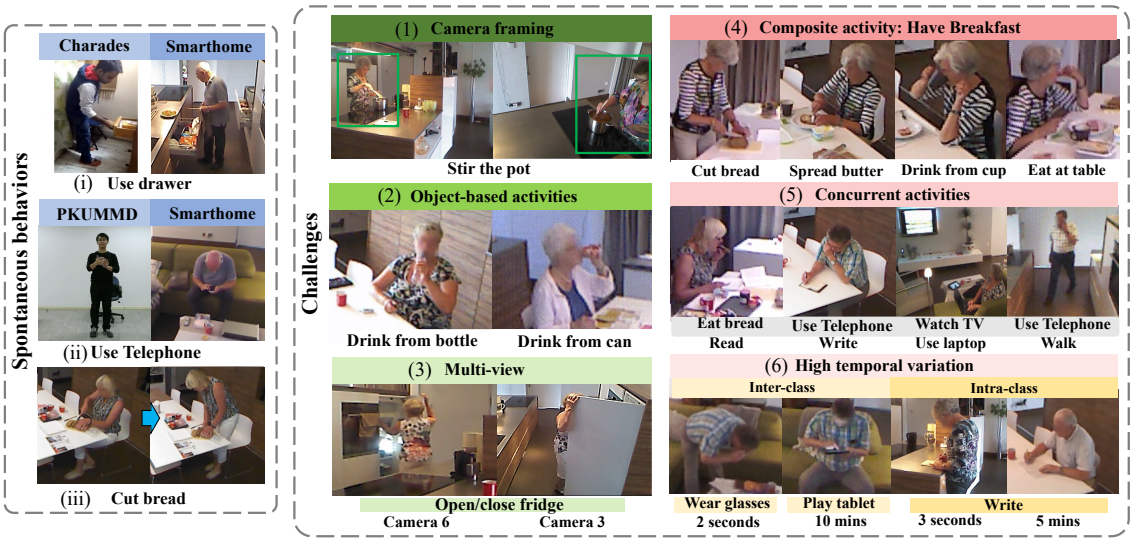


Figure 3.1: Overview of the challenges in TSU.

and repeat several times the same activity before succeeding. The fact that activities are performed in a spontaneous manner also amplifies other challenges such as low camera framing and high temporal variance.

Some of the challenges in TSU dataset are illustrated in Fig. 3.1. In this figure, on the left part, we present challenges related to **spontaneous behaviours**: For the first two examples, we present the activity following a strict script on the left, and the same activity performed spontaneously in TSU on the right: i) In Charades [2], *using drawer* is performed quickly once per video, in TSU, *using drawer* may be repeated several times in a video, and the subject may keep several drawers open at the same time to facilitate finding things. ii) In PKUMMD [64], the subject shortly uses the *telephone* while looking at the camera. In contrast in TSU, the subject is deeply involved with his *telephone* and the activity may last several minutes instead of few seconds. iii) In TSU, subject may stay seated or *stand up* to *cut the bread* in an easier manner.

Besides the spontaneous behaviours, we also illustrate on the right part the following real-world challenges: 1) Camera framing: subject is not in the middle of the image and can be even outside the field of view. 2) Object-based activities: similar activities can be performed while interacting with different objects. 3) Multi-views: activities look differently from different view points. 4) Composite activity: composite activities can be split into several elementary activities (e.g. instead of *having breakfast*, we may *cut bread*, *spread butter* and *eat at the table*). Moreover, these complex composite activities can last a long period of time. Large variations of appearance make the recognition challenging, requiring to understand the composition of elementary activities to better recognize the composite activities. 5) Concurrent activities: activities can be performed concur-

rently (e.g. *take note* while *having a phone call*). 6) High temporal variation: in the same untrimmed video, we may have short activities (e.g. *taking on glasses*) and long ones (e.g. *playing tablet*). Different instances of the same activity class can also be short or long (e.g. *writing*) corresponding to high intra-class temporal variance. In section 3.3, we analyse in detail the characteristics and novelty of the proposed dataset.

Experimentally, we find out that state-of-the-art activity detection methods fail to address the aforementioned real-world challenges offered by TSU. We also find that the model trained on the untrimmed TSU outperforms the same model trained on the trimmed version [50], reflecting the difficulty of handling background actions.

In general, the low performance achieved by activity detection methods on TSU highlights the many challenges that are yet to be addressed. To promote the development of novel activity-detection methods that can better address such challenges, we have released TSU to the research community.

3.2 Related Work

In this section, we give an overview of publicly available untrimmed activity detection datasets.

Dataset	Spontaneous behaviour	Camera framing	Object-based activities	Multi-view	Composite activities	Concurrent activities	Var. activity duration	Temporal annotation	View type	Video type
MEVA[121]	High	Low	Yes	No	No	No	Low	Precise	Monitoring	Surveillance
ACTEV/VIRAT[122]	High	Low	Yes	No	No	No	Low	Precise	Monitoring	Surveillance
DALY[118]	Medium	High	No	No	No	Yes	Low	Precise	Shooting	Web
HACS[12]	Medium	High	Yes	No	No	No	Medium	Precise	Shooting	Web
YouTube'8M-Segments[117]	Medium	High	No	No	No	No	Low	Noisy	Shooting	Web
ActivityNet-200[78]	Medium	High	Yes	No	No	Few	Medium	Precise	Shooting	Web
THUMOS14[11]	Medium	High	No	No	No	No	Low	Precise	Shooting	Web
MultiTHUMOS[13]	Medium	High	No	No	No	Yes	Medium	Precise	Shooting	Web
AVA[119]	Medium	High	No	No	No	Yes	Low	Precise	Shooting	Movie
How2[123]	Low	High	Yes*	No	-	-	-	Noisy	Shooting	Instructional
HowTo100M[120]	Low	High	Yes*	No	-	-	-	Noisy	Shooting	Instructional
Coin[124]	Low	High	Yes	No	No	No	Medium	Noisy	Shooting	Instructional
ADL[125]	High	High	Yes	No	No	No	Low	Precise	Egocentric	ADL
Charades-ego[126]	Medium	High	Yes	No	No	No	Low	Precise	Egocentric	ADL
50 Salades[84]	Medium	High	Yes	No	No	No	Low	Precise	Top-view	Cooking
EGTEA Gaze+[127]	Medium	High	Yes	No	No	No	Low	Precise	Egocentric	Cooking
EPIC-KITCHENS[1, 128]	High	High	Yes	No	Few	Yes	High	Precise	Egocentric	Cooking
MPII Cooking 2[129]	Low	High	Yes	No	woT	No	Medium	Precise	Shooting	Cooking
Breakfast[83]	Medium	Medium	Yes	Yes	woT	No	Medium	Precise	Shooting	Cooking
CAD-120[130]	Low	High	Yes	No	Yes	No	Low	Precise	Shooting	ADL
DAHLIA[110]	High	Low	No	Yes	No	No	High	Precise	Monitoring	ADL
PKU-MMD[64]	Low	High	No	Yes	No	No	Low	Precise	Shooting	ADL
Charades[2]	Low	High	Yes	No	No	Yes	Low	Precise	Shooting	ADL
Toyota Smarthome Untrimmed	High	Low	Yes	Yes	Yes	Yes	High	Precise	Monitoring	ADL

Table 3.1: Untrimmed dataset comparison along the seven real-world challenges. *Indicates that the activity labels are provided in terms of caption. With 'woT' we indicate that the composite labels are provided without the corresponding temporal boundaries. Although MPII Cooking 2 was recorded in multi-camera scenario, the authors have released only a single view version.

The availability of videos replicating real-world challenges is crucial to design robust activity detection algorithms. Among existing datasets, only few of these challenges are properly addressed. To understand the limitations of currently available datasets, we introduce the following 7 real-world challenges.

Spontaneous behaviour: activities in the real-world are performed naturally. However, most existing datasets are acquired by providing the subjects with a strict script. Besides, as the subjects are aware that their activities are being recorded, they often overact. To quantify spontaneous behaviour, we define a heuristic that considers three aspects: (i) Scripted or unscripted: The datasets following a strict script always have lower spontaneity. We assign the datasets that follow strict script 1 point; the datasets following a coarse script (e.g. cooking a specific meal in a video) 0.5 point; unscripted 0 point. (ii) Camera Awareness: Camera awareness also affects spontaneity. We assign 1 point to the datasets recorded by the cameraman/self-recorded/wearable sensor. We assign 0.5 point to datasets with continuous videos that were recorded for a long duration (at least 30 minutes). For monitoring datasets recorded for a long duration, we assign 0 point. (iii) Environment: it is also an important factor for spontaneity. Activities are often more spontaneous when performed in a familiar environment. Here, we assign a dataset that is recorded in an unfamiliar location 1 point, in a familiar location (e.g. home) 0 point. Datasets with continuous videos that were recorded for a long duration in the same environment are given 0.5 point, as people get accustomed to the location. Following these criteria, we re-evaluate all datasets. The datasets with less than 1 point are considered as featuring high spontaneity, more than 2 points obtained low spontaneity, the others are rated with medium spontaneity.

Camera framing: when videos are recorded by a cameraman, subjects mostly appear in the middle of the image and facing the camera (high camera framing). On the other hand, when videos are recorded automatically by a monitoring system using fixed cameras, subjects can often be offset from the center, occluded or partially outside the field of view (low camera framing).

Object-based activities: similar activities that can be performed while interacting with different objects (e.g. *drinking from cup* or *from bottle*) are more challenging to classify. In Table 3.1, *object-based activities* indicates the availability of object level fine-grained annotation for these activities.

Multi-views: activity detection methods need to be robust against view-point variations. Therefore, benchmark datasets should provide samples of the same activities recorded from different views.

Composite activities: Some complex ADLs can be decomposed into several elementary activities. For example, *having breakfast* may contain elementary activities like *cutting bread*, *spreading butter* and *eating at table*. In Table 3.1, the *composite activities* column indicates whether the dataset provides annotation for both composite activities and their respective elementary activities.

Concurrent activities: activities, such as *making a phone call* and *taking notes* may be performed simultaneously. The appearance of activities can drastically change when mul-

tiple activities are performed at the same time. In Table 3.1, *concurrent activities* indicates whether the dataset provides samples and annotations in which activities are performed simultaneously.

Variation of activity duration: this property indicates the level of variation in the length of activities in the dataset. In this table, the high variation indicates that the average duration of an activity class is more than 80 times larger than the one of the lowest activity class. The low variation indicates that the highest average duration of an activity class is less than 30 times than the one of the lowest activity class.

To be noted that, activity detection methods need precise temporal annotation (i.e. start time and end time) for each activity. We consider that a dataset features Noisy annotation when: (i) the dataset provides temporal annotation only for part of the activities in the video [117], or (ii) the dataset only provides caption of the video [120, 123].

Table 3.1 summarizes the comparison of most used public untrimmed video datasets based on the above challenges. Below, we detail how these untrimmed datasets differ from our proposed TSU.

3.2.1 Surveillance Datasets

Surveillance datasets, such as VIRAT and MEVA [122, 121], have fixed camera views and are designed to monitor human activities in the wild. These datasets are collected in natural scenes showing people performing normal activities in standard contexts, most of the time outdoors. Besides, activities look natural as they are performed by actors following a light script. For these datasets, only few simple human activities are annotated (i.e. crouching, standing...) along with the object information (i.e. carrying a box). These datasets thus differ from TSU as the complexity of surveillance activities is significantly lower than the one from daily-living activities, for example, no concurrent & composite activities.

3.2.2 Web & Youtube & Movie Datasets

A large number of datasets are collected from YouTube or movies [118, 117, 12, 78, 11, 13, 119]. Most of these videos are self-recorded or recorded by a cameraman from a single view, which causes the subject to be centered within the image frame (i.e. high camera framing), facing the camera and with limited occlusions. These videos are carefully selected and only the key parts of the activities are retained, in which the subjects always perform the activities smoothly without hesitation in front of the camera (i.e. reduced spontaneity). Thus, these videos are less representative of real-world scenarios compared to TSU videos.

3.2.3 Instructional Videos

Similar to the above category of datasets, instructional videos [123, 120, 124] are collected from internet sources. These videos provide intuitive visual examples for learners to acquire knowledge to accomplish different tasks. In contrast to TSU, these instructional videos have noisy annotations which are often text descriptions [123, 120] and follow strict temporal ordering of the activities [124]. Similar to web videos, the subjects always perform the activities smoothly without hesitation in front of the camera [124, 120]. These characterizations of the instructional videos are not adequate for real-world activity detection task.

3.2.4 Activities of Daily Living (ADL)

Activities of daily living are performed in indoor environments such as homes or labs. These activities are usually characterized by low inter-class variation and subtle motion. Below, we discuss the ADL datasets categorized by their camera viewing angle.

Egocentric view datasets: In Egocentric datasets [1, 128, 126, 127, 125], the videos are recorded with a wearable camera (i.e. reduced spontaneity) or from a top view [84] that captures the scene directly in front of the user at all times, in which only hands are visible in the center of the camera view (i.e. high camera framing). Egocentric videos are designed to study the activities, where the user's hands are manipulating various objects. However, the egocentric paradigm can only collect the activity information from a very restricted viewpoint. This restricted viewpoint makes the appearance of egocentric activities very different from third person view datasets like TSU (e.g. poses are unavailable) and prevent the recording of those activities that cannot be observed from this viewpoint (e.g. *making a phone call*). Due to these characteristics, our comparison mainly focuses on third-view datasets.

Third person view datasets: Many of the ADL datasets [83, 129, 131] are limited to kitchen activities. **MPII Cooking 2** [129] and **Breakfast** [83] contain only cooking activities (like *preparing recipes or making breakfast*). The subject is asked to cook a single dish (i.e. composite activities) in each video in these datasets. However, there are no temporal boundaries (i.e. no ground truth timestamps) for the composite activities as they correspond to a whole video. Besides, some of the composite activities occur only once in the dataset. In these datasets, subjects are asked to prepare a specific recipe in a video clip, therefore the activities are performed in rapid succession without hesitation or mistake (reduced spontaneity). Moreover, the dataset lacks the presence of secondary activities irrelevant to *cooking* (e.g. *drinking water*) but often occurs in real-life. In **MPII Cooking 2**, the subjects follow strict scripts (i.e. low spontaneity) and are always in the center of the frame (i.e. high camera framing). Although the videos are recorded from 8

camera views, only a single view is released for this dataset. In **Breakfast**, the hands of the subjects and the objects used are always at the center of the frame without much occlusion (i.e. medium camera framing). Moreover, the number of views are not fixed, even in the same kitchen (from 2 to 5). As mentioned, the subjects in these two datasets perform the activities quickly without much hesitation, which means the datasets are characterized by medium temporal variation and no concurrent activities. So, in the following, we present the datasets that encompass a larger variety of ADLs which are not only restricted to kitchen activities and where not only the top body part can be observed.

CAD-120 [130] is a small dataset (about 60 K frames in total). This dataset comprises of 20 different activities (including composite and object-based activities) performed by four people in different rooms. The subjects are always in the center of the scene performing short sub-activities following a script (i.e. high camera framing & no spontaneity). Because of the simplicity of activities, current state-of-the-art methods [132, 133] can already achieve excellent results on this dataset. **DAHLIA** [110] is recorded in a single room in a lab with 44 subjects. Each subject has about 40 min recording from 3 fixed camera views (i.e. high spontaneity, low camera framing). The dataset contains only 8 coarse activity classes, thus it does not have the challenges of concurrent, composite and object-based activities. In **PKU-MMD** [64], the videos are recorded from 3 camera views. The activities are performed in the center of the scene by the subjects following a strict script. Besides, there are pauses in between the activities which makes the problem of distinguishing between an activity and background easier compared to real-world scenarios. Thus, this dataset lacks spontaneity & concurrent activities in addition to high camera framing. **Charades** [2] explores object-based activities and concurrent activities. The videos are recorded by hundreds of people in their private homes following strict scripts. Although Charades depicts large numbers of environment diversity, these self-recorded activities are very short (30 sec./video, 10 sec./activity) with low variation of activity duration and in general performed in unnatural manner (overacted), in the center of the camera view (high camera framing). All in all, current ADL datasets address only partially the 7 aforementioned challenges of real-world scenarios. This motivates us to propose TSU.

3.3 Toyota Smarthome dataset

In this section we describe the main features of Toyota Smarthome Untrimmed dataset. Our goal is to create a large scale dataset with daily-living activities performed in spontaneous manner.

3.3.1 Data Collection

3.3.1.1 Collection Setup

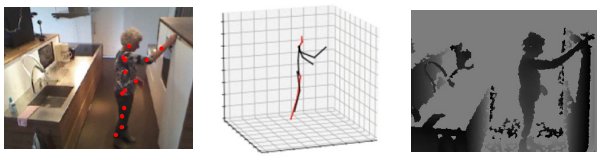
We use 7 Microsoft Kinect sensors in the recording phase. The apartment plan and camera locations are shown in Fig. 3.5. Cameras 1 and 2 cover the dinning room area, 4 and 5 the living room, 3, 6 and 7 the kitchen. Thus, we have a coverage over the entire apartment from at least 2 distinct viewing angles. The videos are recorded at 20 frames per second, the size of RGB is VGA (640×480), the standard resolution in most real-world scenarios. The dataset offers 3 modalities: RGB, depth and 3D skeleton (i.e. pose) (see fig. 3.2).

For the skeleton modality, we fine-tune LCR-Net++ [134] on TSU and then extract the 2D skeletons. As the video recording is unconstrained, the subject may be partially occluded by the objects or equipment in the scenarios. For this reason, we utilise our SSTA-PRS [52] to refine the prediction of 2D skeletons. Finally these 2D skeletons are processed through VideoPose3D [135] to extract the 3D skeletons. We observe that this mechanism extracts 3D poses of better quality compared to those obtained using depth or LCRNet++ [52].

3.3.1.2 Data Collection Protocol

One of the key applications of daily-living activity detection is older patient monitoring. Thus, in our dataset, we invited 18 volunteers to our dataset recording sessions. The age of the volunteers ranges between 60 and 80 years old. Each volunteer was recorded for 8 hours in one day starting from morning at 9 a.m. until afternoon at 5 p.m.. On the day of recording, the volunteer arrived in the apartment at 8 a.m. and had a visit to get acquainted with the place and to learn how to use the household equipment such as coffee machine, television, remote control, etc.. The volunteers also received an informal description of what it was expected with reference to having meals and interacting with anything in the apartment as it was a normal day at home. No further guidance was provided about how the activities should be performed.

In total, we recorded hundreds of hours of video data. Based on these data we prepared two datasets: Toyota Smarthome dataset [50], previously published, and Toyota



(1) RGB (2) 3D Skeleton (3) Depth

Figure 3.2: Available modalities in Toyota Smarthome Untrimmed. Note: in the sub-figure of RGB modality, we also mark the 2D skeleton joints.

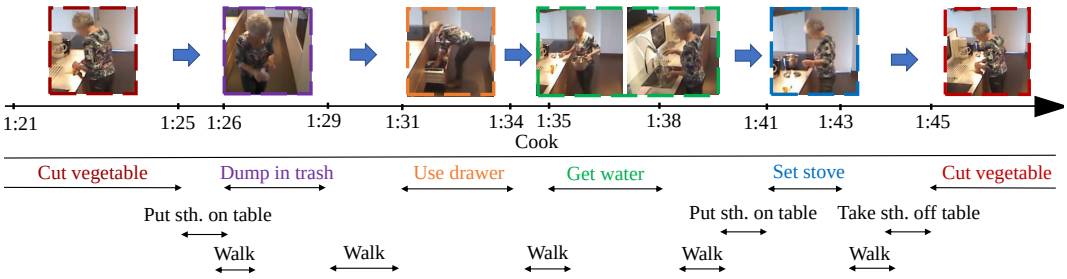


Figure 3.3: An example of annotation on TSU dataset. '←' and '→' indicate respectively the start and end of an activity.

Smarthome Untrimmed dataset that is introduced in this paper.

3.3.2 Toyota Smarthome Trimmed dataset

Toyota Smarthome Trimmed [50] has been designed for the activity classification task. It consists of 16K short RGB+D clips of 31 activity classes. Each clip is about 12.5 sec. long and contains only one activity. Unlike previous datasets [62, 63], activities were performed in a natural manner. As a result, the dataset poses a unique combination of challenges: high intra-class variation, high class imbalance, and activities with similar motion and high duration variance. Activities were annotated with both coarse and fine-grained labels. These characteristics differentiate Toyota Smarthome Trimmed from other datasets for activity classification.

3.3.3 Toyota Smarthome Untrimmed dataset

Toyota Smarthome Untrimmed and Toyota Smarthome Trimmed are obtained from the same recording footage. Different from the Toyota Smarthome Trimmed, TSU is targeting the activity detection task in long untrimmed videos. Therefore, in TSU, we kept the entire recording when the person is visible. The dataset contains 536 videos with an average duration of 21 mins. Since this dataset is based on the same recording as Toyota Smarthome Trimmed version, it features the same challenges and introduces additional ones. In section 3.3.3.1, we describe the annotation protocol. Then, we present the properties of the TSU dataset in section 3.3.3.2, we present its challenges in section 3.3.3.3, and finally we compare this untrimmed version of the dataset (i.e. TSU) with its trimmed version in section 3.3.3.4.

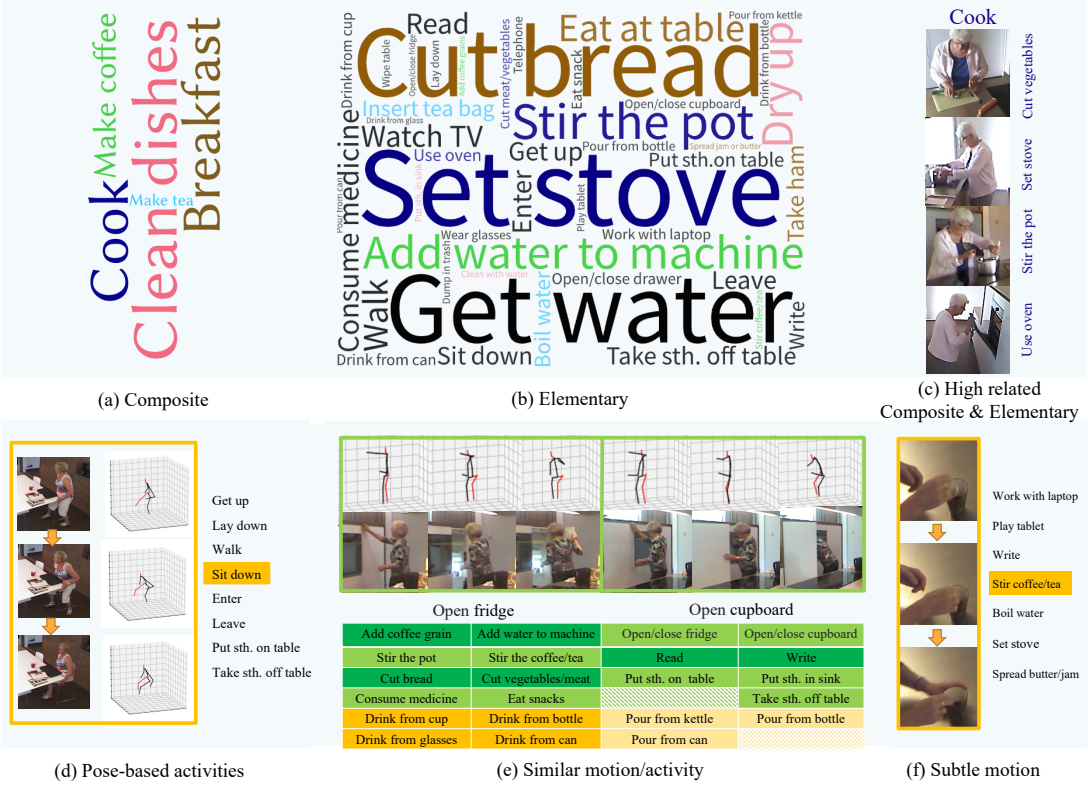
3.3.3.1 Annotation Protocol

TSU is designed particularly for the activity detection task. With the support of a medical staff, we have identified 51 activities of interest to annotate. A team of annotators manually annotated the videos using the open-source toolkit ELAN [136]. The videos were annotated individually without relying on the fact that some camera views overlap. The annotation process took more than 6 months, including verification and quality checks. We performed the quality check with the help of 5 annotators. We estimated the precision of the annotation by considering the same 50 long videos annotated by different annotators. These 50 videos are randomly chosen and cover all the subjects and camera views. The precision of annotation of those 50 videos is 96.8%. Additionally, we reviewed, normalized and corrected the 25 hours of annotation by checking again the videos where the methods were achieving low activity detection performance. Fig. 3.3 shows an example of the annotation. This example corresponds to composite activity *cooking*. While *cooking*, the subject abruptly stops cutting vegetables and starts heating water in a pot so that she can have boiled water after cutting the vegetables. After setting up the stove, she resumes cutting the vegetables. This process does not follow a strict temporal order and reflects the spontaneous behaviour of the participant.

3.3.3.2 Dataset Properties

The result of the extensive annotation process is a rich corpus of activities. Fig. 3.4 presents the diversity of activities in this dataset. The activities are categorized into composite and elementary activities. **Composite activities** are the complex activities that are composed of several **elementary activities** that may or may not follow a temporal ordering. TSU contains 5 composite activities which are relatively long. Elementary activities are atomic activities which may be performed concurrently in time. These activities may or may not be part of a composite activity. TSU contains 46 elementary activities and these activities may be long or short. In Fig. 3.4 (c), we illustrate the composite activity *cooking*, with its elementary activities. In Fig. 3.4 (a) and (b), the composite and its corresponding elementary activities are marked with the same color.

TSU contains a rich diversity of elementary activities. We present three challenging scenarios that might occur while attempting to recognize these activities. Firstly, the dataset contains pose-based activities for which poses could be sufficient for classification. In contrast, the appearance information may not improve the recognition of these activities. In Fig. 3.4 (d), we provide 8 such pose-based activities. For example, *sit down* only needs the 3D poses to be distinguished, whereas the books and laptop around the subject may mislead an appearance-based classifier to recognize an activity related to those objects, such as *reading*. Secondly, TSU contains many elementary activities characterized by similar



(a) Composite (b) Elementary (c) High related Composite & Elementary

(d) Pose-based activities (e) Similar motion/activity (f) Subtle motion

Figure 3.4: On the top row, we divide the 51 activities in TSU into (a) composite and (b) elementary activities. Then, we analyze the activities along four properties: (c) highly related composite and elementary activities, (d) pose-based activities, (e) similar motion/activities, and (f) activities with subtle motion.

motions and interactions with objects. These objects provide strong clues to distinguish an activity. However, a reliable detection of the object while processing the whole video is a challenge. Sometimes, the objects are occluded within the hands of the subject, like in the case of *grasping a cup while drinking*. As a result, these activities with similar motion are often miss-classified amongst each other. In Fig. 3.4 (e), we provide 22 such activities. For example, the subjects performing *use fridge* and *use cupboard* have very similar poses. A fine understanding of the object information (e.g. fridge and cupboard) may facilitate the recognition of these activities. Finally, the dataset contains fine-grained activities characterized by subtle motions, which presents additional challenges for the recognition task. In Fig. 3.4 (f), we describe 7 such activities. For example, subjects who perform the activity *Stir coffee/tea* move only slightly their wrist and forearm. Compared to activities with pronounced motions, such as *sitting down*, learning discriminative representations for these activities with subtle motions is very challenging.

We further analyze the distribution of the activities in TSU in Fig. 3.5. We first provide a pictorial representation of the apartment along with the camera placements. TSU fea-

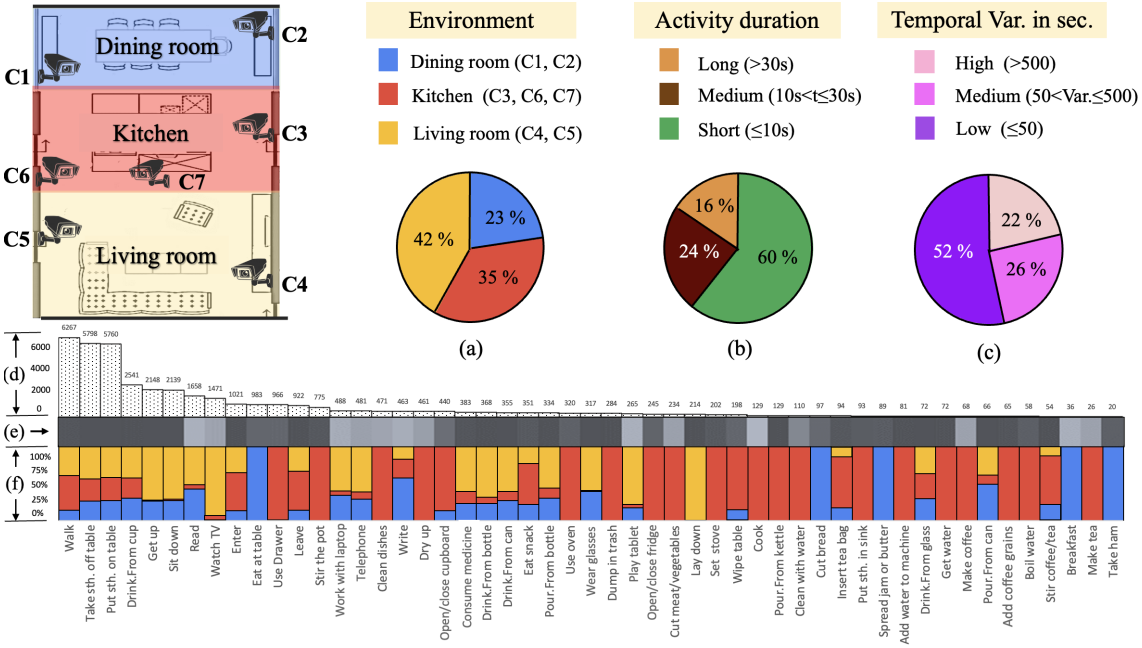


Figure 3.5: On `top` row (from left to right): we provide the 7 camera locations (`C`: camera); activity distribution along the different (a) environments, (b) duration and (c) temporal variance. Remark: (a) is per activity instance, (b),(c) are per activity class. On `bottom` row: we provide the (d) instance frequency and corresponding (e) temporal variance heat map (e.g. the lighter the larger variance), (f) distribution of performing environment for each activity.

tures multi-view settings, as all the activities are captured by more than one camera. Then, we provide 6 statistics pertaining to the activity distribution in the dataset. Fig. 3.5 (a) depicts a distribution of activity instances across the different rooms. Most activities occur in the living room, then kitchen and dinning room. This is similar to real life distribution as we spend most of our time in the living room. Correspondingly, Fig. 3.5 (f) presents the distribution of environment for each activity. We find that 51% of the activities are environment independent. For instance, we can *eat snack* or *work with laptop* in all these three environments. However, activities that rely on specific equipment occur in the same environment, such as *using oven* in the kitchen. Fig. 3.5 (b) shows the activity distribution across the activity duration. We find that in TSU, most activities are short activities, followed by medium and long activities. This is because long activities have few occurrences but longer duration. Interestingly, short activities are often more challenging to detect compared to the longer ones [137]. Fig. 3.5 (c) shows the distribution of activities based on their intra-class temporal variance. We notice that 22% of the activities have high temporal variance (i.e. vary more than 500 sec.). Correspondingly, Fig. 3.5 (e) provides the heat map of the temporal variance of these activities. The lighter grey means that the tem-

Figure 3.6: Spatial Distribution of the person location in normalized image coordinates for 3 datasets, dark regions correspond to high frequency areas of the person position. The green bounding boxes embrace the high frequency locations. From the size of the bounding box, we find that TSU exhibits the largest spatial scatter, indicating the low camera framing property.

poral variance is higher. Such intra-class variance within the same activity class further complicates the task of detection. Finally, Fig. 3.5 (d) provides the occurring frequency for every activity in the dataset. We have a non-uniform distribution of activities following the Zipf's law [36]. This long-tail distribution characterizes the real-world scenarios [38].

In addition, we leverage the spatial distribution of the person location to illustrate the camera framing property. We use the key-joint locations of Poses to compute the coordinates of the human position. Fig. 3.6 shows the spatial distribution of the person center location in different views. Compared to other similar datasets, TSU exhibits a significantly larger spatial scatter for all camera views. In most cases, the subjects move along the edge of the camera coverage area. Consequently, we consider TSU to have relatively low camera framing.

3.3.3.3 Challenges

TSU provides the 7 real-world challenges which are discussed in Section 3.2. (1) **Spontaneous behaviour**: TSU is an untrimmed ADL dataset where people are recorded while performing activities in a spontaneous manner. This property defines the uniqueness of TSU dataset. (2) **Low camera framing**: because of the long duration of the recording, the subjects do not pay attention to the fixed cameras. Therefore, activities can be performed very far, very close or out of view of the camera. Activities can also be partially occluded by furniture. (3) **Object-based activities**: The annotations in TSU include the fine-grained details of activities performed using different objects (e.g. *drinking from a cup, can or bottle*). TSU contains 7 object-based activities. (4) **Multi-views**: TSU fea-

tures 7 camera views. As shown in Fig. 3.5, the camera placement enables 2-3 camera views for each environment. In this work, we use these different views for increasing the view diversities in order to design view-invariant methods. (5) **Composite activities**: TSU contains 5 composite activity classes and 16 related elementary activity classes. (6) **Concurrent activities & dense annotation**: TSU contains up to 4 concurrent activities for a single frame. About 10% of the frames contains more than one activity label. On an average, there are about 76 activity instances per video. (7) **High temporal variance**: This new dataset offers a large variation of activity duration and intra-class temporal variance. TSU features short activities (e.g. *taking on glasses*), long activities (e.g. *reading book*), and instances of the same class that can be long or short (e.g. *writing* ranges from 3 seconds to 10 minutes). As a result, handling temporal information is critical to achieve good detection performance on TSU.

3.3.3.4 Toyota Smarthome Trimmed Vs Untrimmed dataset

The Toyota Smarthome Trimmed dataset contains only a single activity instance per video. In contrast, TSU dataset is composed of untrimmed videos and these videos are intermixed with multiple activity instances and backgrounds. The complexity of the problem is increased by the presence of **concurrent activities** and **composite activities**. Learning the dependencies across such activity instances is an important prospect for video understanding which was not considered in the previous trimmed version of Smarthome. Both the trimmed version and TSU feature **spontaneous behaviours**. As untrimmed videos contain multiple activities, the degree of spontaneity is also enhanced by the dependencies among the activities. For example, with spontaneous behaviour, the order of the elementary activities in composite activities can vary largely in untrimmed videos. For **intra-class temporal variance**, activity recognition methods on trimmed videos can handle this issue easily by sampling a fixed number of frames from different videos. However, in untrimmed videos where the task involves predicting the activity occurring at each timestamp, sampling mechanisms could lead to imprecise detection of activity boundaries. Thus, learning an activity classifier for untrimmed videos which is robust to intra-class temporal invariance is a real-world challenge and is often ignored in trimmed scenarios. Concerning **data size**, as shown in Table 3.2, TSU is 1.6 times larger in terms of activity classes compared to the previous version of the dataset, 2.8 times larger in terms of activity instances, and 3.5 times larger in terms of total number of frames.

3.3.4 Benchmark Evaluation

In TSU, we define 2 evaluation protocols: *Cross-Subject* and *Cross-View*. We provide also two evaluation metrics (frame-based and event-based mAP). For frame-based evaluation,

Table 3.2: Comparison between the two versions of Toyota Smarthome.

Dataset Version	Smarthome Trimmed [50]	Smarthome Untrimmed
Task	Recognition	Localization
#Classes	31	**51**
#Instances	16 K	**41 K**
#Frames	3.9 M	**13.8 M**

we adapt the protocol of [114] to evaluate the same mAP metric on single frames. This way of evaluating detection is robust to annotation ambiguity. For event-based evaluation, we adapt the protocol of [64]. This metric enables us to get a better insight into activity detection as not biased by activity duration.

Cross-Subject (CS): For cross-subject evaluation, we split the 18 subjects into training and test sets. To balance the number of videos for each activity category, we use 11 subjects for training and the 7 remaining ones for testing. This protocol considers all the 51 activities.

Cross-View (CV): For cross-view evaluation, the training set contains the videos from cameras 1, 3, 4, 6, 7. The remaining cameras (2, 5) are reserved for testing. The training set contains all the 51 activities and the testing set contains 32 activities from these two camera views.

3.4 Experiments

The goal of these experiments is to verify that the TSU dataset provides the novel challenges that are not yet addressed by the state-of-the-art algorithms. We evaluate 9 popular methods on TSU dataset, which represent the state-of-the-art on other densely-annotated datasets [2, 13]. Note that we have also proposed a multi-modal baseline method along with the dataset. This multi-modal method is introduced in section 6.3 with more analysis on TSU.

3.4.1 Implementation Details

3.4.1.1 Video Encoding

We use three types of encoders to extract the encoding of the input videos. For AGCN [67] and I3D [22] (pre-trained on Kinetics [41]), we fine-tune them on TSU and then the features are extracted. Besides, we also evaluate this dataset on frame-level feature. We use Inception V1 [73] pre-trained on ImageNet [138] to extract the features. The channel size of I3D and Inception V1 is 1024, channel size of AGCN is 256.

3.4.1.2 State-of-the-Art Methods

Nine activity detection methods are evaluated on our dataset, namely, bottleneck, Non-local network [28], LSTM [26], Bidirectional-LSTM [139], Dilated-TCN [5], R-I3D [23], Super-event [17], TGM [18] and MS-TCN [19]. The method using Bottleneck has only one dropout layer (with dropout probability 0.5) followed by a bottleneck layer as the classifier. Non-local [28] has one non-local block applied on the features of the whole video before the classifier. LSTM [140] has one LSTM layer with 512 hidden units and one dropout layer (with dropout probability 0.5). Similarly, for Bidirectional-LSTM [139], we have two opposite direction 512 hidden units LSTM layers. The features are concatenated before the classifier. R-I3D [81] uses I3D [22] as its SD-TCN. We set the anchor scale value to [0.3, 0.6, 1.0, 1.5, 2, 2.5, 2.75, 3, 3.5, 4, 4.5, 5,5.5, 6, 6.5, 7, 7.5, 8, 10, 12, 14, 16, 18, 20, 24, 28, 32, 38, 42, 50, 58, 66, 78, 84, 90, 96]. For TGM [18], we add one layer to have a 4-layer structure. All the methods use the same video encoding and they are trained with binary cross-entropy loss with sigmoid activation [108]. The unspecified parameters are similar to the original papers.

3.4.2 Comparative Study on TSU

	CS	CV
AGCN+Bottleneck [67]	10.1	12.6
AGCN+LSTM [140]	17.0	14.8
Inception+Bottleneck [73]	11.5	5.2
Inception+LSTM [140]	13.2	5.3
R-I3D [81]	8.7	-
I3D (Trimmed)+Bottleneck [22]	7.4	4.3
I3D+Bottleneck [22]	15.7	9.2
I3D+Non-local block [28]	16.8	9.6
I3D+Super event [17]	17.2	10.9
I3D+LSTM [141]	22.6	12.9
I3D+Bidirectional-LSTM [139]	24.5	15.1
I3D+Dilated-TCN [5]	25.1	13.9
I3D+MS-TCN [19]	25.9	13.1
I3D+TGM [18]	26.7	13.4

Table 3.3: Frame-level mAP on TSU dataset.

Table 3.3 provides the results of the considered activity detection methods on TSU. Here, we focus on the comparison of performance of the representative baselines on TSU. The comparative study is conducted with the I3D RGB features. The first method is a proposal-based method that adopts R-C3D [81] with I3D base network (we call this method R-I3D). This method fails to generate precise proposals for long activities with dense labels due to high computational cost. Consequently, it yields the worst detec-

	CS			CV		
IoU Threshold (θ)	0.3	0.5	0.7	0.3	0.5	0.7
Bottleneck [22]	5.0	2.5	0.5	2.3	1.1	0.2
Non-local block [28]	4.9	2.2	0.6	1.6	0.7	0.1
Super event [17]	5.7	2.8	0.7	1.8	0.9	0.1
LSTM [140]	11.6	6.4	2.2	6.0	3.2	0.7
Bidirectional-LSTM [139]	13.3	7.9	3.5	9.0	5.4	1.2
Dilated-TCN [5]	12.8	6.9	3.0	5.8	3.3	0.8
MS-TCN [19]	13.2	7.6	3.0	5.3	3.1	0.4
TGM [18]	15.1	9.4	4.2	5.5	3.2	0.4

Table 3.4: Event-based mAP (%) for different IoU thresholds for the TSU dataset. Note that, the input are I3D feature from RGB stream.

tion performance on TSU. The second and the third methods are the Bottleneck [22] and the Non-local block [28]. We find that the non-local block can provide the information of one-to-one temporal dependency to the local features (+ 0.9% w.r.t. Bottleneck on TSU-CS), however, Non-local block is not effective enough. Similarly, Super-event [17] utilizes temporal structure filters to model latent representation of composite activities and then compute their affinity with each frames (+4.2% w.r.t. Bottleneck on TSU-CS). However, videos in TSU are long and complex, thus it is hard to model latent representation of composite activities in this dataset. We need the temporal filter to gradually embed the information of the local frames to the current frame. LSTM [140] and Bidirectional-LSTM [139] are RNN based methods. These methods can model short temporal relations (up to +8.8% w.r.t. Bottleneck on TSU-CS), but fail to model the long temporal relationships in the complex activities of TSU. Dilated-TCN [5], TGM [18], MS-TCN [19] use temporal Gaussian/Convolutional filters which better capture the temporal relationships in long activities (up to +13.5% w.r.t. Bottleneck on TSU-CS). Thanks to the effective temporal filters, these methods can process long-term temporal relations.

We then show that the method trained on the trimmed version (i.e. I3D(Trimmed)+Bottleneck) fails to generalize to the untrimmed version. Firstly, we train an I3D model with the trimmed version of TSU (51 class version). Secondly, we leverage a sliding window framework to utilize the I3D model to predict the action class for each window, in which the classifier is fine-tuned for the frame-level action detection task. Note that I3D (Trimmed)+Bottleneck is very close to the I3D+Bottleneck model. The difference mainly lies in the I3D training process. For this baseline I3D (Trimmed)+Bottleneck, I3D is trained with the clipped action instances, whereas I3D + Bottleneck is trained with random snippets that may include action instances or even a mix of actions and background. From Tab. 3.3, we find that this baseline trained on the trimmed version under-performs in detecting actions in TSU. This is due to the lack of contextual relationships present among the action instances in the trimmed version and hence the baseline fails to generalize over the untrimmed scenario.

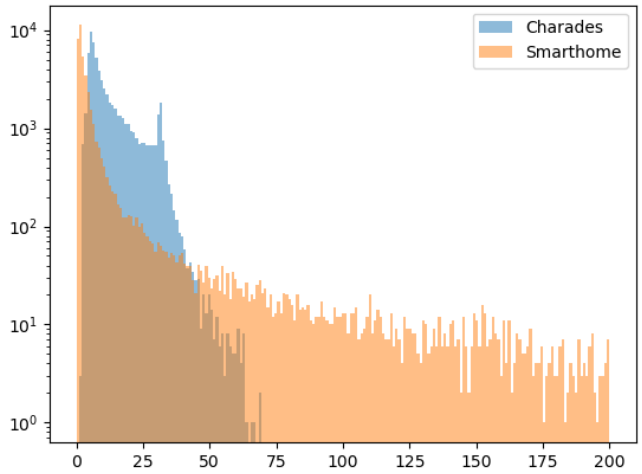


Figure 3.7: Histogram of activity instance duration in Smarthome and Charades. X axis represents the duration in seconds, Y axis represent the number of instances in log scale.

In table 3.4, we present the event-based evaluation of the baselines. The overall low performance indicates that current methods are far from addressing real-world situations.

3.4.3 Comparative Analysis between TSU & Charades

The results of the activity detection methods on different datasets provide us valuable insights into the key properties of the datasets themselves. Closely related to TSU, we choose the Charades dataset to perform a comparative study. Both datasets focus on daily living activities. They are densely annotated containing many concurrent activities and object-based activities. However, these datasets differ on several points. (1) In Charades, due to the self-recorded video settings, the activities are fast and the camera framing is high, and as a consequence, the subject is always in the center of the camera view. In contrast, in TSU, the subjects performing the activities have high spontaneity leading to higher intra-class variability and lower camera framing. (2) In Charades, the larger number of activity classes originates from the combination of only 33 verbs with different objects (e.g. holding some food, holding a sandwich). In comparison, the 51 activities in TSU originate from 35 different semantic verbs. Therefore, the Charades dataset has more activity classes relative to objects while having less semantic verbs of daily living activities. (3) TSU has longer videos (20 mins on average), compared to the on average 30 second clips in Charades. As a result, Charades does not have long activities, and the temporal variance of activity instances is low in this dataset. Fig. 3.7 presents the temporal duration of activity instances in Charades and Smarthome. We find that Smarthome has

	TSU-CS		Charades	
	Human	Center	Human	Center
I3D + Bottleneck [22]	**15.7**	10.8	15.8	15.6
I3D + Super event [17]	**17.2**	12.1	18.4	18.6

Table 3.5: Address the camera framing challenge

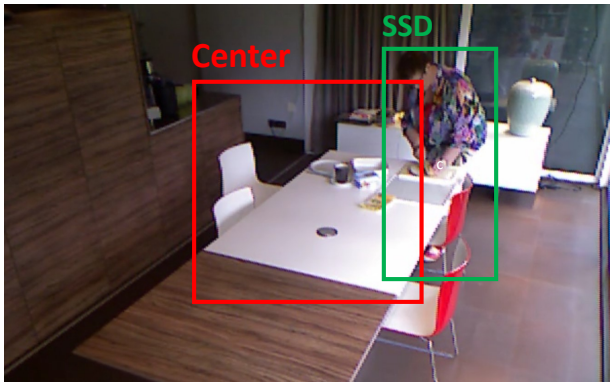


Figure 3.8: SSD & Center crops

larger scope and higher temporal variance for the activity duration.

To quantify the level of camera framing in TSU as compared to Charades, we evaluate three baseline methods trained/tested using crops around the human body or crops in the middle of the images (Fig. 3.8). The crops around the human body are extracted using SSD [142]. The results are reported in Table 3.5. To evaluate the performance on Charades, we measure the frame-based mAP for activity detection [114, 17]. For Charades, the methods using human crops and center crops obtain similar results, suggesting that Charades has high camera framing—that is, the subject in the videos is usually centered within the frames. On the other hand, in TSU, the use of human crops improves performance significantly (+5.1%). Indeed, TSU has low camera framing—that is, subjects often perform activities at the image borders.

3.5 Conclusion

In this chapter, we introduce a novel untrimmed benchmark: Toyota Smarthome Untrimmed (TSU) that features spontaneous behaviors and several real-world challenges for activity detection. This dataset contains hundreds of hours of videos of elderlies' daily life recorded in indoor smart home scenarios. Our comparative study shows that the activity detection performance for the SoTA methods on TSU is still low, highlighting the remaining open issues related to real-world conditions. Currently, TSU dataset is licensed

for academic research purposes[1]. This will allow researchers to develop novel action understanding approaches for smart home scenarios.

However, the TSU dataset still remains some limitations, such as the lack of generality to new locations and annotation bias of the manual annotations. We will refine the annotation quality and enrich the environmental diversity in the future.

Along with the dataset, we have also proposed a multi-modal baseline method: Attention Guided Network (AGNet). This network leverages multiple modalities provide in TSU. More experimental analysis of TSU, especially the analysis of modalities, is given in section 6.3.

[1]TSU Dataset is available at: https://project.inria.fr/toyotasmarthome

Chapter 4

Temporal Relational Reasoning for Action Detection

Temporal relational reasoning – the ability to link meaningful transformations of objects or entities over time – is a fundamental property of intelligent species. In this chapter, we introduce three neural networks for temporal reasoning in untrimmed videos. All three networks enhance temporal modelling by self-attention mechanism. Thanks to the different temporal modelling and self-attention strategies, the three proposed networks focus on different challenges in temporal modelling. The works in this chapter have been published in IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) 2019 [44], IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2021 [45] and IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022 [46].

4.1 Introduction

The capacity to reason about the relations between entities over time is crucial for intelligent decision-making, such as action recognition and detection tasks. A single action can consist of several temporal relations at both short-term and long-term timescales. For example, the composite action "*making sandwich*" contains the sub-actions with long-term temporal relations of *cutting bread*, *spreading the butter* and *putting bread together*. Short-term temporal relations are also needed to capture the correlations between different states of the sub-actions in a video sequence. Temporal relational reasoning allows the model to analyze the current situation relatively to the past and to formulate hypotheses on what may happen next. The detection decision at each frame should be done considering both short-term and longer-term temporal structures. This is critical especially when processing data with multiple actions occurring concurrently over different time spans.

More specifically, in the current action detection framework, the long-term videos are encoded into snippet-level features. Those snippet-level features are fed into the temporal module and then into a classifier for the action detection task. Because a snippet is often shorter than the action instance and action instances are usually highly relevant in an untrimmed video. Therefore, the action detection models rely on the temporal module that explores the contextual information of snippet sequences. With such temporal reasoning across snippets, action detection models can learn discriminative action representation and then recognize the action label for each snippet-level feature in the video. To this end, in this work, we focus on how to effectively perform temporal reasoning across the snippet-level features.

Following the recent advances of Recurrent Neural Networks (RNNs) in processing sequence data, numerous approaches are using RNN-based model to model temporal relations for action detection [143, 144, 96, 13]. Memory cells help RNNs capture temporal information from video sequences [26], while forgetting cells drop information that is irrelevant for the long-term encoding. Therefore, RNNs can only capture a limited amount of temporal context in videos, which is not suitable to process long-term data.

Temporal Convolutional Networks (TCNs) utilize one-dimensional convolutions and are another way to compute features encoded across time. Contrary to RNN-based methods, TCN computations are performed layer-wise: that means that at every time-step the network weights are updated simultaneously, which allows TCN to process long-term sequences. In this work, we choose Temporal Convolutional Network as the base temporal network.

There have been already several applications of TCN in action detection [5, 100, 19]. However, recent studies focus on short-term action datasets as [84, 145], where the mean action duration is less than 30 seconds. This cannot be straightforwardly generalized to ADL datasets, where actions can last dozens of minutes [110]. Because of the limited receptive field of CNN kernels, TCNs still have limitations when dealing with dependencies between long-range patterns in videos. As a result, we firstly introduce **Self-Attention - Temporal Convolutional Network (SA-TCN)** for modelling long-term temporal relations. SA-TCN is a TCN-based model embedded with a temporal self-attention block. This network features an encoder-decoder architecture where the temporal information is abstracted by the temporal convolutions and an attention block extracts a global temporal attention mask from the hidden representation laying between encoder and decoder. Thanks to TCN structure and self-attention block, our proposed attention mechanism can better focus on long temporal patterns and their dependencies. In this work, we used DAHLIA [110] as the main dataset to evaluate our proposed method, along with a medium-term dataset, Breakfast [83], to show the robustness of the framework. Our proposed method achieves state-of-the-art performance on both datasets.

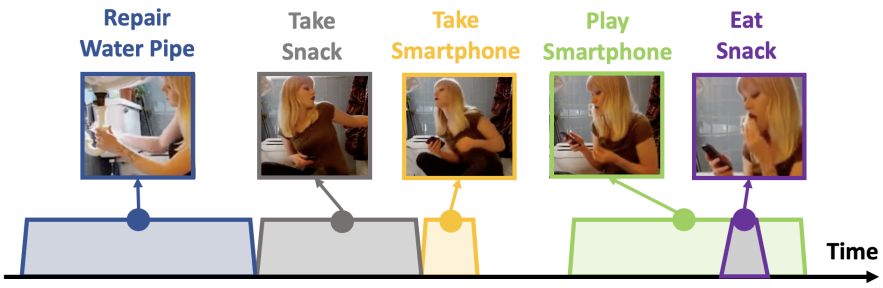


Figure 4.1: An example of complex temporal relation in a video. The actions occur densely in a video. The point indicates the center of the action instance. We provide the sampled image for each action center.

Besides the long-term temporal dependencies, there are many other challenges related to complex temporal relation in untrimmed videos (see Figure 4.1), including: (i) manage concurrent actions occurring at the same time. For example, *eating snack* while *playing smartphone*, and (ii) model both long-term and short-term dependencies in the video. For example, short-term dependencies from action *'playing smartphone'* and long-term dependencies from action *'taking snack'* can both provide contextual information to detect the action *'eating snack'*.

To handle the challenges of complex temporal relation, we firstly propose the **Pyramid Dilated Attention Network (PDAN)**, which is composed of a series of Dilated Attention Layers (DAL). The main novelty of this architecture is how the attention weights are allocated to local frames at multi-temporal scales. A standard temporal convolution layer features shareable kernels which allocate the same importance to local frames in the kernel. This property prevents the temporal convolutional kernels from selecting the key information. This is a limitation especially when large temporal receptive fields are required for modeling long untrimmed videos. To overcome this limitation, we build a novel attention mechanism to explore the local context inside the kernel. The kernel ultimately processes the entire video, but at each time step the inputs are only those frames comprised in the kernel (i.e., a small window). DAL explores the relations between the center frame and the neighbouring frames in the kernel (called local context). This local attention mechanism enables the proposed framework to learn representations for short actions. Additionally, by introducing dilation in the aforementioned temporal attentional operations, we build a Pyramid Dilated Attention Network (PDAN) which consists of a hierarchy of DALs. These DALs are configured with different dilation rates to increase exponentially the size of the filter receptive field. This hierarchical structure allows PDAN to allocate attention weights to different temporal resolutions using the different DAL layers. This structure design is instrumental for the action detection of densely annotated videos. We evaluate PDAN using three densely annotated action detection datasets: Charades [2], MultiTHUMOS [13],

and our TSU. PDAN achieves competitive state-of-the-art performance on all the datasets.

PDAN is an effective manner for modelling complex temporal dependencies in a video. However, convolution-based methods as PDAN are limited by their kernel size and can only directly access to local information in a video. Thus, such methods fail to model long-range interactions between segments (i.e., snippets) which may be important for action detection. With the success of Transformers [107, 94, 146, 25] in natural language processing and more recently in computer vision, recent methods [7, 93] have leveraged multi-head self-attention (MHSA) to model long-term relations in videos for action detection. Such attention mechanisms can build direct one-to-one global relationships between temporal segments (i.e., temporal token) of a video to detect highly-correlated and composite actions. However, existing methods rely on modeling such long-term relationships on input frames themselves. In this case, a temporal token covers only a few frames, which is often too short w.r.t. the duration of action instances. Moreover, transformers need to explicitly learn strong relationships between adjacent tokens which arise due to temporal consistency, whereas it comes naturally for temporal convolutions (i.e., local inductive bias). Therefore, a pure transformer architecture may not be sufficient to model complex temporal dependencies for action detection.

By rethinking the manner of combining convolutions and self-attention, we propose ***Multi-Scale Temporal ConvTransformer (MS-TCT)***. In this network, we use convolutions in a token-based architecture to promote multiple temporal scales of tokens, and to blend neighbouring tokens imposing a temporal consistency with ease. In fact, MS-TCT is built on top of temporal snippets encoded using a 3D convolutional backbone [22]. Each temporal snippet is considered as a single input token to MS-TCT, to be processed in multiple stages with different temporal scales. These scales are determined by the size of the temporal segment (i.e., snippet), which is considered as a single token at the input of each stage. Having different scales allows MS-TCT to learn both fine-grained relations between atomic actions (e.g. *'open fridge'*) in the early stages, and coarse relations between composite actions (e.g. *'cooking'*) in the latter stages. To be more specific, each stage consists of a temporal convolution layer for merging tokens, followed by a set of multi-head self-attention layers and temporal convolution layers, which model global temporal relations and infuse local information among tokens, respectively. As convolution introduces an inductive bias [147], the use of temporal convolution layers in MS-TCT can infuse positional information related to tokens [148, 149], even without having any positional embeddings, unlike pure transformers [146]. Followed by the modeling of temporal relations at different scales, a mixer module is used to fuse the features from each stage to get a unified feature representation. Finally, to predict densely-distributed actions, we introduce a heat-map branch in MS-TCT in addition to the usual multi-label classification branch. This heat-map encourages the network to predict the relative temporal position

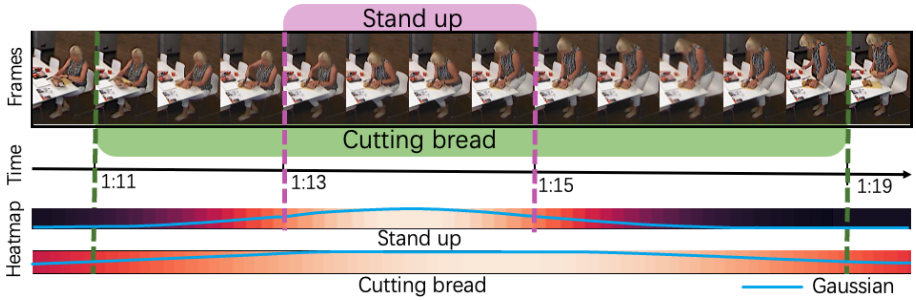


Figure 4.2: **Relative temporal position heat-map ($\mathbf{G}^*$):** We present a video clip which contains two overlapping action instances. The *Gaussians* indicate the intensities of temporal heat-maps, which are centered at the mid point of each action in time.

of instances of each action class. Fig. 4.2 shows the relative temporal positions, which are computed based on a Gaussian filter parameterized by the instance center and its duration. It represents the relative temporal position w.r.t. the action instance center at any given time. With this new branch, MS-TCT can embed a class-wise relative temporal position in token representations, encouraging discriminative token classification in complex videos.

To summarize, our contributions in this chapter: (i) We introduce SA-TCN, which leverages encoder-decoder architecture to abstract the salient temporal information of the video and utilizes the self-attention mechanism to model dependencies across time for long videos. (ii) We design PDAN, which can effectively learn the dependencies between action instances by applying DAL at different temporal scales. The DAL inside PDAN can improve the quality of the local feature representation across time. (iii) We propose MS-TCT, which is an effective and efficient ConvTransformer for modeling complex temporal relations in untrimmed videos. Moreover, we introduce a new branch to learn the position relative to instance-center, which promotes action detection in densely-labelled videos.

In the following sections, we first revise the related work in temporal modelling and attention mechanism. After that, we introduce the three model structures and experiment for each method.

4.2 Related Work

In this section, we review how previous works learn temporal relations and utilize attention for action detection.

4.2.1 Temporal Modelling

After encoding the video, action detection can be seen as a sequence-to-sequence problem. Inspired by the advancement in natural language processing, there are three principle branches for sequential modelling in recent years: Recurrent Neural Networks, Temporal Convolution Networks, Transformers. We will revise these techniques below.

Recurrent Neural Networks (RNNs) [13, 98, 99] have been popularly used to model the temporal relation between the action instances. Singh et al. [144] feed per-frame CNN features into a bi-directional long short-term memory network (LSTM) model and apply non-maximal suppression to the LSTM output. MultiLSTM [13] extends the vanilla LSTM for handling videos with dense action regions. This method expands the temporal receptive fields at both input and output to be a window length of frames. Moreover, a soft-attention weighting is learned over the input window to select the action related frames. Huang et al. propose Although the above methods is LSTM methods only implicitly capture relationships between certain actions with high motion. Furthermore, due to the vanishing gradient problem, RNN based models can only capture a limited amount of temporal information and short-term dependencies.

Temporal Convolutional Networks (TCNs) are another group of temporal processing methods. In contrast to RNN based methods, TCNs can process long videos due to the kernels sharing weight for all the time steps. The result is a feature vector preserving the spatio-temporal information, along with contextual information from the neighboring frames. Some recent variants of TCNs for action detection include ED-TCN, Dilated TCN [5] and MS-TCN [19]: Lea et al. [5] design two temporal convolutional networks for action segmentation and detection task, transforming successful approaches from natural language processing. ED-TCN uses pooling and up-sampling to efficiently capture long-range temporal patterns whereas Dilated-TCN increases the temporal reception field by using dilated convolutions to model long temporal patterns. Dilated-TCN is extended by MS-TCN [19] which stacks multiple Dilated-TCNs to construct a multi-stage structure, where each stage refines the prediction of the previous one. In addition, Temporal Aggregation Network (TAN) [102] consists of dedicated temporal aggregation blocks, which is designed to encode multi-scale spatio-temporal patterns, and larger temporal context can be captured by dilated convolutions effectively. However, standard convolutions allocate the same importance to each local feature in the kernel. This property prevents temporal convolution kernels from extracting the key information efficiently from long complex untrimmed videos.

With the introduction of datasets like MultiTHUMOS [13] and Charades [2] having dense labelling and concurrent actions (i.e. multi-label), more and more methodological attempts to model complex temporal relations between action instances have been made.

There are also some temporal operations in the community that can be seen as variants to temporal convolution. Piergiovanni et al. proposed a global representation, namely super-event [17]. In this model, Cauchy distribution based filters process the video across time to learn a latent contextual representation of the actions on particular sub-intervals of the video. The set of filters are summed by a soft attention mechanism to form the global super-event features. During prediction, the local I3D features are used with the super-event features to better model the global context. Similarly, Piergiovanni et al. [18] introduced Temporal Gaussian Mixture (TGM) layers. In contrast to standard convolution layer, TGM computes the filter weights based on Gaussian distributions, which enables TGM to learn longer temporal structures with a limited number of parameters. Although the above methods [17, 18] achieve state-of-the-art results in modeling complex temporal relations, the non-adaptive receptive field limits the ability of the models to capture the dynamics for both short and long patterns.

4.2.2 Self-Attention Mechanisms

Self-attention mechanisms focus on the salient part of a scene relative to a target task, which was proposed by Transformer Networks [107] for natural language processing. This operation enforces a network to establish one-to-one relations to understand the dependencies between their local representations. Employing self-attention mechanisms has gained popularity for different downstream tasks: Ramachandran et al. [150] proposed “fully attentional network", which achieves competitive prediction results on image classification tasks. This model replaces the standard 2D convolution layer with local attention layer in ResNet [72]. This layer learns the representation based on the relative position of the spatial features in the kernel. Similar to [107], Girdhar et al. [58] proposed the Action Transformer model for the task of action detection. This model inherits the transformer-style architecture to modulate features with attention weights from the spatio-temporal context within a video. This attention mechanism emphasizes the region-of-interest (e.g. actors’ hands, faces), which are often crucial to recognize an action. However, Action Transformer is embedded in I3D [22] as the base network, which restricts its input size to only short video clips (i.e. 64 frames). Our target is to detect both long and short actions in a long video, far beyond 64 frames. Thus, we need a better attention mechanism that is dedicated to model temporal relations. Wang et al. [28] designed a Non-Local (NL) layer that achieves SOTA performance in action recognition task. This block leverages the self-attention mechanism to learn an attention map representing the spatial-temporal one-to-one dependencies of the 3D features. Extending NL layer, Cao et al. [151] introduced Global Context (GC) layer, which has same performance as the NL layer but with fewer parameters. While adapting NL layer and GC layer for action

detection task, the receptive field of the layer is always the full video. The fixed global receptive field introduces more noise of the irrelevant actions in the attention map, thus can not provide effective attention information especially for the videos that concurrently have both multiple long and short actions.

4.2.3 Video Transformer

Recently, with the advent of Vision Transformer [146], Transformer architectures have been successful in both image and video domain [94, 25, 152, 153, 154, 155, 156, 24, 157, 76, 27]. Although Vision Transformers [25, 24, 76], such as TimeSformer [76] can consider frame-level input tokens to model temporal relations, it is limited to short video clips which is insufficient to model fine-grained details in longer real-world videos. As a compromise, recent video understanding methods use multi-head self-attention layers on top of the visual segments encoded by 2D/3D convolutional backbones [22]: Video Transformer Network [71] builds on top of a given 2D convolutional network and feeds the encoded feature to the transformer encoder for temporal modelling. VidTr [158] is another Transformer model features pooling layers for attention. This operation drops non-informative features along temporal dimension thus achieve better performance in video understanding with higher efficiency. MTCN [159] is a multi-modal Transformer model, which benefits from the temporal context of action and labels to enhance the action predictions using a Transformer encoder. TQN [30] is designed for recognizing fine-grained actions. TQN factorizes categories into pre-defined attribute queries to predict fine-grained actions with a Transformer Decoder. However, all these methods are designed for action recognition and not trivial to extend them to action detection in untrimmed videos. Regarding untrimmed video: LSTR [160] employs a long- and short-term memory mechanism to model streaming data. This model consists of an encoder that obtains coarse-scale historical information, together with an LSTR decoder to model the fine-scale characteristics of the data. However, similar to the drawback of LSTM, this method only benefits from the previous time-steps of the current time, and thus is utilized for online action detection. RTD-Net [93], an extension of DETR [94], uses a transformer decoder to model the relations between the proposal and the tokens. However, this network is designed only for sparsely-annotated videos [11, 78], where only a single action exists per video. In dense action distributions, the module that detects the boundaries in RTD-Net fails to separate foreground and background regions. MLAD [7] learns class-specific features and uses a transformer encoder to model class relations at each time-step and temporal relations for each class. However, MLAD struggles with datasets that has complex labels [2], since it is hard to extract class-specific features in such videos.

In the following sections, we introduce the proposed temporal models in detail.

4.3 Self-Attention - Temporal Convolutional Network (SA-TCN)

In this section we propose our model: the Self-Attention - Temporal Convolutional Network (SA-TCN), which retains the encoder-decoder architecture of ED-TCN to capture long-range patterns and embeds a self-attention mechanism to capture the long range dependencies between those patterns. The overview of this architecture is shown in Fig. 4.3 and consists of 3 main components: visual encoding, encoder-decoder TCN, and self-attention block.

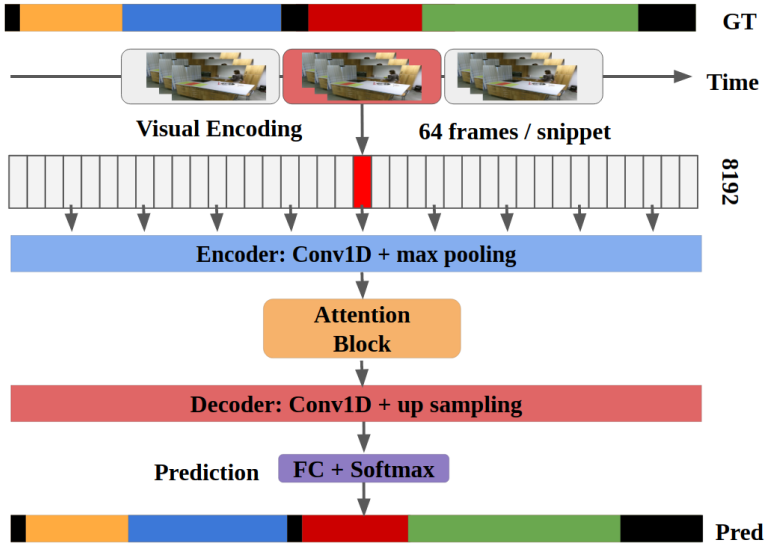


Figure 4.3: **SA-TCN model.** Given an untrimmed video, we represent each non-overlapping snippet by a visual encoding over 64 frames. This visual encoding is the input to the encoder-TCN, which is the combination of the following operations: 1D temporal convolution, batch normalization, ReLu, and max pooling. Next, we send the output of the encoder-TCN into the self-attention block to capture long-range dependencies. After that, the decoder-TCN applies the 1D convolution and up sampling to recover a feature map of the same dimension as visual encoding. Finally, the output will be sent to a fully connected layer with softmax activation to get the prediction.

4.3.1 Visual Encoding

The first step in our architecture is the extraction of a visual encoding. As opposed to the other TCN-based methods [5, 100] that use multi-modal inputs (i.e. RGB+flow), we attempted to use RGB only. To reduce the redundancy coming from extracting background features, we apply SSD [142] to detect the subjects and crop patches based on those detections. The patches are then resized to 224×224 and fed into an Imagenet pre-

trained Resnet-152. We extract features from the penultimate layer of Resnet-152. We group 64 contiguous extracted feature sets per snippet. The temporal context of the video is handled by the aggregation operator using max and min pooling across the snippets. This pooling mechanism helps to choose salient values from the feature map. The visual encoding that we obtain from this step will be the input of encoder-TCN.

4.3.2 Encoder-Decoder TCN

SA-TCN retains the encoder-decoder architecture of [5], with the addition of some points of improvement.

As shown in Fig. 4.4, we have k layers for both the encoder and the decoder. In the encoder part, each layer consists of temporal convolutions, batch normalization, ReLU activation, and a temporal max pooling. We set a fixed convolution kernel size for all the layers. First, we applied temporal convolution (Conv-1D) to extract high-level features. Second, differently from ED-TCN, we applied batch normalization to avoid vanishing or exploding gradients. Third, we added a spatial dropout layer along with a ReLU non-linearity to help controlling over-fitting and to speed up convergence. Finally, we max pool the feature map across time to halve the temporal dimension. Pooling enables us to efficiently compute activations over long temporal windows.

Our decoder is similar to the encoder, except for the fact that we replace the pooling operation with up sampling. This up sampling step is similar to [5]: each entry is repeated twice. After that, another temporal convolution is performed to reduce the aliasing effect of up sampling. Finally, a snippet-wise fully-connected layer with softmax activation is used to generate the class probabilities at each time step.

4.3.3 Self-Attention Block

In this section, we introduce our temporal self-attention block. We construct this temporal attention mechanism based on the scoring system presented in [107].

The purpose of attention block is to build a one-to-one association between all the temporal moments. We do not rely on any outside information, so it is called self-attention. To implement this, the input I is branched out into three copies $Query$, Key and $Value$. Through the calculation of similarity between $Query$ and each Key, we can get the attention score s, which is the importance of different temporal moments. This attention score is then normalized by softmax to have a mask α. Finally, we multiply the $Value$ by this mask to have the attention-weighted feature, and then, add back the input to have our output result O.

Fig. 4.5 shows a diagram of the self-attention block, where $I \varepsilon \mathbb{R}^{C \times T}$ denotes the input features from the previous hidden layer. I is first transformed into two feature spaces

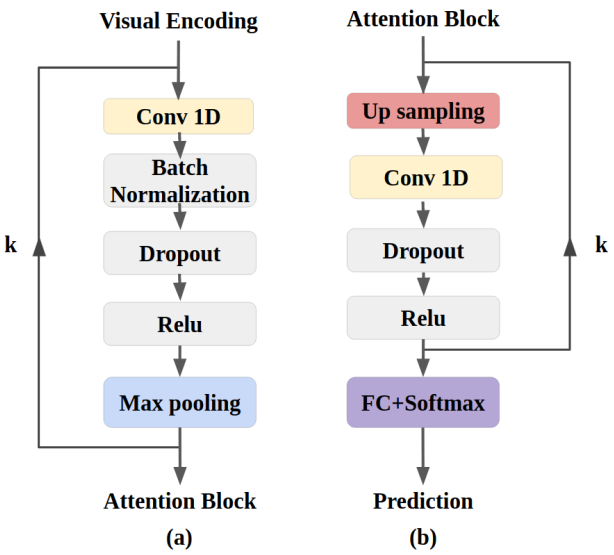


Figure 4.4: **Encoder-decoder architecture**. This figure represents the network structure of (a) encoder-TCN and (b) decoder-TCN. As the architecture has k layers, it will have k iterations.

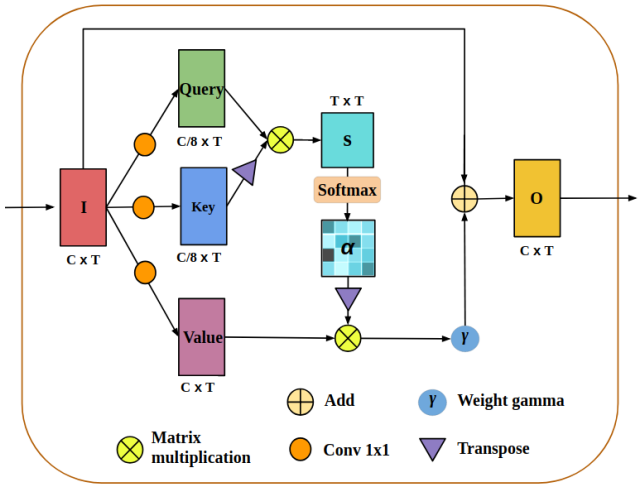


Figure 4.5: **Structure of self-attention block** between encoder-TCN and decoder-TCN.

$Query$, Key, where $Query(I) = W_{Query}I$, $Key(I) = W_{Key}I$. Both W_{Query} and W_{Key} ε $\mathbb{R}^{C\times\frac{C}{8}}$. In this work, $Value$ is computed from I with a 1×1 convolution layer. Thus we have $Value(I_i) = W_{Value}I_i$, where W_{Value} ε $\mathbb{R}^{C\times C}$. The number of filters of $Value$ is same as the channel size of I. $Query$ and Key are similar to $Value$, except for the fact that the number of filters is one-eighth of $Value$. If $\alpha_{j,i}$ indicates the extent to which the model

attends to the i^{th} location when synthesizing the j^{th} region, we have:

$$\alpha_{j,i} = \frac{exp(s_{ij})}{\sum_{i=1}^{T} exp(s_{ij})}, \tag{4.1}$$

where $s_{ij} = Query(I_j)Key(I_i)^T$.
Then the output of the weighted attention map is $Att = (Att_1, Att_2, ..., Att_j, ..., Att_T)\ \varepsilon\ \mathbb{R}^{C\times T}$, where,

$$Att_j = \sum_{i=1}^{T} \alpha_{j,i} Value(I_i) \tag{4.2}$$

Finally, we add back the input feature map to assign weight to non-local evidence. Therefore the output O_i is given by:

$$O_i = \gamma \times Att_i + I_i \tag{4.3}$$

where γ represents a learnable parameter. The output O will be fed into decoder-TCN.

4.3.4 Experiments

In this work, we performed experiments mainly on DAHLIA [110], which is a dataset contains long-term video and long-range temporal dependencies. We also evaluate on Breakfast [83] dataset to show the robustness of our method. In the following, we describe the baseline methods used in our study. We provide a comparative analysis of our method against other action detection architectures. In all experiments, frame-wise accuracy(FA_1), F-score, Intersection over Union(IoU) and mean Average Precision(mAP) [5] are reported.

4.3.4.1 Implementation Details

We implemented our model in Keras 2.0.8 with Tensorflow as back-end. The experiments were performed on a GTX 1080 Ti GPU with 11 GB memory. For the visual encoding, we performed experiments using both Resnet-152 [72] and I3D [22] as the feature extractor. With Resnet, we extracted the features as described in detail in section *3.1* leading to 8192 features per snippet. With I3D, we chose the Kinetics pre-trained I3D. First, we added a fully connected layer with 1024 units before the classification layer. Secondly, we fine-tuned the architecture on the NTU-dataset[62] and extracted features from the new fully connected layer (1024 features per snippet). We ran experiments with both Resnet-152 and I3D on DAHLIA. The results obtained with the two feature extractors are similar. On the Breakfast dataset, we use the features provided on the dataset's website. The length of these features is 64/snippet.

In our model, the attention operation does not change the dimension of the feature

map. Besides, we assign the parameters of the encoder-decoder TCN so that the size of the feature map before the first encoder layer is the same as the output of the last decoder layer: we set the pooling and up sampling rate to 2, the number of filters in the three layers to {48, 64, 96} and {96, 64, 48} for encoder and decoder respectively. Finally, we compared several kernel sizes for the 1D convolution, and found that a size of 25 for every layer gives the best results.

The training was conducted with RMSprop with a learning rate of 0.001 and batch size 8 for both DAHLIA and Breakfast datasets. On DAHLIA, we split the train and validation set with 15% validation rate. We trained the model for 100 epochs and measured detection performance on the test set.

Model	FA_1	**F-score**	**IoU**	**mAP**
DOHT [161]	0.803	0.777	0.650	-
GRU* [4]	0.759	0.484	0.428	0.654
ED-TCN* [5]	0.851	0.695	0.625	0.826
Negin *et al.* [6]	0.847	0.797	0.723	-
TCFPN* [100]	0.910	**0.799**	0.738	**0.879**
SA-TCN	**0.921**	0.788	**0.740**	0.862

Table 4.1: Action detection results on DAHLIA dataset with the average of view 1, 2 and 3. *marked methods have not been tested on DAHLIA in their original paper.

Model	FA_1	**F-Score**	**IoU**	**mAP**
GRU [4]	0.368	0.295	0.198	0.380
ED-TCN [5]	0.461	0.462	0.348	0.478
TCFPN [100]	**0.519**	0.453	0.362	0.466
SA-TCN	0.497	**0.494**	**0.385**	**0.480**

Table 4.2: Action detection results on Breakfast dataset.

Actions	**Background**	**House work**	**Working**	**Cooking**
AP	0.36	0.65	0.95	0.96
Actions	**Laying table**	**Eating**	**Clearing table**	**Wash dishes**
AP	0.90	0.97	0.80	0.97

Table 4.3: Average precision of ED-TCN on DAHLIA.

4.3.4.2 Results Analysis

In this section, we analyze the results of our method and of the other state-of-the-art baselines.

Model	FA_1	F-score	IoU	mAP
TCFPN [100]	0.910	**0.799**	0.738	0.879
SA-TCFPN	**0.917**	**0.799**	**0.748**	**0.894**

Table 4.4: Combination of attention block with other TCN-based model: TCFPN. (Evaluated on DAHLIA dataset)

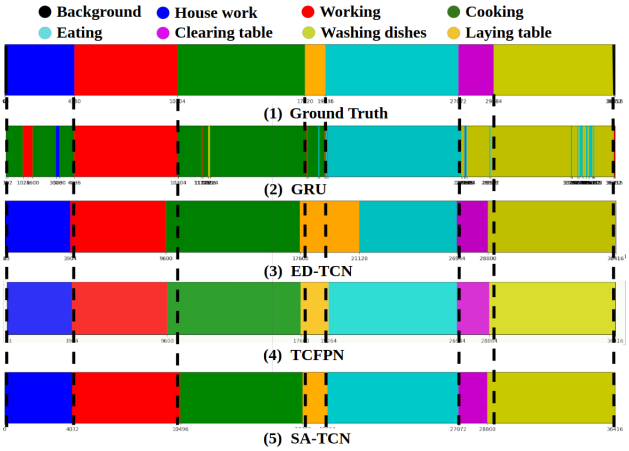


Figure 4.6: **Detection visualization**. The detection visualization of video 'S01A2K1' in DAHLIA: (1) ground truth, (2) GRU [4], (3) ED-TCN [5], (4) TCFPN [6] and (5) SA-TCN.

Table 4.1 and 4.2 show the results of all the methods considered on DAHLIA and Breakfast datasets, respectively. Our method achieves state-of-the-art performance on both datasets.

DOHT and Negin *et al.*'s method, which train a SVM with deep or hand-crafted feature encoding, do not perform well on DAHLIA. This is because approaches based on a sliding window can only capture window-size patterns. Although a post-processing step is used to filter noise, these approaches still fail at capturing long temporal information.

Compared to TCN-based networks, GRU does not perform well on DAHLIA. Fig. 4.6 shows that GRU fails at distinguishing short actions performed between long actions (i.e. laying table and clearing table). Moreover, GRU produces noise while detecting long actions due to the fact that RNN-based networks can not focus on long temporal information.

ED-TCN lacks precision in detecting the action boundaries. As CNNs have a limited receptive field for each layer, they fail in detecting the dependencies between long-distanced features. The results obtained by ED-TCN on DAHLIA are reported in Table 4.3. The low precision achieved on the 'Background' action is due to the shorter duration of this action compared to the others, which results in a lower number of training samples.

Both TCFPN and our SA-TCN outperform ED-TCN. The pyramid structure with lateral connections helps TCFPN to make use of both low-level and high-level features. The tem-

poral attention block of our SA-TCN enables a better understanding of the dependencies between the different actions performed in the video.

To understand if our solution can be integrated with other temporal models, we embedded our temporal self-attention block in TCFPN to obtain SA-TCFPN. As reported in Table 4.4, SA-TCFPN outperforms TCFPN on all the metrics on DAHLIA. This shows that our temporal attention block is general and can be effectively integrated with other temporal models.

4.4 Pyramid Dilated Attention Network (PDAN)

In this section, we introduce Pyramid Dilated Attention Network (PDAN), an end-to-end model for action detection. The main goal is to learn frame-level feature representation that encodes spatio-temporal information so that the model can effectively exploit information from multiple temporal scales to predict the actions for each frame. The basic building block in PDAN is a Dilated Attention Layer (DAL) followed by ReLU activation and a bottleneck with residual connection. Note that in this work, bottleneck indicates the 1D convolution that processes across time and kernel size is 1. Different from previous dilated-TCN layers [5, 19], DAL computes adaptable probabilistic scores for each local feature of the kernel through a self-attention mechanism. Thanks to multiple DALs with pyramid dilation setting, PDAN weights local input feature to capture their saliency at several temporal scales, which enables the model to capture meaningful temporal relationships between complex atomic actions. Our intuition for the design of PDAN is that using attention, dilation and residual connections together can capture salient segments of an action at several temporal resolutions and provide a robust representation against temporal variation of actions. An overview of the proposed PDAN is shown in Fig. 4.7. The RGB and the Flow stream have similar structure, the only difference is the input to the 3D CNN. When both modalities are available, we apply a late fusion of their prediction logits. In the following sub-sections, we elaborate our model.

4.4.1 Video Feature Extraction

Similar to most action detection models, our model can process on top of video segment representations (usually from frame-level or segment-level CNN features). In this work, we use spatio-temporal features extracted from the RGB and Flow I3D networks [22] to encode appearance and motion information respectively. To achieve this, a video is divided into T non-overlapping segments, each segment consisting of 16 frames. The inputs to the RGB and Flow deep networks are the color images and corresponding Flow frames of a segment respectively. We stack the segment-level features along temporal axis to form a

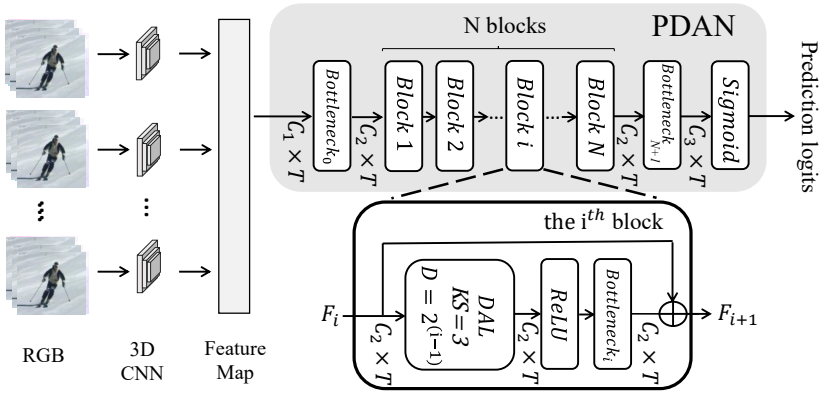


Figure 4.7: Overview of the Pyramid Dilated Attention Network (PDAN). In this figure, we present the structure of PDAN for one single stream. Note that RGB and Flow stream have same structure inside PDAN. Two streams are connected by late fusion operation before classification. DAL indicates the dilated attention layer, in which, KS is the kernel size, D is the dilation rate.

$T \times C_1$ dimensional video representation where $1 \times C_1$ is the feature shape per segment. This video representation denoted as F_0 is further input to the RGB or Flow stream in our architecture. Below, we detail the basic component of our proposed PDAN, which is DAL.

4.4.2 Dilated Attention Layer (DAL)

As earlier said, standard temporal convolution layer assigns the same importance to all the input features of the kernel. However, with multi-scale receptive fields, providing relevant attention weights can benefit modelling of complex temporal relationships. To this end, we propose DAL with multiple dilation rates that inherently learns the attention weights at different temporal scales. As most temporal filters [18, 17], DAL processes the feature maps across the temporal domain only to preserve spatial information.

As shown in Fig. 4.8, the input features are processed in two steps in each kernel of DAL. Take the i^{th} block as an example: First, the elements (i.e. segment) around a center element f_{it} at time $t \in [1, T]$ are extracted to form a representative vector f'_{it}. This feature representation is based on the kernel size: ks and dilation rate at i^{th} block. Note that: feature $f_{it} \in \mathbb{R}^{1 \times C_2}$, $f'_{it} \in \mathbb{R}^{ks \times C_2}$. Second, the self-attention scoring system [107] is invoked by projecting the representative vector f'_{it} to a memory embedding (Key: K_i and Value: V_i) using 2 independent bottleneck convolutions: $K_i(f'_{it}) = W_{K_i} f'_{it}$, $V_i(f'_{it}) = W_{V_i} f'_{it}$, both W_{K_i} and $W_{V_i} \in \mathbb{R}^{C_2 \times C_2}$. Then, f_{it} is projected to the Query Q_i using another bottleneck convolution: $Q_i(f_{it}) = W_{Q_i} f_{it}$ and $W_{Q_i} \in \mathbb{R}^{C_2 \times C_2}$. The output of the attentional operation for the t^{th} time step is generated by a weighted sum of values V_i, with the attention

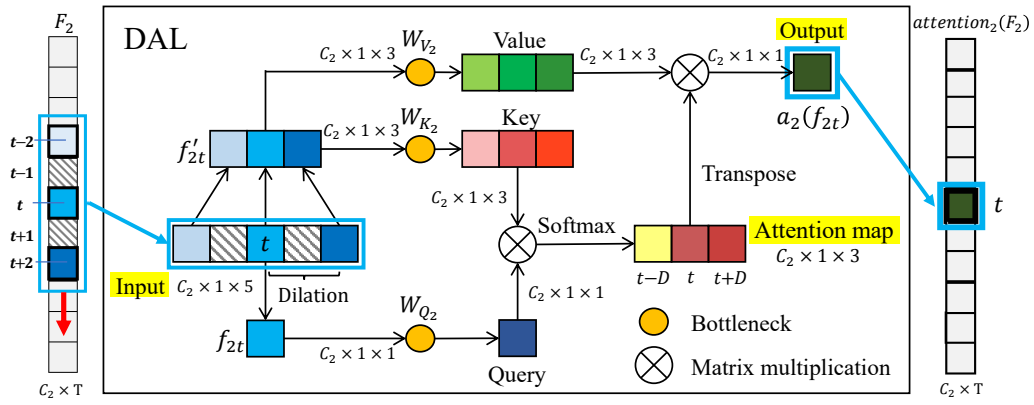


Figure 4.8: Dilated Attention Layer (DAL). In this figure, we present an example of a computation flow inside the kernel at time step t (kernel size ks is 3, dilation rate is 2). Note: in this figure, the subscript of block i should be 2.

weights obtained from the product of the query Q_i and keys K_i:

$$a_i(f_{it}) = V_i(f'_{it})[softmax(Q_i(f_{it})K_i(f'_{it}))]^{\mathrm{T}} \tag{4.4}$$

In contrast to the previous work [28] where the authors calculate one-to-one correlation between all the elements, the attention mechanism in DAL computes the correlation inside the kernel between the center element and the other local elements, which significantly reduces the number of parameters. Finally, the output of a DAL is obtained by concatenating the outputs for all the time steps t of the video.

$$attention_i(F_i) = [a_i(f_{i1})^{\mathrm{T}}, a_i(f_{i2})^{\mathrm{T}}, ..., a_i(f_{iT})^{\mathrm{T}}] \tag{4.5}$$

where F_i is the input feature map of DAL at the i^{th} block.

4.4.3 Comparison with Non-Local layer

Transformer [107] is not directly applicable to action detection. Its extension to video, Action-Transformer [58] can only process short video clips (i.e. 64 frames) and its attention mechanism is not designed to model temporal relations. Non-Local (NL) [28] has a similar structure to that of the attention head in Transformer, and is used in action detection task. Hence, we only compare DAL with the NL layer. The 1-dimensional NL layer's receptive field corresponds to the full video. These filters learn an attention map of dimension $T \times T$ reflecting the one-to-one dependency for every frame in the full video. On the other hand, DAL's receptive field at each time step t covers only the neighbouring frames in the kernel. The kernel ultimately processes the entire video, but at each time step t, the input are only those frames included in the kernel (of size KS). Thus, DAL learns an attention map of dimension $T \times KS$, i.e. it explores the relations between the center frame and its KS neighbouring frames in the kernel. Moreover, by stacking

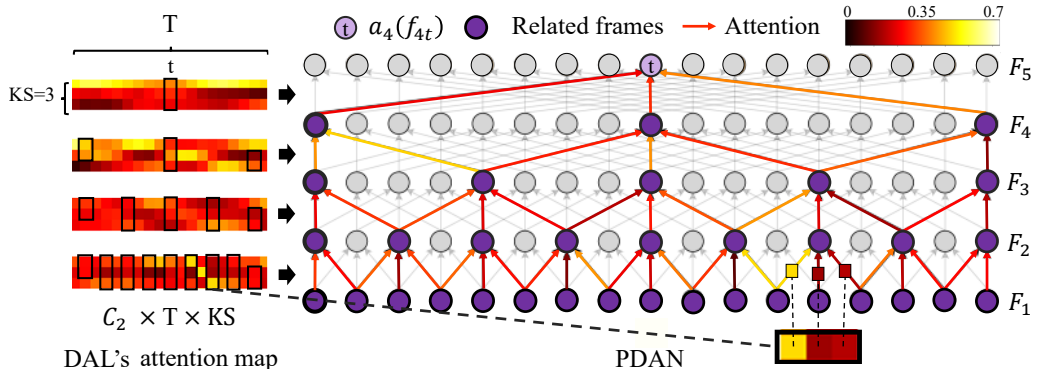


Figure 4.9: On the left, we visualize the attention map for DAL for four layers ($i \in$[1,4]). On the right, we present a group of frames at different temporal scales that are associated with $a_4(f_{4t})$ along with the corresponding attention weights. The circle represent the frame-level features (i.e. feature in F_i), and the arrow represents the attention-enhanced connection between the corresponding frames provided by DAL. The bounding box in the attention map corresponds to the colored arrow at right.

multiple layers with different dilation rates, the receptive field is expanded gradually in higher layers to model longer actions. Consequently, both DAL and NL layers explore the whole content of the video. Real-world untrimmed videos [13] have long duration, large temporal variance, and concurrent actions. While processing such videos, the fixed global receptive field of the NL layer implies that information linked to irrelevant actions happening potentially far away from the current frame will introduce noise to the representation of the current frame. In contrast, DAL reformulates the attention mechanism for detecting long and short actions in a sparse and hierarchical manner. This design enables the attention mechanism at each layer to focus on actions of different temporal lengths, thus providing better context information and filtering irrelevant information from the distant actions. Our ablation study confirms the effectiveness of DAL. In Fig 4.9, we give an example where DAL assigns different attention weights for local frames at every time step and at multi-temporal scales. The efficiency and effectiveness of NL layer and DAL is discussed in Sec. 4.4.5.3. In the following section, we describe how we use DALs at multiple-temporal scales.

4.4.4 Pyramid Structure of Temporal Layers

Applying self-attention on multi-temporal scale is an essential ingredient for modeling complex temporal relations. PDAN is based on a pyramid of DALs with same kernel size and different dilation rates. The pyramid increases exponentially the size of the receptive field of the model. This structure allows the network to model short and long action patterns by focusing on the local segments at the level of low and high temporal receptive

fields.

As shown in Fig. 4.7, the input feature $F_0 \in \mathbb{R}^{T \times C_1}$ is firstly fed to a bottleneck layer to lightweight the model by reducing the channel size from C_1 to C_2. Then, N blocks are stacked, each block i is a cascade of a DAL with ReLU activation, $bottleneck$ convolution and a residual link. This structure allows the receptive field to increase exponentially while keeping the same temporal length T as the input. In our experiment, we set the kernel size (*ks*) to 3 for all blocks, dilation and padding rate to 2^{i-1}, thus the reception field is up to $2^i + 1$ for the i^{th} block. The set of operations in each block can be formulated as follow:

$$F_{i+1} = F_i + W_i * ReLU(attention_i(F_i)) \tag{4.6}$$

where F_i indicates the input feature map of the i^{th} block. In the attention layer $attention_i$ the dilation rate varies with i. $W_i \in \mathbb{R}^{C_2 \times C_2}$ indicates the weights of the $bottleneck_i$. Finally, we compute per-frame binary classification score for each class (i.e. prediction logits). Therefore, the N^{th} block is followed by a bottleneck convolution with $sigmoid$ activation:

$$P = sigmoid(W_{B_{N+1}} F_{N+1}) \tag{4.7}$$

where $P \in \mathbb{R}^{T \times C_3}$ is the prediction logits and $W_{B_{N+1}} \in \mathbb{R}^{C_3 \times C_2}$, C_3 corresponds to the number of action classes. To learn the parameters, we optimize the multi-label binary cross-entropy loss [108].

4.4.5 Experiments

The goal of these experiments is to verify that our proposed method can effectively model complex temporal relations. First, we perform an ablation study to validate the design choice of our model. Second, we compare our model with the current SOTA models on 3 densely annotated datasets to prove its effectiveness.

4.4.5.1 Evaluation datasets

We evaluate our PDAN on three challenging datasets: MultiTHUMOS, Charades and Toyota Smarthome Untrimmed (TSU) dataset. All these three datasets are densely annotated with concurrent actions, allowing us to validate the effectiveness of PDAN in handling complex temporal relations. For all these datasets, we follow the original evaluation settings for the action detection task (i.e., frame-level mAP).

4.4.5.2 Implementation details

In PDAN, we set $N = 5$ blocks, $C_1 = 1024$ and $C_2 = 512$ (see Fig. 4.7). For each DAL in the aforementioned blocks, the kernel and stride size are set to 3 and 1, respectively.

	Dilation	Residual link	DAL in block 1	2	3	4	5	Charades	TSU
Simple(STCL)	×	×	×	×	×	×	×	17.8	15.0
Simple(DAL)	×	×	✓	✓	✓	✓	✓	18.9	16.1
Dilation (STCL)	✓	×	×	×	×	×	×	21.8	24.0
Dilation (DAL)	✓	×	✓	✓	✓	✓	✓	23.2	26.1
Residua (STCL)	×	✓	×	×	×	×	×	21.8	24.3
Residua (DAL)	×	✓	✓	✓	✓	✓	✓	23.5	26.5
PDAN (STCL)	✓	✓	×	×	×	×	×	24.1	29.0
PDAN(Low)	✓	✓	✓	✓	×	×	×	25.3	30.1
PDAN(High)	✓	✓	×	×	×	✓	✓	25.4	30.1
PDAN (DAL)	✓	✓	✓	✓	✓	✓	✓	**26.5**	**32.7**

Table 4.5: Frame-based mAP (%) to show the effectiveness of the components in PDAN. The ✓ indicates that we use this component in all the PDAN blocks. PDAN (DAL) is our proposed PDAN.

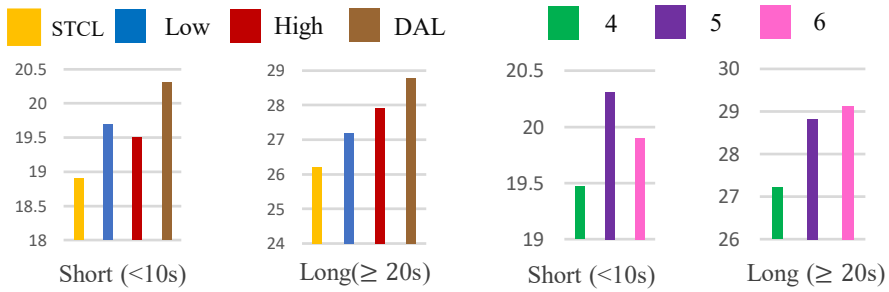


Figure 4.10: The frame-based mAP performance for Short and Long actions on Charades with (1) different levels of attention, (2) different numbers of PDAN Blocks.

The dilation and padding rates are set to $2^{(i-1)}$ for block $i \in [1, N = 5]$. We use Adam optimizer [162] with an initial learning rate of 0.001, and we scale it by a factor of 0.3 with a patience of 10 epochs. The network is trained on a 4-GPU machine for 300 epochs with a mini batch of 32 videos for Charades, 8 videos for MultiTHUMOS and 2 videos for TSU dataset. Depending on the available modalities within the datasets, we use RGB-stream only for TSU dataset and two-stream structure for Charades and MultiTHUMOS datasets. Mean pooling of the prediction logits has been performed to fuse the RGB and Flow streams.

4.4.5.3 Ablation studies

In this section, we demonstrate the effectiveness of each component of our PDAN.

Block components:

In Table 4.5, we first alternatively apply or remove dilation, residual link and DAL in all

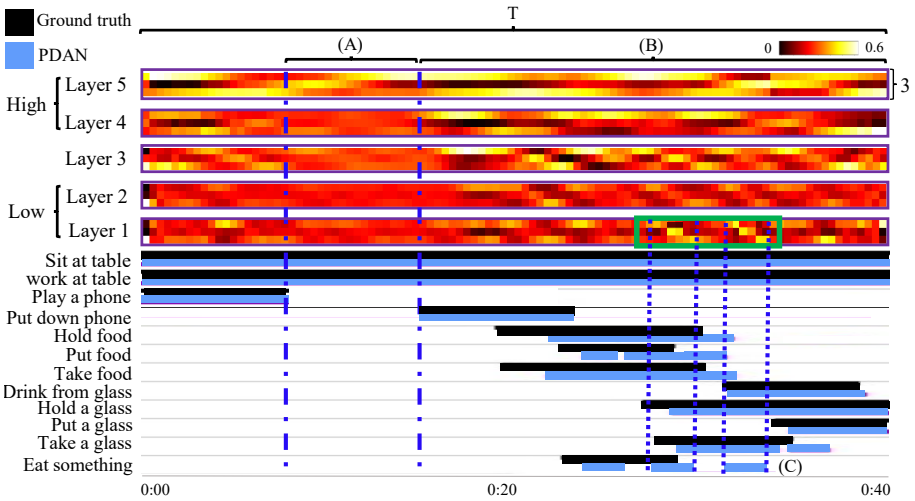


Figure 4.11: Qualitative analysis of the attention map. On the top, we visualize the attention map of DAL for 5 layers ($C_2 \times T \times 3$ for each layer). On the bottom, we present the corresponding **ground truth** and **PDAN** detection for this video.

the blocks to show the effectiveness of these components (see Fig. 4.7). We test three configurations: (1) Simple: no residual link and no dilation[1] in any PDAN's block. (2) Dilation: no residual link but dilation in all the blocks. (3) Residual: no dilation but residual link in all the blocks. We indicate between the brackets when DAL or Standard Temporal Convolution Layer (STCL) is used in the blocks. Note that, DAL and STCL have the same kernel size and dilation rate. Results show that for both datasets dilation and residual link lead to similar improvements (+4.0% on Charades). When accompanied by the residual link (i.e. PDAN (STCL)), dilation boosts the action detection performance by up to 2.3% on TSU w.r.t. dilation only. Using DAL in all the layers, PDAN outperforms all these ablation baselines (+1.1%, +2.1%, +2.2% and +3.7% w.r.t. Simple, Dilation, Residual and PDAN (STCL) on TSU). These results suggest that DAL is a more effective temporal filter than STCL and that dilation with residual link help boost DAL's performance. We then study to which block, attention should be integrated. We apply attention mechanism on different blocks to build four ablation baselines: PDAN (STCL), PDAN (Low), PDAN (High) and PDAN (DAL). Low and high indicates that instead of using STCL, we apply DAL in the first two blocks and last two blocks, respectively. PDAN (Low) and PDAN (High) correspond to a low ($<$ 5.6 sec.) and high ($>$ 24.8 sec.) receptive field respectively. Table 4.5 shows that both baselines can improve the performance (up to 1.3% w.r.t. Residual+Dilation on Charades). In Fig. 4.10 (1), we show that PDAN (Low) can better detect short actions, and PDAN (High) can better detect the Long actions. PDAN incorporates the attention mechanism on all the blocks and achieves the best performance

[1]No dilation indicates that all the blocks are set with dilation rate 1.

Num. Blocks	Temp. Field	Charades	TSU
3	15	23.3	29.4
4	31	25.0	30.3
5	63	**26.5**	**32.7**
6	127	25.6	30.5

Table 4.6: Ablation study to determine the number of blocks in PDAN. "Temp. Field" indicates the length of temporal reception field (expressed in seconds) for the kernel at the last block.

for both long and short actions (+2.4% w.r.t. PDAN (STCL) on Charades dataset). In Fig. 4.11, we present the attention map of DAL for 5 layers (on top), and the corresponding ground truth vs PDAN detection results (on the bottom). In area (A), with only long actions (e.g.*work at table*), only the higher layers allocate high attention weights to the frames in the kernel. This reflects that the higher layers are more sensitive to long-term actions. In area (B), with both long and short actions, both higher and lower layers allocate high attention weights to the frames in the kernel. In area (C) (at the bottom), while detecting short actions, DAL allocates high attention weights at the lower layer, corroborating that the lower layer is particularly sensitive to short actions.

Number of blocks:

Table 4.6 reports the performance while using different numbers of blocks in PDAN. This performance depends on the size of the temporal receptive field and the average action length in the videos. With more blocks, PDAN can have a larger temporal reception field. Here, 5 block structure indicates that PDAN's reception field explores up to 63 segments (i.e. about 1 min), which can satisfy the requirements of both datasets. In Fig. 4.10 (2), we analyse the performance of the number of PDAN blocks for actions with different duration. 5-blocks structure achieves the best performance for frame-based mAP (up to 2.4% w.r.t. 4 block structure on TSU). While increasing to 6-blocks improves the performance for long actions (+0.4%), it deteriorates the performance for short actions. This can be explained by the fact that having more layers tends to diminish the importance of local context.

DAL & NL layer:

. In Table 4.7, we measure the efficiency of DAL compared to the Non-Local (NL) layer [28]. While replacing all the DALs by STCLs in the PDAN block, we obtain PDAN (STCL) (see Fig. 4.7). We have tried two different ways of integrating the NL layer. NL-T1 indicates that we add one NL layer before the classifier in PDAN (STCL); NL-T2 indicates that we replace the DAL layer by a STCL and a NL layer in every PDAN block (see Fig. 4.7). As mentioned in Sec. 4.4.2, PDAN (STCL) and PDAN have similar parameters. Besides, DAL outperforms both NL-T1 and NL-T2 with large margin (+1.9% and +2.4% w.r.t. NL-T1 and NL-T2 on Charades), while having less parameters and less operations (i.e. FLOPs). This result reflects that DAL is more efficient and effective than

	#Param (M)	FLOPs (GMac)	Charades	TSU
PDAN (STCL)	5.9	0.59	24.1	29.0
PDAN (STCL)+NL-T1	6.4	0.65	24.6	29.2
PDAN (STCL)+NL-T2	8.5	0.88	23.9	28.5
PDAN (DAL)	5.9	0.62	**26.5**	**32.7**

Table 4.7: Frame-based mAP (%) to show the effectiveness of the components in PDAN. PDAN (STCL) indicates that we replace DAL in the PDAN block by the standard temporal convolution layer. NL-T1 indicates that we add one Non-Local layer before the PDAN (STCL) classifier. NL-T2 indicates that we add one NL-layer after every STCL in PDAN (STCL).

	#Param	FLOPs	Charades	TSU
I3D+Timeception (STCL)	4.8 M	0.46 G	21.8	27.0
I3D+Timeception (DAL)	4.8 M	0.47 G	23.0	29.3

Table 4.8: Frame-based mAP (%) to show the effectiveness of DAL integrated in Timeception structure.

NL layer for action detection in densely annotated videos.

Timeception + DAL:

Finally, we embed DAL in another structure based on temporal convolution [75] to confirm the effectiveness of DAL. Different from PDAN, Timeception [75] utilizes several temporal convolutions in parallel with different dilation rates. This design enables Timeception to explore multi-temporal scales in one layer. However, Timeception is designed for multi-label action classification, not for action detection. So, it applies max pooling to aggregate the temporal information and halve the temporal resolution at every layer. Hence, we remove the max pooling from the original Timeception structure to utilize the temporal information for the action detection task (i.e. Timeception (STCL)). Based on this new structure, we replace the standard temporal convolution with our proposed DAL (i.e. Timeception (DAL)) to demonstrate that DAL can be combined with other architectures. In Table 4.8, we report the mAP performance of 3-layer Timeception. We find out that Timeception (DAL) improves the base network performance (up to +2.3% on TSU w.r.t. Timeception (STCL)), but it under-performs compared to PDAN.

4.4.5.4 Comparison with State-of-the-Art Methods

The proposed PDAN is compared with previous methods on the MultiTHUMOS, Charades and TSU (CS) datasets in Table 4.9, Table 4.10 and Table 4.11. To be noticed, the I3D baseline (i.e. I3D in the tables) used for comparison is a classifier on top of the segment-level I3D features. Unlike the other SOTA, I3D baseline does not have further temporal processing after the visual encoding part. Thus, this method cannot model long temporal

	mAP
Two-stream [13]	27.6
Two-stream+LSTM [13]	28.1
Multi-LSTM [13]	29.6
SSN [59]	30.3
I3D [18]	29.7
I3D + LSTM [18]	29.9
I3D + temporal pyramid [18]	31.2
TAN [102]	33.3
I3D + Dilated-TCN* [5]	43.2
I3D + 3 TGMs [18]	44.3
I3D + MS-TCN* [19]	45.3
I3D + 3 TGMs + Super event [18]	46.4
I3D + **PDAN**	**47.6**

Table 4.9: Performance of the state-of-the-art methods and our approach on MultiTHUMOS. I3D model is two-stream, using both RGB and optical flow input. Note: cited papers may not be the original paper but the one providing this mAP results. *indicates the results obtained by running the available code.

	Modality	mAP
Two-stream [114]	RGB + Flow	8.9
Two-stream+LSTM [114]	RGB + Flow	9.6
R-C3D [81]	RGB	12.7
Asynchronous Temporal Fields [114]	RGB + Flow	12.8
I3D [17]	RGB	15.6
I3D [17]	RGB + Flow	17.2
I3D + 3 temporal conv.layers [18]	RGB + Flow	17.5
TAN [102]	RGB + Flow	17.6
I3D + WSGN (supervised) [163]	RGB	18.7
I3D + Stacked-STGCN [164]	RGB	19.1
I3D + Super event [17]	RGB + Flow	19.4
I3D + 3 TGMs [18]	RGB + Flow	21.5
I3D + 3 TGMs + Super event [18]	RGB + Flow	22.3
I3D + Dilated-TCN* [5]	RGB + Flow	23.5
I3D + MS-TCN* [19]	RGB + Flow	24.2
I3D + **PDAN**	RGB	**23.7**
I3D + **PDAN**	RGB + Flow	**26.5**

Table 4.10: Per-frame mAP on Charades, evaluated with the Charades localization setting. Note: cited papers may not be the original paper but the one providing this mAP results. *indicates the results obtained by running the available code.

	mAP
R-I3D [81]	8.7
I3D+Dilated-TCN [5]	25.1
I3D+MS-TCN [19]	25.9
I3D+TGM [18]	26.7
I3D+PDAN	**32.7**

Table 4.11: Frame-level mAP on TSU dataset (CS protocol).

information, which is crucial for action detection. In contrast, the other action detection

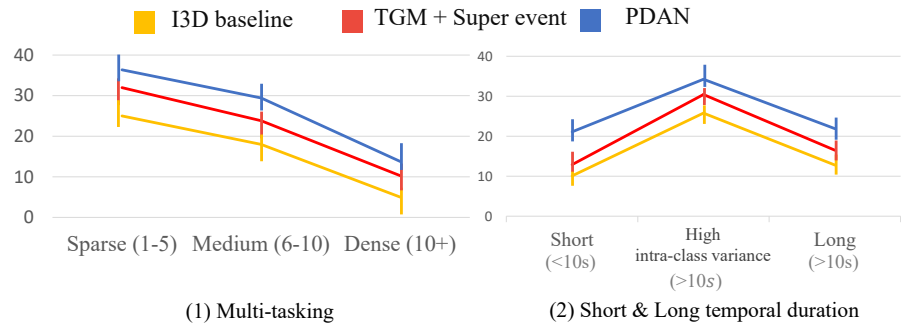


Figure 4.12: Handling 2 challenges related to complex temporal relations on Charades dataset: (1) Multi-tasking, (2) Short and long temporal duration. We calculate the mAP for each group of actions for each challenge.

baselines as [17, 18, 13] focus on the temporal processing. The improvement over I3D baseline reflects the effectiveness of modeling temporal information. PDAN consistently outperforms the prior methods [13, 164, 17, 102, 18] for action detection on all the three challenging datasets. For Dilated-TCN and MS-TCN, PDAN improves the performance with a large margin. In the community, some researchers have also applied the proposed PDAN in egocentric fine-grained action segmentation and revealed promising results [165].

We then study how our proposed method can tackle complex temporal relations. We perform this comparison with I3D baseline [22], and TGM + Super event [18]. In Fig. 4.12, we first study the performance along the multi-tasking challenge on Charades dataset and for detecting both long-term and shot-term temporal duration on TSU dataset with the appropriate metrics. To study the ability of the different approaches to handle concurrent actions, we created 3 groups of actions depending on the number of co-occurring actions per frame. Sparse: 1-5 concurrent actions, Medium: 6-9 concurrent actions and Dense: more than 10 concurrent actions. We compute the mAP for these three groups and find out that PDAN consistently achieves the best performance (see Fig. 4.12 (2)). Secondly, we study the performance along different temporal lengths of the actions. High intra-class temporal variance indicates the actions where the temporal variance is larger than 10 $seconds$. We then separate the remaining actions into short actions ($\leq$10 sec) and long actions ($>$ 10 sec). We find out that PDAN outperforms TGM + Super event for all these action types reflecting better handling of both short-term and long-term duration. Thanks to the use of the dilated attention layers with multi-temporal scales, PDAN can deal with actions of variable length. This comparison with SOTA methods confirms that PDAN can better handle complex temporal relations for actions from densely annotated untrimmed videos.

4.5 Multi-Scale Temporal ConvTransformer

First, we define the problem statement of action detection in densely-labelled settings. Formally, for a video sequence of length T, each time-step t contains a ground-truth action label $y_{t,c} \in \{0, 1\}$, where $c \in \{1, ..., C\}$ indicates an action class. For each time-step, an action detection model needs to predict class probabilities $\tilde{y}_{t,c} \in [0, 1]$. Here, we describe our proposed action detection network: *MS-TCT*. As depicted in Fig. 4.13, it consists of four main components: (1) a ***Visual Encoder*** which encodes a preliminary video representation, (2) a ***Temporal Encoder*** which structurally models the temporal relations at different temporal scales (i.e., resolution), (3) a ***Temporal Scale Mixer***, dubbed as *TS Mixer*, which combines multi-scale temporal representations, and (4) a ***Classification Module*** which predicts class probabilities. In the following sections, we present the details of each MS-TCT component.

4.5.1 Visual Encoder

The input to our action detection network: MS-TCT, is an untrimmed video which may span for a long duration [43] (e.g. multiple minutes). However, processing long videos in both spatial and temporal dimensions can be challenging, mainly due to computational burden. As a compromise, similar to previous action detection models [7, 18], we consider features of video segments extracted by a 3D CNN as inputs to MS-TCT, which embed spatial information latently as channels. Specifically, we use an I3D backbone [22] to encode videos. Each video is divided into T non-overlapping segments (during training), each of which consists of 8 frames. Such RGB frames are fed as an input segment to the I3D network. Each segment-level feature (output of I3D) can be seen as a transformer token of a time-step (i.e., temporal token). We stack the tokens along the temporal axis to form a $T \times D_0$ video token representation, to be fed in to the Temporal Encoder.

4.5.2 Temporal Encoder

As previously highlighted in Section 5.1, efficient temporal modeling is critical for understanding long-term temporal relations in a video, especially for complex action compositions. Given a set of video tokens, there are two main ways to model temporal information: using (1) a 1D Temporal Convolutional layer [5], which focuses on the neighboring tokens but overlooks the direct long-term temporal dependencies in a video, or (2) a Transformer [107] layer that globally encodes one-to-one interactions of all tokens, while neglecting the local semantics, which has proven beneficial in modeling the highly-correlated visual signals [166, 167]. Our Temporal Encoder benefits from the best of both worlds, by exploring both local and global contextual information in an alternating

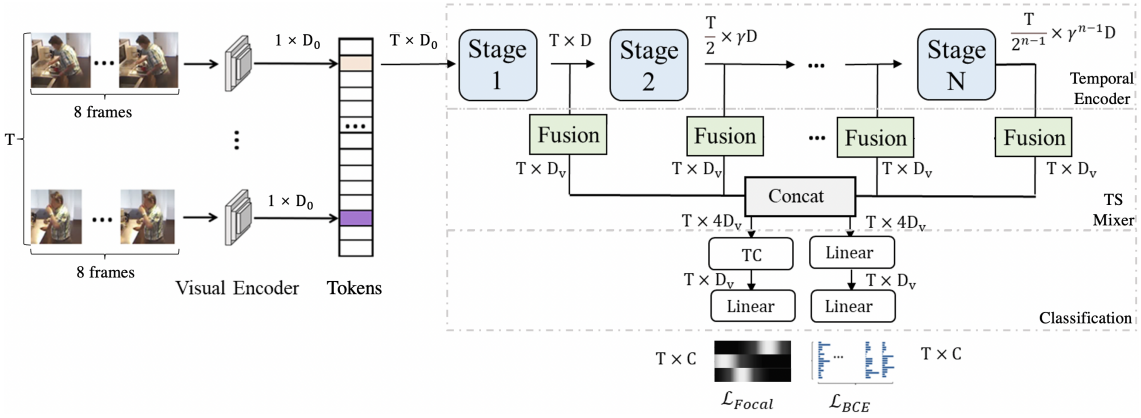


Figure 4.13: The Multi-Scale Temporal ConvTransformer (MS-TCT) for action detection is composed of four main parts. (1) Visual Encoder, (2) Temporal Encoder, (3) Temporal Scale Mixer (TS Mixer) and (4) Classification Module. Note that TC indicates the 1-dimensional convolutional layer with kernel size k.

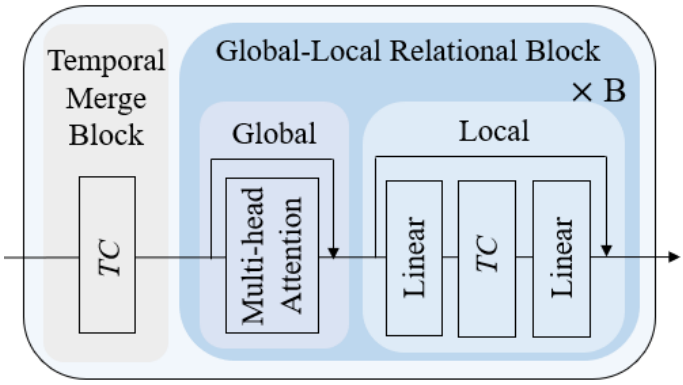


Figure 4.14: **A single stage of our Temporal Encoder** consists of (1) a Temporal Merging Block and (2) $\times B$ Global-Local Relational Blocks. Each Global-Local Relational Block contains a Global and a Local Relational Block. Here, $Linear$ and TC indicates the 1D convolutional layer with kernel size 1 and k respectively.

fashion.

As shown in Fig. 4.13, Temporal Encoder follows a hierarchical structure with N stages: Earlier stages learn a fine-grained action representation with more temporal tokens, whereas the latter stages learn a coarse representation with fewer tokens. Each stage corresponds to a semantic level (i.e., temporal resolution) and consists of one Temporal Merging block and $\times B$ Global-Local Relational Blocks (see Fig. 4.14):

Temporal Merging Block is the key component for introducing network hierarchy, which shrinks the number of tokens (i.e., temporal resolution) while increasing the feature dimension. This step can be seen as a weighted pooling operation among the neighboring tokens. In practice, we use a single temporal convolutional layer (with a kernel size of k,

and a stride of 2, in general) to halve the number of tokens and extend the channel size by $\times\gamma$. In the first stage, we keep a stride of 1 to maintain the same number of tokens as the I3D output, and project the feature size from D_0 to D (see Fig. 4.13). This is simply a design choice.

Global-Local Relational Block is further decomposed in to a *Global Relational Block* and a *Local Relational Block* (see Fig. 4.14). In Global Relational Block, we use the standard multi-head self-attention layer [107] to model long-term action dependencies, i.e., global contextual relations. In Local Relational Block, we use a temporal convolutional layer (with a kernel size of k) to enhance the token representation by infusing the contextual information from the neighboring tokens, i.e., local inductive bias. This enhances the temporal consistency of each token while modeling the short-term temporal information corresponding to an action instance.

In the following, we formulate the computation flow inside the Global-Local Relational Block. For brevity, here, we drop the stage index n. For a block $j \in \{1, ..., B\}$, we represent the input tokens as $X_j \in \mathbb{R}^{T' \times D'}$. First, the tokens go through multi-head attention layer in Global Relational Block, which consists of H attention heads. For each head $i \in \{1, ..., H\}$, an input X_j is projected in to $Q_{ij} = W^Q_{ij} X_j$, $K_{ij} = W^K_{ij} X_j$ and $V_{ij} = W^V_{ij} X_j$, where W^Q_{ij}, W^K_{ij}, $W^V_{ij} \in \mathbb{R}^{D_h \times D'}$ represent the weights of linear layers and $D_h = \frac{D'}{H}$ represents the feature dimension of each head. Consequently, the self-attention for head i is computed as,

$$Att_{ij} = \texttt{Softmax}(\frac{Q_{ij} K^\top_{ij}}{\sqrt{D_h}}) V_{ij} \ . \tag{4.8}$$

Then, the output of different attention heads are mixed with an additional linear layer as,

$$M_j = W^O_j \texttt{Concat}(Att_{1j}, ..., Att_{Hj}) + X_j \ , \tag{4.9}$$

where $W^O_j \in \mathbb{R}^{D' \times D'}$ represents the weight of the linear layer. The output feature size of multi-head attention layer is the same as the input feature size.

Next, the output tokens of multi-head attention are fed in to the *Local Relational Block*, which consists of two linear layers and a temporal convolutional layer. As shown in Fig. 4.14, the tokens first go through a linear layer to increase the feature dimension from D' to $\theta D'$, followed by a temporal convolutional layer with a kernel size of k, which blends the neighboring tokens to provide local positional information to the temporal tokens [149]. Finally, another linear layer projects the feature dimension back to D'. The two linear layers in this block enable the transition between the multi-head attention layer and temporal convolutional layer. The output feature dimension remains the same as the input feature for the Local Relational Block. This output is fed to the next Global Relational Block if block $j < B$.

The output tokens from the last Global-Local Relational Block from each stage are combined and fed to the following Temporal Scale Mixer.

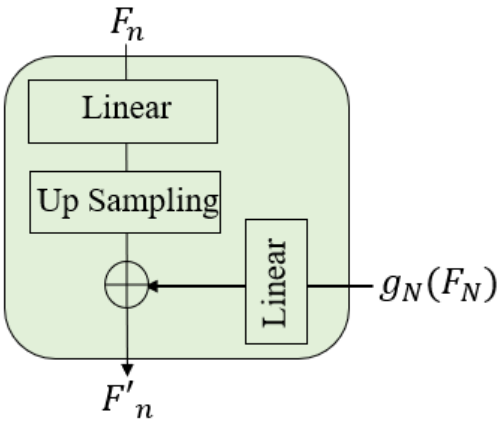


Figure 4.15: **Temporal Scale Mixer Module:** The output tokens F_n of stage n is resized and up-sampled to $T \times D_v$, then summed with the tokens from the last stage N.

4.5.3 Temporal Scale Mixer

After obtaining the tokens at different temporal scales, the question that remains is, *how to aggregate such multi-scale tokens to have a unified video representation*? To predict the action probabilities, our classification module needs to make predictions at the original temporal length as the network input. Thus, we require to interpolate the tokens across the temporal dimension, which is achieved by performing an up-sampling and a linear projection step. As shown in Fig. 4.15, for the output F_n from stage $n \in \{1, ..., N\}$, this operation can be formulated as,

$$g_n(F_n) = \texttt{UpSampling}_n(F_n W^n) \ , \tag{4.10}$$

where $W^n \in \mathbb{R}^{D_v \times \gamma^{n-1} D}$ with an upsampling rate of n. In our hierarchical architecture, earlier stages (with lower semantics) have higher temporal resolution, whereas the latter stages (with high semantics) have lower temporal resolution. To balance the resolution and semantics, upsampled tokens from the last stage N is processed through a linear layer and summed with the upsampled tokens from each stage ($n < N$). This operation can be formulated as,

$$F'_n = g_n(F_n) \oplus g_N(F_N) W_n \ , \tag{4.11}$$

where F'_n is the refined tokens of stage n, $\oplus$ indicates the element-wise addition and $W_n \in \mathbb{R}^{D_v \times D_v}$. Here, all the refined token representations have the same temporal length. Finally, we concatenate them to get the final multi-scale video representation $F_v \in \mathbb{R}^{T \times N D_v}$.

$$F_v = \texttt{Concat}(F'_1, ..., F'_{N-1}, F_N) \ . \tag{4.12}$$

Note that more complicated fusion methods [168, 169] can be built on top of these multi-scale tokens. However, we see that the simple version described above performs the best.

The multi-scale video representation F_v is then sent to the classification module for making predictions.

4.5.4 Classification Module

Training MS-TCT is achieved by jointly learning two classification tasks. As mentioned in Section 4.1, in this work, we introduce a new classification branch to learn a heat-map of the action instances. This heat-map is different from the ground truth label as it varies across time, based on the action center and duration. The objective of using such heat-map representation is to encode temporal relative positioning in the learned tokens of MS-TCT.

In order to train the heat-map branch, we first need to build the *class-wise* ground-truth heat-map response $G^* \in [0,1]^{T \times C}$, where C indicates the number of action classes. In this work, we construct G^* by considering the maximum response of a set of one-dimensional Gaussian filters. Each Gaussian filter corresponds to an instance of action class in a video, centered at the specific action instance, in time. More precisely, for every temporal location t the ground-truth heat-map response is formulated as,

$$G^*_c(t) = \max_{a=1,\dots,A_c} \texttt{Gaussian}(t, t_{a,c}; \sigma) \ , \tag{4.13}$$

$$\texttt{Gaussian}(t, t_{a,c}; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \texttt{exp}^{-\frac{(t-t_{a,c})^2}{2\sigma^2}} \ . \tag{4.14}$$

Here, $\texttt{Gaussian}(\cdot, \cdot; \sigma)$ provides an instance-specific Gaussian activation according to the center and instance duration. Moreover, σ is equal to $\frac{1}{2}$ of each instance duration and $t_{a,c}$ represents the center for class c and instance a. A_c is the total number of instances for class c in the video. As shown in Fig. 4.13, heat-map G is computed using a temporal convolutional layer with a kernel size of k and a non-linear activation, followed by another linear layer with a sigmoid activation. Given the ground-truth G^* and the predicted heat-map G, we compute the *action focal loss* [170] which is formulated as,

$$\mathcal{L}_{\text{Focal}} = \frac{1}{A} \sum_{t,c} \begin{cases} (1 - G_{t,c})^2 \texttt{log}(G_{t,c}) & \text{if } G^*_{t,c} = 1 \ , \\ (1 - G^*_{t,c})^4 (G_{t,c})^2 \texttt{log}(1 - G_{t,c}) & \text{if } G^*_{t,c} \neq 1; \ , \end{cases} \tag{4.15}$$

where A is the total number of action instances in a video.

Similar to the previous work [7], we leverage another branch to perform the usual multi-label classification. With video features F_v, the predictions are computed using two linear layers with a sigmoid activation, and Binary Cross Entropy (BCE) loss [108] is computed against the ground-truth labels. Only the scores predicted from this branch are used in evaluation. Input to both the branches are the same output tokens F_v. The heat-map branch encourages the model to embed the relative position w.r.t. the instance center in to video tokens F_v. Consequently, the classification branch can also benefit from

such positional information to make better predictions. The overall loss is formulated as a weighted sum of the two losses mentioned above, with the weight α is chosen according to the numerical scale of losses.

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{BCE}} + \alpha \, \mathcal{L}_{\text{Focal}} \, . \tag{4.16}$$

4.5.5 Experiments

Datasets: Similar to PDAN, we evaluate our framework on three challenging multi-label action detection datasets: Charades, TSU and MultiTHUMOS.

Implementation Details: In the proposed network, we use number of stage $N = 4$ the number of Global-Local Relational Blocks $B = 3$ for each stage. Note that for small dataset as MultiTHUMOS, $B = 2$ is sufficient. The number of attention heads for the Global Relational Block is set to 8. We use the same output feature dimension of I3D (after `Global Average Pooling`) as input to MS-TCT, and thus $D_0 = 1024$. Input features are then projected in to $D = 256$ dimensional feature using the temporal merging block in the first stage. We consider feature expansion rate $\gamma = 1.5$ and $\theta = 8$. Kernel size k of temporal convolutional layer is set to be 3, with zero padding to maintain the resolution. The loss balance factor $\alpha = 0.05$. The number of tokens is fixed to $T = 256$ as input to MS-TCT. During training, we randomly sample consecutive T tokens from a given I3D feature representation. At inference, we follow [7] to use a sliding window approach to make predictions. Our model is trained on two GTX 1080 Ti GPUs with a batch-size of 32. We use Adam optimizer [162] with an initial learning rate of 0.0001, which is scaled by a factor of 0.5 with a patience of 8 epochs.

4.5.5.1 Ablation Study

In this section, we study the effectiveness of each component in the proposed network on Charades dataset.

Importance of Each Component in MS-TCT: As shown in Table 4.12, I3D features with the classification branch only, is considered as the representative baseline. This baseline consists in a classifier that discriminates the I3D features at each time-step without any further temporal modeling. On top of that, adding our Temporal Encoder significantly improves the performance (+ 7.0%) *w.r.t.* I3D feature baseline. This improvement reflects the effectiveness of the Temporal Encoder in modeling the temporal relations within the videos. In addition, if we introduce a Temporal Scale Mixer to blend the features from different temporal scales, it gives a + 0.5% improvement, with minimal increase in computations. Finally, we study the utility of our heat-map branch in the classification module. We find that the heat-map branch is effective when optimized along with

Temporal Encoder	TS Mixer	Heat-Map Branch	Classification Branch	mAP (%)
×	×	×	✓	15.6
✓	×	×	✓	23.6
✓	✓	×	✓	24.1
✓	✓	✓	×	10.7
✓	✓	✓	✓	**25.4**

Table 4.12: **Ablation on each component in MS-TCT:** The evaluation is based on per-frame mAP on Charades dataset.

Temporal Merge	Global Layer	Local Layer	mAP (%)
✓	✓	×	24.0
✓	×	✓	20.9
×	✓	✓	22.7
✓	✓	✓	**25.4**

Table 4.13: **Ablation on the design of a single stage in our Temporal Encoder**, evaluated using per-frame mAP on Charades dataset.

the classification branch, but fails to learn discriminative representations when optimized without it (25.4% vs 10.7%). The heat-map branch encourages the tokens to predict the action center while down-playing the tokens towards action boundaries. In comparison, the classification branch improves the token representations equally for all tokens, despite action boundaries. Thus, when optimized together, both branches enable the model to learn a better action representation. While having all the components, the proposed network achieves a significant + 9.8% improvement *w.r.t.* I3D feature baseline validating that each component in MS-TCT is instrumental for the task of action detection.

Design Choice for a Stage: In Table 4.13, we present the ablation related to the design choices of a stage in the Temporal Encoder. Each row in Table 4.13 indicates the result of removing a component in each stage. Note that, removing the Temporal Merge block indicates replacing this block with a temporal convolutional layer of stride 1, i.e., only the channel dimension is modified across stages. In Table 4.13, we find that removing any component can drop the performance with a significant margin. This observation shows the importance of jointly modeling both global and local relations in our method, and the effectiveness of the multi-scale structure. These properties in MS-TCT make it easier to learn complex temporal relationships which span across both (1) neighboring temporal segments, and (2) distant temporal segments.

Analysis of the Local Relational Block: We also dig deeper in to the Local Relational Block in each stage. As shown in Fig. 4.14, there are two linear layers and one temporal convolutional layer in a Local Relational Block. In Table 4.14, we further perform

Feature Expansion Rate (θ)	Temporal Convolution	mAP (%)
8	×	22.3
×	✓	22.4
1	✓	24.2
4	✓	24.9
8	✓	**25.4**

Table 4.14: **Ablation on the design of Local Relational Block:** Per-frame mAP on Charades using only RGB input. × indicates we remove the linear or temporal convolutional layer. Feature expansion rate 1 indicates that the feature-size is not changed in the Local Relational Block.

	Backbone	GFLOPs	Charades	MultiTHUMOS	TSU
R-C3D [81]	C3D	-	12.7	-	8.7
Super-event [17]	I3D	0.8	18.6	36.4	17.2
TGM [18]	I3D	1.2	20.6	37.2	26.7
PDAN [45]	I3D	3.2	23.7	40.2	32.7
Coarse-Fine [16]	X3D	-	25.1	-	-
MLAD [7]	I3D	44.8	18.4	42.2	-
MS-TCT	I3D	6.6	**25.4**	**43.1**	**33.7**

Table 4.15: **Comparison with the state-of-the-art methods** on three densely labelled datasets. Backbone indicates the visual encoder. Note that the evaluation for the methods is based on per-frame mAP (%) using only RGB videos.

ablations of these components. First, we find that without the temporal convolutional layer, the detection performance drops. This observation shows the importance of mixing the transformer tokens with a temporal locality. Second, we study the importance of the transition layer (i.e., linear layer). When the feature size remains constant, having the transition layer can boost the performance by + 1.8%, which shows the importance of such transition layers. Finally, we study how the expansion rate affects the network performance. While setting different feature expansion rates, we find that temporal convolution can better model the local temporal relations when the input feature is in a higher dimensional space.

4.5.5.2 Comparison to the State-of-the-Art

In this section, we compare MS-TCT with the state-of-the-art action detection methods (see Table 4.15). Proposal based methods, such as R-C3D [81] fail in multi-label datasets due to the highly-overlapping action instances, which challenge the proposal and NMS-based methods. Superevent [17] superimposes a global representation to each local feature based on a series of learnable temporal filters. However, the distribution of actions varies from one video to the other. As super-event learns a fixed filter location for all the videos in the training distribution, this location is suitable to mainly actions with high frequency.

	$\tau = 0$				$\tau = 20$				$\tau = 40$			
	P_{AC}	R_{AC}	$F1_{AC}$	mAP_{AC}	P_{AC}	R_{AC}	$F1_{AC}$	mAP_{AC}	P_{AC}	R_{AC}	$F1_{AC}$	mAP_{AC}
I3D	14.3	1.3	2.1	15.2	12.7	1.9	2.9	21.4	14.9	2.0	3.1	20.3
CF	10.3	1.0	1.6	15.8	9.0	1.5	2.2	22.2	10.7	1.6	2.4	21.0
MLAD [7]	19.3	7.2	8.9	28.9	18.9	8.9	10.5	35.7	19.6	9.0	10.8	34.8
MS-TCT	**26.3**	**15.5**	**19.5**	**30.7**	**27.6**	**18.4**	**22.1**	**37.6**	**27.9**	**18.3**	**22.1**	**36.4**

Table 4.16: **Evaluation on the Charades dataset using the action-conditional metrics:** Similar to MLAD, both RGB and Optical flow are used for the evaluation. P_{AC} - Action-Conditional Precision, R_{AC} - Action-Conditional Recall, $F1_{AC}$ - Action-Conditional F1-Score, mAP_{AC} - Action-Conditional Mean Average Precision. τ indicates the temporal window size.

TGM [18] and PDAN are methods based on temporal convolution of video segments. Nevertheless, those methods only process videos locally at a single temporal scale. Thus, they are not effective in modeling long-term dependencies and high-level semantics. Coarse-Fine Network [16] achieves 25.1% on Charades. However, this method is built on top of the visual encoder X3D [57], which prevents the usage of higher number of input frames. Moreover, it relies on a large stride between the frames. Therefore, it fails to model fine-grained action relations, and can not process long videos in MultiTHUMOS and TSU. MLAD [7] jointly models action class relations for every time-step and temporal relations for every class. This design leads to a huge computational cost, while under-performing on datasets with a large number of action classes (e.g. Charades). Thanks to the combination of transformer and convolution in a multi-scale hierarchy, the proposed MS-TCT consistently outperforms previous state-of-the-art methods in all three challenging multi-label action detection datasets that we considered. We also compare the computational requirement (FLOPs) for the methods built on top of the same Visual Encoder (i.e., I3D features), taking as input the same batch of data. We observe that the FLOPs of MS-TCT is higher with a reasonable margin than pure convolutional methods (i.e., PDAN, TGM, super-event). However, compared to a transformer based action detection method MLAD, MS-TCT uses only $\frac{1}{7}$th of the FLOPs.

We also evaluate our network with the action-conditional metrics introduced in [7] on Charades dataset in Table 4.16. These metrics are used to measure a method's ability to model both co-occurrence dependencies and temporal dependencies of action classes. Although our network is not specifically designed to model cross-class relations as in MLAD, it still achieves higher performance on all action-conditional metrics with a large margin, showing that MS-TCT effectively models action dependencies both within a time-step (i.e., co-occurring action, $\tau = 0$) and throughout the temporal dimension ($\tau > 0$).

Finally, we present a qualitative evaluation for PDAN and MS-TCT on the Charades dataset in Fig. 4.16. As the prediction of the Coarse-Fine Network is similar to the X3D network which is limited to dozens of frames, thus we can not compare with the Coarse-

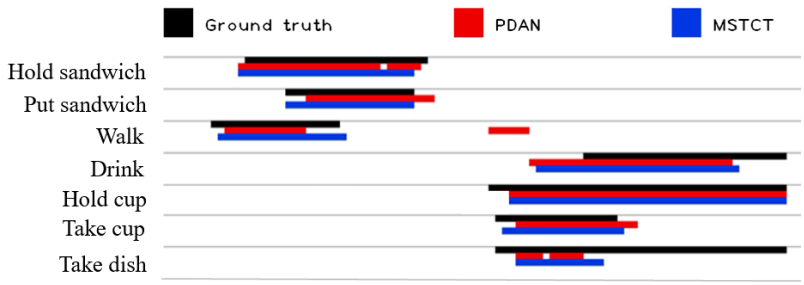


Figure 4.16: **Visualization of the detection results** on an example video along time axis. In this figure, we visualize the ground truth and the detection of PDAN and MS-TCT.

Fine network on the whole video. Here, we observe that MS-TCT can predict action instances more precisely compared to PDAN. This comparison reflects the effectiveness of the transformer architecture and multi-scale temporal modeling.

Table 4.17: **Study on stage type** showing the effect of having both convolutions and self-attention.

Stage-Type	mAP
Pure Transformer	22.3
Pure Convolution	21.4
ConvTransformer	**25.4**

Table 4.18: **Study on** σ showing the effect of scale of Gaussians in heat-maps.

Variance: σ	mAP
1/8 duration	24.6
1/4 duration	24.8
1/2 duration	**25.4**

4.5.5.3 Discussion and Analysis

Transformer, Convolution or ConvTransformer? To confirm the effectiveness of our ConvTransformer, we compare with a pure transformer network and a pure convolution network. Each network has the same number of stages as MS-TCT with similar settings (e.g. blocks, feature dimension). In pure transformer, a pooling layer and a linear layer constitute the temporal merging block, followed by B transformer blocks in each stage. A transformer block is composed of a multi-head attention layer, norm-add operations and a feed-forward layer. A learned positional embedding is added to the input tokens to encode the positional information. This pure transformer architecture achieves 22.3% on Charades. In pure convolution-based model, we retain the same temporal merging block as in MS-TCT, followed by a stack of B temporal convolution blocks. Each block consists of a temporal convolution layer with a kernel-size of k, a linear layer, a non-linear activation and a residual link. This pure temporal convolution architecture achieves 21.4% on Charades. In contrast, the proposed ConvTransformer outperforms both the pure transformer and the pure convolutional network by a large margin (+ 3.1%, and + 4.0% on Charades, respectively. See Table 4.17). It shows that ConvTransformer can better model the temporal relations of complex actions.

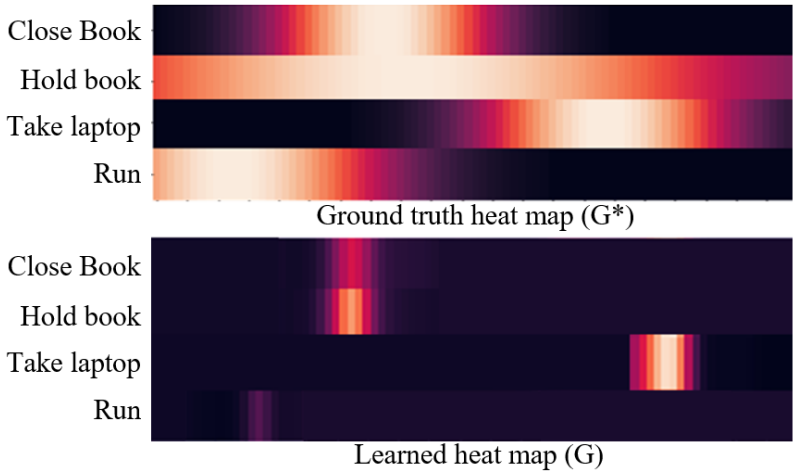


Figure 4.17: **Heat-map visualization along time axis:** On the top, we show the ground truth heat-map (G^*) of the example video. On the bottom is the corresponding learned heat-map (G) of MS-TCT. As the heat-map is generated by a Gaussian function, the lighter region indicates closer to the center of the instance.

Heat-map Analysis: We visualize the ground truth heat-map (G^*) and the corresponding predicted heat-map (G) in Fig. 4.17. We observe that with the heat-map branch, MS-TCT predicts the center location of the action instances, showing that MS-TCT embeds the center-relative information in to the tokens. However, as we optimize with the focal loss to highlight the center, the boundaries of the action instance in this heat-map are less visible. We then study the impact of σ on performance. As shown in Table 4.18, we set σ to be either $\frac{1}{8}$, $\frac{1}{4}$ or $\frac{1}{2}$ of the instance duration while generating the ground-truth heat-map G^*. MS-TCT improves by + 0.5%, + 0.7%, + 1.3% respectively *w.r.t.* the MS-TCT without the heat-map branch, when G^* set to different σ. This result reflects that a larger σ can better provide the center-relative position. We investigate further by adding a heat-map branch to our PDAN (see Sec. 4.4). Although the heat-map branch also improves PDAN (+ 0.4 %), the relative improvement is lower compared to MS-TCT (+ 1.3 %). Our method features a multi-stage hierarchy along with a TS Mixer. As the heat-map branch takes input from all the stages, the center-relative position is embedded even in an early stage. Such tokens with the relative position information, when fed through the following stages, benefits the multi-head attention to better model temporal relations among the tokens. This design makes MS-TCT to better leverage the heat-map branch compared to PDAN.

Number of Tokens T. As mentioned in the implementation details, we randomly select consecutive T tokens for each video in the training phase and utilize the sliding window at inference. Here, we have studied how the number of tokens T affects the action detection performance. When T is set to 128, 256 and 512 tokens, MS-TCT achieves 25.0%, 25.4%

and 25.5% on Charades. There is no significant difference in the action detection performance while changing the number of input tokens. However, increasing the number of tokens T in MS-TCT increases the FLOPs. For the trade-off between the computation cost and performance precision, we set T to 256 tokens, which corresponds to 2048 frames (about 86 sec.) of video.

Temporal Positional Embedding: We further study whether the Temporal Encoder of MS-TCT benefits from positional embedding. We find that the performance drops by 0.2% on Charades when a learnable positional embedding [146] is added to the input tokens before processing them with the Temporal Encoder. This shows that the current design can implicitly provide a temporal positioning for the tokens. Adding further positional information to the tokens makes it redundant, leading to lower detection performance.

4.6 Conclusion

In this chapter, we study different strategies for modelling temporal dependencies in untrimmed videos. Our focus lies in how to leverage the attention mechanism to enhance temporal modelling.

Firstly, we introduce the SA-TCN. This network features an encoder-decoder architecture along with a self-attention block in-between the encoder and decoder to model the temporal dependencies in long-term videos. However, the encoder network shrinks the temporal features into a low-resolution status and the decoder network then recovers the information. Limited by the low temporal resolution in the middle stage, SA-TCN can not effectively detect the fine-grained short actions from the video and the region with co-occurring actions. As SA-TCN aims at processing temporal relations in long-term videos, we evaluate this model with DAHLIA dataset, which has an average video duration of 40 mins. SA-TCN achieves competitive performance with respect to the state-of-the-art methods.

For detecting actions with variant length and dense occurrence, we propose PDAN, which is a temporal convolutional network. PDAN features temporal kernels which are adaptive to the input data based on the proposed kernel-level attention mechanism. This property makes PDAN able to model the complex temporal relations across snippets in videos. Although PDAN achieves state-of-the-art performance in modelling complex temporal relations, the distant cross-snippet relations can only be obtained based on the result of low-level layers. We still need a framework that can model the long-term temporal relations more effectively and directly. Recently, we propose the MS-TCT, which inherits a transformer encoder architecture, while also gaining benefits from temporal convolution. With the hierarchy structure, our method can model temporal dependencies both globally and locally at different temporal scales. Moreover, the heat-map branch that introduce

in MS-TCT can help model the action centre location and predict the action occurrence, especially for videos with dense action occurrence. Both PDAN and MS-TCT outperform state-of-the-art methods on the multi-label action detection benchmarks. As MS-TCT can further model the temporal dependencies in multiple temporal scales, it provides a more precise detection than PDAN.

Chapter 5

Semantic Relational Reasoning for Action Understanding

As explained in the previous chapters, one of the major challenges in video analytic involves detecting fine-grained actions in the video. We argue that learning the semantic relations in the video can help to learn the representation of the challenging fine-grained actions. Consequently, in this chapter we propose two models following a similar framework aiming to be effective for fine-grained action recognition and detection. The work presented in this chapter has been published as full conference papers in The British Machine Vision Conference (BMVC) in 2021 [47] and International Conference on Pattern Recognition (ICPR) in 2022 [48].

5.1 Introduction

Real-world videos contain rich semantic information. For instance, understanding the relation between *knife* and *vegetables* can help to detect the action *cutting vegetables*. Also, knowing the existence of the action class "*open the book*" can help detect the action "*reading book*" in a video. Such semantic information can help represent the challenging fine-grained actions as these actions always feature low inter-class motion variations and fine-grained object details (e.g., *drink from bottle* and *eat snack*).

As introduced in the previous chapter, existing methods have mostly focused on modelling the variations of visual cues (i.e., features extracted by visual encoder) across time locally [5] or globally [17] within a video. However, these methods only take into account the temporal information without any further "semantics". Real-world videos contain many complex actions with inherent relationships between action classes at the same time steps or across distant time steps (see Fig. 5.1). Modelling such class-temporal relationships can be extremely useful for locating actions in those videos. Moreover, a sequential

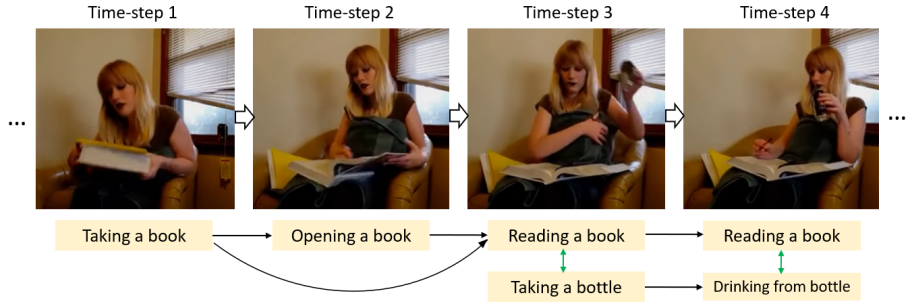


Figure 5.1: Class-temporal relation. In a densely labelled video, there are dependencies between action classes (1) across **different** time steps in black arrows and (2) at the **same** time step (i.e. co-occurring actions) in green arrows.

processing model is problematic when the action instances in a video have non-sequential dependencies or non-linear temporal ordering: for example, overlapping action instances or re-occurrence of action instances over the course of the video [77].

To this end, we introduce a **Class-Temporal Relational Network (CTRN)** to harness the relationships among the action classes in a video to enhance action detection. To explore such relations, CTRN first filters the class-specific representations from the input features at each time step in a video. Then, the transformed per-class representation is utilized for modelling the inter-class relations. (1) Across **different** time steps, a graph-based layer is proposed to learn the dependencies between different action classes of the video. This learned relation map is shared between all the time steps to refine the action features of the related actions (e.g. *open the book* and *read book*). Then, a temporal layer is used to aggregate features from the same class over time to allow the graph-based layer to explore both short-term and long-term class dependencies. (2) At the **same** time step, a graph-based classifier is proposed to leverage the privileged co-occurring action probabilities to improve co-occurring action detection. We evaluate our model on challenging densely labelled datasets such as Charades and MultiTHUMOS for the action detection task. Our method outperforms state-of-the-art results using fewer parameters and FLOPs.

Due to the limitation of computation resources, offline action detection methods are built on top of pre-extracted flattened 1-dimensional features (i.e., snippet-level feature). Although those features still preserve the spatial video information latently, the dissociation between the visual encoder and the temporal module limits the action detection model to directly model the appearance and spatial semantics in the video. To verify that our method can be generalized to extract the spatial-temporal semantics and to model their relationships (see Fig. 5.2), we construct a semantic-reasoning enhanced visual encoder, **Temporal Human Object Relation Network (THORN)**, for action recognition. THORN follows a similar framework as CTRN, but it aims at modelling the relations of object semantics in the video clips. The main difference lies in the semantic extraction

part: THORN extracts the semantics of the objects from the spatial-temporal representation of the visual encoder. To ensure the object-specific semantics, those extracted object representations are supervised by the pseudo-labels generated by an pre-trained object classifier. With the semantic extraction module, THORN can enhance the human or object representation (i.e., noun) and capture the relation across the human and the objects in the videos (i.e., verb), which results in better representing the fine-grained actions and categorizing them. To show the robustness of THORN, we evaluate it on EPIC-KITCHENS 55 and EGTEA Gaze+, two challenging first-person datasets with many human-object interactions. THORN achieves competitive state-of-the-art performance on both datasets.

In the following sections, we review the previous semantic reasoning methods for action detection and we introduce the proposed methods in detail.

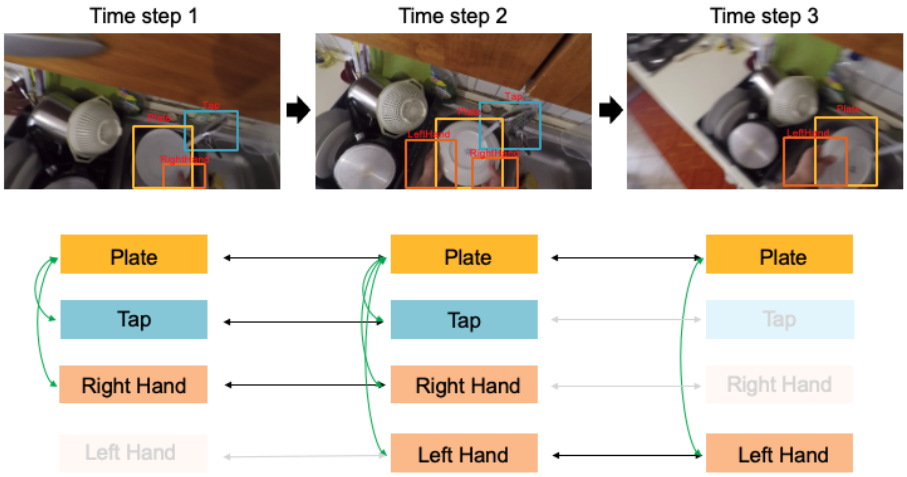


Figure 5.2: An example of the Human-Object Interactions of *wash plate* in an first-view video. Green arrows represent interactions at the same time step (i.e., spatial relation) while black arrows represent interactions across time. In practice, the model captures all the detected objects. For simplicity reasons, here we highlight only the relevant objects related to *wash plate*. The sampled frames are taken from EPIC-KITCHENS.

5.2 Related Work

In this section, we review the methods that designed to model the relations between different semantics. Recently, graphs have been a popular way for modelling relation between the semantics in the video [96, 171, 15, 172].

In action understanding, Lan et al. [173] propose to represent videos by a hierarchy of mid-level action elements (MAEs), where each MAE corresponds to an action-related spatio-temporal segment in the video in an unsupervised manner. This method is capable of distinguishing action-related segments from background segments and representing

actions at multiple spatio-temporal resolutions. Sigurdsson et al. [114] propose a fully-connected temporal CRF model for reasoning over variant intent of videos. The intent are defined as the clustering of similar activities (e.g., actions, object) in a video. Fully-connected CRFs are applied as a post-processing of per-frame CNN features and object features. Although these approaches can structural the video using semantics, they do not learn the explicit temporal structure, nor are they learned in an end-to-end fashion.

In recent years, Wang et al. [174] propose to utilize the graph to represent the video and modelling the interaction between objects and humans via graph reasoning. The method is built on top of I3D features and formulates the node representation by ROI alignment and via an object detector. However, the method that defines its nodes by using ROI-Align from the encoded feature is not optimal. This is because multiple objects are present at the scene and some of them are very close to each other in most cases. As a result, the projected coordinates of different objects tend to be in the same feature patch (a set of pixels). Therefore, extracting an object's specific feature from a feature map with low resolution becomes difficult. For this reason, they can not extract the object semantics precisely. Likewise, Ghosh et al. [164] proposed a method based on Graph Convolutional Network (GCN), namely stacked-STGCN, which extend STGCN [171] for action detection. Different from standard STGCN where the nodes of a graph represent the body joints, in stacked-STGCN, the nodes represent different elements related to the actions such as actors, objects, etc. Nodes are connected along the spatial and temporal dimensions to form the edges of the graph. Such a graph representation characterizes better the complex object-based actions in videos. But the challenge of handling actions over a long range of time still persists. Moreover, the ROI Align issue also exists in stacked-STGCN. Recently, Zhang et al. [175] extract action-specific feature descriptors for each action and learn action correlations using an attention mechanism. However the framework roots in video-level multi-label classification as the mechanism is designed for summarizing the video content. Therefore this method is not trivial to extend to action detection tasks. Most related to our research direction, Tirupattur et al. [7] introduced MLAD that can explore the class-temporal relations with a set of self-attention layers: an inter-class attention map for every time step and an inter-time attention map for every action class. However, the large number of attention maps leads to huge computational costs for long untrimmed videos and hence, limits the model to learn the discriminative relations among the action classes.

To tackle this, we propose CTRN, which is a graph-based model. Different from MLAD, CTRN explores the action class relation shared by all the time steps but in different temporal scales. This design enables CTRN to effectively handle both short-term and long-term action relations simultaneously. We also extend CTRN's framework in visual encoder level (i.e., THORN) to show the generalization of this framework. In the following sections, we

introduce the proposed networks.

5.3 Class Temporal Relational Network

In this section, we formulate the proposed end-to-end model Class-Temporal Relational Network (CTRN) for action detection. As shown in Fig. 5.3, our model is composed of four major components. The **Visual Encoder** encodes the video into a sequence of snippet-level spatio-temporal representation. This representation is fed to a Class-temporal Relational Network (CTRN) that predicts the action labels at each time instant. The sub-components in CTRN consist of the following: Firstly, a **Representation Transform Module**, which transforms the mixed visual representation into a class-wise representation. Secondly, a **Class-Temporal Module** explores the action class relations across different time steps and at different temporal resolutions. Finally, a **G-Classifier** which classifies the class-temporal features into action categories. Unlike previous binary classifiers [17, 18] that overlook the dependencies between the action classes, G-Classifier leverages the privilege class dependencies within the training data, thus improving the co-occurring action detection performance. In the following, we introduce these modules in details.

5.3.1 Visual Encoder

Similar to most action detection models [5, 17], our model processes the features on top of video snippet representations extracted from 2D/3D CNNs. In this work, we use spatio-temporal features extracted from RGB and Optical Flow (OF) I3D networks [22] to encode appearance and motion information respectively. Then, a video is divided into T non-overlapping snippets, each snippet consisting of 16 frames. The inputs to the RGB and Flow deep networks are either the color images or the corresponding OF frames of a snippet. We stack the snippet-level features along the temporal axis to form a $T \times D_1$ dimensional video representation, denoted as X. The action instances in X are always longer than a snippet and their visual representation mixes information of all action classes. As a result, X is not discriminative enough, and needs both temporal and class modelling. To this end, we develop the class-temporal relationship from the input representation X within CTRN which is described in the following. Note that the model architecture remains the same for both RGB and OF streams.

5.3.2 Representation Transform Module

The input X is first fed into the Representation Transform Module (RTM). The goal of this module is to transform the input to a class-specific representation and to lightweight the channel size to facilitate the following computation (see Fig. 5.4). In practice, RTM

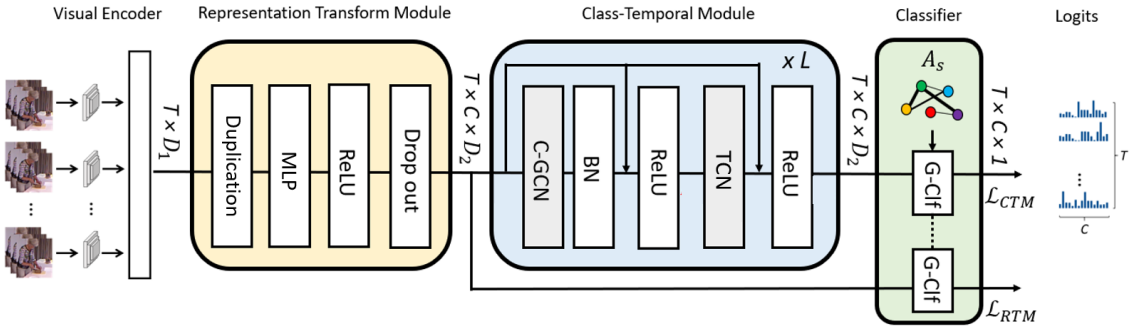


Figure 5.3: Overall structure. The model composed of a Visual Encoder, a Representation Transform Module, a Class-Temporal Module (with C-GCN and TCN) and a G-Classifier (i.e. G-Clf). Note: Two G-Clfs are sharing the weights.

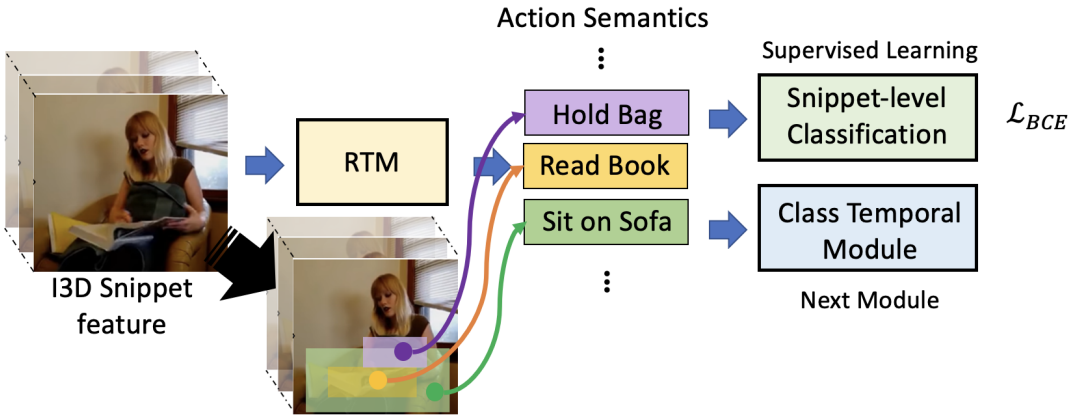


Figure 5.4: The RTM extracts the class-specific information of action from the I3D feature. The semantic extraction is supervised by the action occurrence of each snippet.

duplicates the input features C times into a new dimension representing the action classes followed by a channel-mixer MLP with non-linear activation and dropout. MLP is the linear transformation layer [176] to do the linear projection. The equation can be formulated as:

$$X'_i = ReLU(MLP(X)) \tag{5.1}$$

$$X' = DropOut([X'_1, X'_2, ...X'_C])) \tag{5.2}$$

where $X' \in \mathbb{R}^{T \times C \times D_2}$ is the output representation of RTM. $D_2 = \frac{D_1}{\beta}$ in which β is larger than 1 to shallow the channel size. In order to learn class-specific representation, we embed an auxiliary branch with a G-classifier that maps X' to the action labels (see Fig. 5.3). This transformed feature representation is further exploited to explore the class and temporal relations in the subsequent modules of the network. The computation flow is given in Fig. 5.5.

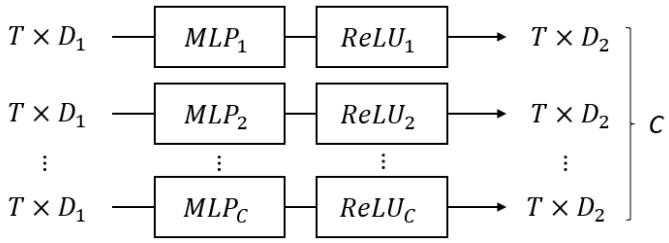


Figure 5.5: Computation flow of RTM.

5.3.3 Class-Temporal Modeling

The Class-Temporal Module (CTM) is the key component of CTRN that exploits the class-temporal relations of its input feature. Inspired by the recent success of Graph Convolutional Network (GCN) in relational reasoning [177, 15, 96, 178], we build this module with GCN. The objective of this component is to update the feature representations by propagating the information across different classes and across different time steps. For modelling the action class relations, we introduce a Class-GCN (C-GCN) layer while the traditional Temporal Convolutional Network (TCN) layer [5] is utilized to aggregate the temporal information. The combination of C-GCN and TCN enables CTM to capture the class semantic information along different temporal hierarchies. Thanks to the learnable graph structure, C-GCN is adaptive with the temporal scale set by TCN.

In the following, we first introduce how we map the feature representation to the graph structure and then we introduce the CTM components.

5.3.3.1 Representation-to-Graph Mapping

For GCN to process the action relations, the data is to be converted into a graphical structure. As we have transformed the representation into the class-specific format, thus each vertex of the graph represents an action class at a time step with an embedding vector belonging to $\mathbb{R}^{D_2}$. In total, the graph consists of $C \times T$ vertices whose topology is defined by an adjacency matrix (A_C). This matrix determines whether there are connections (i.e. relations) and its weights determine the intensity of the connections.

5.3.3.2 Class-GCN (C-GCN)

Class-GCN aims at performing the cross-class reasoning over the constructed graph representation. The relations between the many action instances are complex and are different across videos. Besides, multiple C-GCNs are stacked in CTM through which C-GCNs capture different levels of semantic information. Consequently, the graph adjacency A_C learns from the data itself for it to be adaptive across different temporal scales.

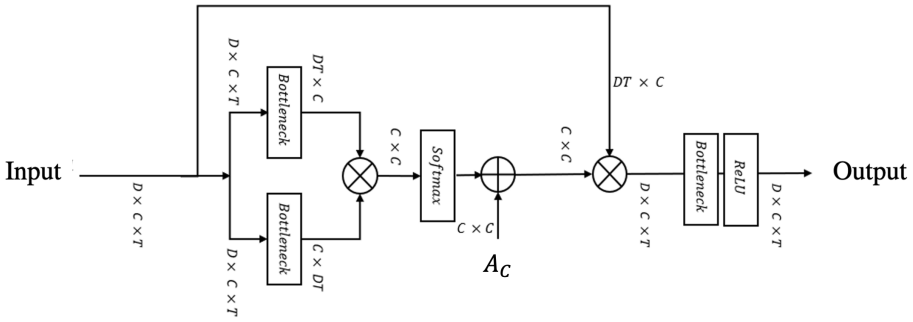


Figure 5.6: Computation flow of C-GCN.

In practice, $A_C \in \mathbb{R}^{C \times C}$ is parameterized and is optimized together with other parameters in the training process. Moreover, to differentiate the class relations owing to different videos, the adjacency matrix A_C learns the inter-dependencies among the classes using a self-attention mechanism. For this, the input feature $X_{C_{in}} \in \mathbb{R}^{D_2 \times T \times C}$ is first embedded using bottleneck convolutional layer (i.e. 1×1). After that, the output feature maps are rearranged into $\mathbb{R}^{D_2 T \times C}$ and $\mathbb{R}^{C \times D_2 T}$ followed by a matrix multiplication. The value of the resultant matrix is then normalized by a `softmax` activation. Now, the superimposed adjacency matrix A'_C can be formulated as:

$$A'_C = A_C + softmax(W_1^\top X_{C_{in}}^\top W_2 X_{C_{in}}) \tag{5.3}$$

where $X_{C_{in}}$ is the input of the C-GCN, and W_1 and W_2 are the weights of the bottleneck convolutions. Each value in this matrix can be seen as a soft edge between two vertices. The learned graph is shared across different time steps but unique for different layers and videos. This design choice can capture the inter-class dependencies in a video and makes C-GCN scalable across different temporal scales. Finally, we perform the graph convolutional operation with the formulation in [177]:

$$X_{C_{out}} = A'_C X_{C_{in}} W_3 \tag{5.4}$$

where $W_3 \in \mathbb{R}^{D_2 \times D_2}$ is the learnable weight matrix. The operation with A'_C and with W_3 represents the message passing and vertex feature updating, respectively. Finally, $X_{C_{out}}$ is rearranged to $\mathbb{R}^{D_2 \times T \times C}$. A computation flow of a C-GCN block is given in Fig. 5.6.

5.3.3.3 CTM Block

As shown in Fig. 5.3, there are L blocks in CTM, each block is composed of a C-GCN and a TCN layer along with batch normalization and non-linear activations. To stabilize the training, two residual connections are added in each block.

As mentioned earlier, TCN [5] aggregates the features across the temporal dimension while increasing the size of the temporal receptive field. In this work, we set a fixed kernel

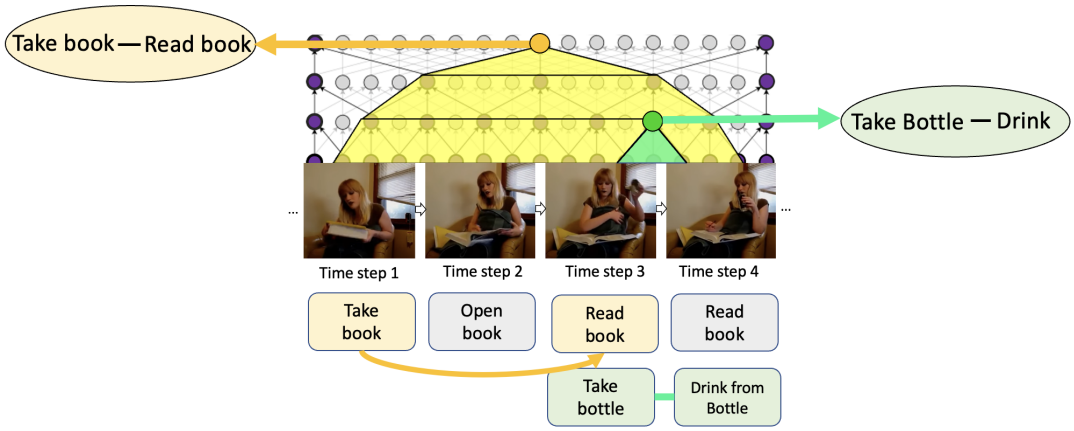


Figure 5.7: Thanks to the hierarchical structure of CTM, the Class-GCN can focus on short-term action-dependencies in lower blocks and long-term action dependencies in higher blocks.

size K for all the TCNs. Thanks to the hierarchical structure of CTM, C-GCN can focus on short-term action-dependencies in lower blocks and long-term action dependencies in higher blocks (See Fig. 5.7). The refined feature representation from the last block is fed into G-Classifier for the snippet-level classification.

Note that such a hierarchical structure is close to PDAN's pyramid structure that is introduced in Chapter 4.4. The main difference is that PDAN features the novel dilated attention layer and modelling only the dependencies across time (i.e., snippets). CTM utilises the standard temporal convolutional layers and explores the dependencies among extracted semantics (i.e., action classes) in multiple temporal levels.

5.3.4 G-Classifier

Finally, we introduce a graph-based G-Classifier to perform the final snippet-level classification. In action detection, multiple actions could happen simultaneously; thus, prior knowledge of inter-dependencies among different action classes can benefit in making precise predictions. To this end, inspired by [179, 180] in multi-label image recognition, we introduce a GCN-based classifier in action detection. Compared to the standard binary classifier for multi-label classification, G-Classifier has an additional message passing step between the potential co-occurring action pairs, thus improving the co-occurring action detection performance. Different to C-GCN, G-Classifier focuses only the actions that occur simultaneously (i.e., in a snippet-level feature).

In practice, we follow the similar computation process as [180]. Firstly, we compute the co-occurrence probabilities of all the action pairs in the training snippets. M_{ij} indicates the concurring times for action class C_i and C_j. Then, the conditional probability matrix

$P_{ij} = P(C_j|C_i)$ is given by:

$$P_{ij} = M_{ij}/N_i \tag{5.5}$$

where N_i indicates the occurrence times of C_i in training set, and $P_{ij} \in \mathbb{R}^{C\times C}$ indicates the probability of class C_j given that C_i occurs at the same time. In fine-grained action datasets, some rare co-occurrences may add noise for detecting other common actions, and the number of co-occurrences from training and test set may not be completely consistent. In this work, we perform a thresholding operation to binarize the conditional probability matrix to filter the noisy edges and make the classifier more robust to inconsistent action classes. If $P_{ij} \geq \theta$, $A_{S_{ij}}$ is assigned 1, otherwise 0, where θ is the threshold. The computed co-occurrence matrix A_S is a binary correlation matrix which in turn defines the adjacency matrix of the graph for G-Classifier. The feature of a node is computed by the weighted sum of its own features and the adjacent nodes' features. However, the binary correlation matrix may change the feature scale [177] and make the node feature over-smoothed [181]. To alleviate this problem, we normalized the A_S following the re-weighted scheme in [179]. Different to the learnable adjacency matrices in C-GCN, A_S is fixed during training. The formulation of this G-Classifier is given below [180]:

$$S = \sigma(A_S X^L W_S) \tag{5.6}$$

where S is the prediction score, σ is the `sigmoid` activation. X^L is the output feature from the last block of the Class-Temporal Module, and $W_S \in \mathbb{R}^{1\times D_2}$ are the learnable weights of the G-Classifier.

To learn the parameters, we optimize the multi-label binary cross-entropy loss with the prediction results from the RTM and CTM. The total objective is formulate as:

$$\mathcal{L}_{total} = \mathcal{L}_{CTM} + \alpha\mathcal{L}_{RTM} \tag{5.7}$$

where α is a weighting factor. Thus, by jointly optimizing both the entropy losses, the model learns the relevant action labels per segment along with learning the class-specific semantics across the Representation transform module.

5.3.5 Experiments

5.3.5.1 Datasets

To evaluate the capacity of the model for handling the complex fine-grained action relations in the video, we choose three densely labelled action detection datasets: Charades [114], TSU and MultiTHUMOS [13]. We follow the original settings of these datasets for action detection. By default, all these datasets are evaluated by the per-frame mAP.

5.3.5.2 Implementation

To have a fair comparison with previous works [7, 17], our network is built on top of I3D, where D_1 is 1024 and D_2 is 64. Dropout probability is 0.3. For CTM, we choose a 5-block (L) structure. For C-GCN, the adjacency matrix is initialized by 1 and normalized by columns. In TCN, the kernel size K is 9 and padding rate is 4. For G-Classifier, θ is set to 0.05. While learning the parameters, the weighting factor α is 1.2 and the random seed is fixed. We use Adam optimizer [162] with an initial learning rate of 0.001, and we scale it by a factor of 0.3 with a patience of 10 epochs. The network is trained on a 4-GPU machine for 300 epochs. For two-stream network, a mean pooling is performed between the prediction logits of the RGB and Flow streams.

5.3.5.3 Comparison with State-of-the-Art Methods

The proposed CTRN is compared with previous state-of-the-art methods on the Charades, TSU and MultiTHUMOS datasets in Table 5.1. Our proposed method outperforms current state-of-the-art methods on all three datasets. For example, +6.9% (relatively +37.5%) w.r.t. MLAD [7] on Charades while using only RGB. We then show the ability of CTRN capturing action co-occurrence, we evaluate with the action-conditional metric [7] in Table 5.2. Compared with state-of-the-art methods, our method achieves higher performance on all action-conditional metrics showing that CTRN effectively models action dependencies both within a time-step (i.e. co-occurring action, $\tau = 0$) and throughout time ($\tau > 0$).

To confirm the advancement of our method, we present further comparisons with MLAD. We compare the model efficiency and complexity. MLAD is about 2 times larger in parameters and 3.5 times larger in FLOPs than CTRN while processing the same batch of videos. Hence, our method is more lightweight and computationally efficient than MLAD. This is because MLAD predicts an inter-class attention map for every time step and predicts an inter-time attention map for every action class. For CTRN, we construct a temporal hierarchy structure. In each temporal scale, CTRN learns a single global class relational graph shared by all time steps. Therefore, CTRN is more lightweight and efficient.

5.3.5.4 Ablation Study

In Table 5.3, we study the complementation of the components in the proposed network on the Charades dataset. We first discuss how RTM leads to a better feature representation of the input spatio-temporal feature map from I3D. RTM is an essential pre-step before class-temporal modelling. Thanks to RTM that filters the class-specific feature, the model can slightly improve the detection performance (+0.5%). We then explore how the different

Model	Modality	Charades	TSU	MultiTHUMOS
R-C3D [81]	RGB	12.7	8.7	-
I3D + TAN [102]	RGB+OF	17.6	-	33.3
I3D + Superevent [17]	RGB	18.6	17.2	36.4
I3D + TGM [18]	RGB	20.6	26.7	37.2
I3D + TGM [18]	RGB+OF	21.5	-	44.3
I3D + TGM + Superevent [18]	RGB+OF	22.3	-	46.4
I3D + MLAD [7]	RGB	18.4	-	42.2
I3D + MLAD [7]	RGB+OF	22.9	-	49.6
I3D + CTRN	RGB	25.3	**33.5**	44.0
I3D + CTRN	RGB+OF	**27.8**	-	**51.2**

Table 5.1: Comparison with the State-of-the-art on three densely labelled datasets. The results are given in per-frame mAP (%). RGB +OF indicates the late fusion performance.

	$\tau = 0$				$\tau = 20$				$\tau = 40$			
	P_{AC}	R_{AC}	$F1_{AC}$	mAP_{AC}	P_{AC}	R_{AC}	$F1_{AC}$	mAP_{AC}	P_{AC}	R_{AC}	$F1_{AC}$	mAP_{AC}
I3D	14.3	1.3	2.1	15.2	12.7	1.9	2.9	21.4	14.9	2.0	3.1	20.3
CF	10.3	1.0	1.6	15.8	9.0	1.5	2.2	22.2	10.7	1.6	2.4	21.0
MLAD [7]	19.3	7.2	8.9	28.9	18.9	8.9	10.5	35.7	19.6	9.0	10.8	34.8
CTRN	**23.9**	**8.0**	**11.9**	**29.7**	**21.7**	**9.1**	**12.9**	**36.8**	**23.0**	**9.3**	**13.2**	**35.5**

Table 5.2: Evaluation on the Charades dataset using the action-conditional metric [7]. P_{AC} - Action-Conditional Precision, R_{AC} - Action-Conditional Recall, $F1_{AC}$ - Action-Conditional F1-Score, mAP_{AC} - Action-Conditional Mean Average Precision. τ indicates the temporal window size. $\tau = 0$ corresponds to the actions occuring at the same time.

components in CTM affects the action detection performance. We find that both C-GCN and TCN improve the performance w.r.t. a model with only RTM (+23.6, and 32.9% relatively). The action detection performance is further improved by the combination of both C-GCN and TCN, thus reflecting the complementary nature of both the operations. Finally, we study the performance with/without G-Classifier. With the proposed classifier, RTM and RTM+CTM further improve the action detection performance by +2.3% and +0.6% respectively. Note that for the baseline without G-Classifier, similar to the previous work [17], we utilize a 1×1 convolution as the classifier. These results show that the different components of CTRN contribute to the overall performance of our network.

5.3.5.5 Qualitative Study

In Fig. 5.6, we show the adjacency matrix of G-Classifier in Charades (157 classes), which provides the information of all the co-occurring action pairs with high probabilities. For example, *holding a vacuum* & *tiding something on the floor* and *fixing hair* & *watching in a mirror* are the actions always occur at the same time. Prior access to such privilege knowledge is crucial for detecting the co-occurring actions in the densely labelled videos.

In CTM, TCN is used to aggregate the temporal information which enables C-GCN to explore action relations at different temporal scales. To validate the usage of these layers, in Fig. 5.9, we visualize the learned adjacency matrix of C-GCN from three different

CTRN Components				Charades
RTM	C-GCN	TCN	G-Classifier	Per-frame mAP
×	×	×	×	15.6
✓	×	×	×	16.1
✓	✓	×	×	19.9
✓	×	✓	×	21.4
✓	✓	✓	×	24.7
✓	×	×	✓	18.4
✓	✓	✓	✓	**25.3**

Table 5.3: Ablation study on Charades dataset using only RGB.

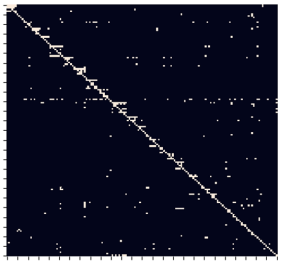


Figure 5.8: The adjacency matrix of the G-Classifier A_S.

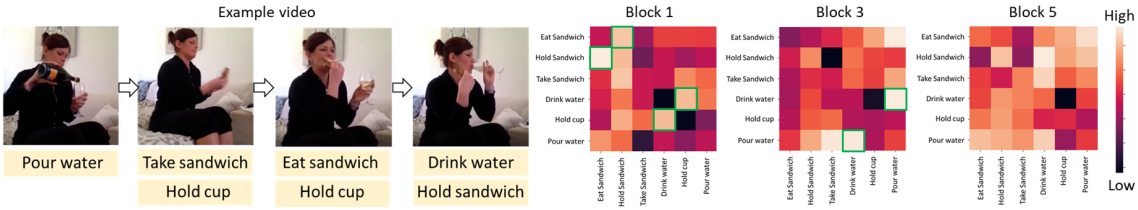


Figure 5.9: Visualization of the learned C-GCN adjacency matrix A'_C for different layers. Here, we visualize the 1^{st}, 3^{rd} and 5^{th} block's adjacency matrices. For simplicity, we provide only the relevant action classes in the example video.

blocks. We find that in Block 1, C-GCN focuses on capturing the contextual information pertaining to locally related action classes. For example, *eat sandwich* & *hold sandwich* and *drink water* & *hold cup* are always occurring closely in the video. Then we find that, Block 3 has increased the temporal receptive field, thus, C-GCN can capture the long-term dependencies between distant action classes. For example, *Pour water* and *Drink water*. Finally, Block 5 possess the largest receptive field where each local snippet feature contains the whole video information. Therefore, C-GCN in this block models all the potential action relations in the video, resulting in many activated links in the adjacency matrix.

5.3.5.6 Additional Studies

In the following, we provide more studies of CTRN. This includes the number of blocks, the design choice of the adjacency matrix in CTM, and results with different modalities.

Number of Blocks

We first explore the impact of the number of blocks (L) of CTM in CTRN. As mentioned in the proposed method, TCN is used to aggregate the temporal information. Thus, with more blocks, CTRN can model high level temporal information while expanding the scale across time for very long videos. Table 5.4 shows the results on Charades with different blocks, we find that CTRN achieves similar performance for 5 and 6 blocks. Thus, 5-block is sufficient for encoding the temporal information in complex untrimmed videos.

#Blocks	4	5	6
Performance (%)	24.2	25.3	25.3

Table 5.4: Study on number of blocks L of CTM in CTRN. We evaluate on Charades dataset for action detection using only RGB.

Adjacency Matrix A'_C

As mentioned earlier, C-GCN's graph is composed of a learnable adjacency matrix A_C and an attention mask which is superimposed on the former. Here we further analyze that both the components are complementary. In Table 5.5, we find that the performance declines in the absence of either A_C or the attention mask in C-GCN, reflecting both components are crucial for learning the the graph structure.

Adjacency Matrix A_C	Attention Mask	mAP (%)
×	×	21.4*
✓	×	24.3
×	✓	24.5
✓	✓	25.3

Table 5.5: Study on adjacency matrix in C-GCN. We evaluate on Charades dataset for action detection using only RGB. * indicates the results of CTM w/o C-GCN but only a TCN.

Modalities

Our model can be used with both RGB and Optical Flow (OF). Here, we provide the results with RGB and OF. For a fair comparison, similar to the previous works [17, 18], we fuse the two modalities through a late-fusion of the logits. From Table 5.6, we find that: (1) For sport actions in MultiTHUMOS, Optical Flow stream yields better performance than RGB stream (+4.3%). (2) For object-based actions with low motion in Charades, RGB stream achieves better performance (+3.8% w.r.t. Optical Flow stream), which indicates that RGB can better model the object appearance information, especially for low motion frames.

5.4 Temporal Human Object Relation Network

CTRN is a two steps method, which is built on top of pre-extracted flattened 1-dimensional features. The dissociation between the visual encoder and temporal module makes the model overlook the appearance and spatial information in the video. To validate that the proposed mechanism can also perform effective semantic reasoning on the spatio-temporal representation, we propose Temporal Human-Object Relation Network (THORN). This model can leverage such semantic relation modeling mechanism for **ac-**

Modalities	RGB	OF	RGB+OF
Charades	25.3	20.3	27.8
MultiTHUMOS	44.0	47.5	51.2

Table 5.6: Study on RGB and optical flow. RGB+OF indicates the late fusion.

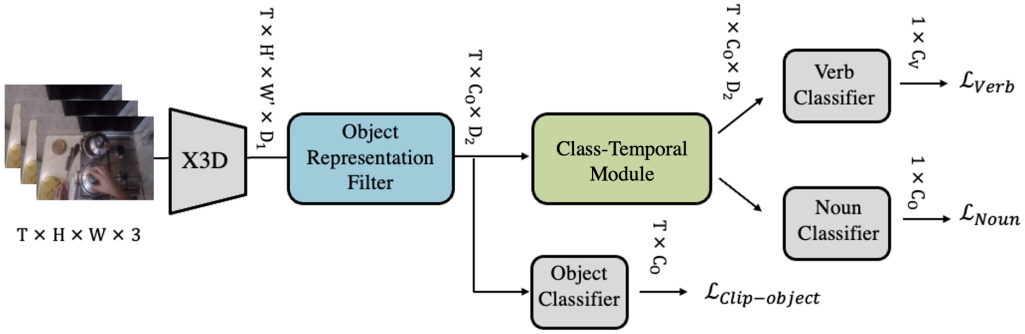


Figure 5.10: THORN architecture contains three main components: (1) a **Visual encoder** (i.e., X3D) encodes the input RGB clip into a primary spatio-temporal representation. (2) The obtained representation is fed to the **Object Representation Filter**, which maps the previous representation into object-class representation. To ensure a discriminative object representation, an object classifier is added on top of the object-class representation. This classifier is trained with the pseudo-object ground truth provided by an object detector. (3) The object-class representation is also sent to the **Class-Temporal Module** to model the temporal-object relation in a dissociated manner. Finally, two classifiers are used to predict the verbs and nouns relevant to the action.

tion recognition to extract detailed action semantics (e.g. object, verb) in an end-to-end manner (as shown in Fig. 5.2). THORN features a similar semantic reasoning framework as CTRN. Firstly, a **3D Visual Encoder** which encodes the video into a spatio-temporal embedding. Then, the previously extracted embeddings are passed to the **Object Representation Filter** (ORF). This filter extracts class-specific features. Finally, the **Class-Temporal Module** computes the relation between the different objects to predict the "verb" of action. This module also refines the node representation to predict the "noun" of action. Fig. 5.10 provides an overview of the model.

In the following, we detail the architecture of THORN, especially, the difference between the THORN and CTRN.

5.4.1 Visual Encoder

Different from CTRN, the visual encoder in THRON is trained end-to-end with the following modules, thus can better extract the primary spatial-temporal representation. In this work, we utilized X3D [57] as the visual encoder. The lightweight property of X3D can help to train the *Visual Encoder* jointly with the proposed modules. In practice, the input to

the visual encoder is a series of frames. Different from CTRN, the visual encoder outputs a spatio-temporal representation F of shape $T \times H' \times W' \times D_1$, where: $H' = W' = 7$, $D_1 = 432$, while T is the same as the input. This embedding carries both spatial and temporal information. The spatial information is important, as it provides object-related information, such as its appearance, shape and position (e.g. drawers usually appear at the bottom of the image). That is why instead of using the X3D final output of shape $T \times 2048$ to construct our nodes, we use a finer spatial representation of shape $T \times 7 \times 7 \times 432$, making nodes of our graph contain more and finer information about the objects.

5.4.2 Object Representation Filter

Our main objective through THORN is to have object-based reasoning. This objective relies on obtaining effective object representation in scene representations. Therefore, we developed the *Object Representation Filter* module, capable of extracting semantic representation specific to each object class from the previous overall representation. This module serves as a filter to obtain the object-specific representation from the output of the visual encoder. Note that this module is similar to Representation Transform Module (see Sec. 5.3.2) in CTRN. The difference mainly lies in that the semantics are extracted from spatio-temporal feature maps and the semantics represent objects.

In practice, firstly, we reshape the representation F from the visual encoder to shape $T \times H'W'D_1$. After that, we duplicate the reshaped features F' for C_o times, where C_o indicates the number of object classes in the dataset. For each class, we use a channel-mixer MLP (i.e., linear transformation layer), followed by non-linear activation and dropout. We argue that each MLP layer learns to extract features specific to a certain object class. The equations in this module can be formulated as:

$$F'_i = ReLU(MLP(F)) \tag{5.8}$$

$$F' = DropOut([F'_1, F'_2, F'_3 ...., F'_{C_o}]) \tag{5.9}$$

where $F' \in \mathbb{R}^{T \times C_o \times D_2}$. D_2 is smaller than D_1 to shallow the channel size. Here $F'' \in \mathbb{R}^{T \times C_o \times 1}$. To ensure the object-specific representation, we add a frame-level object classifier on F''.

$$F'' = ReLU(MLP(F')) \tag{5.10}$$

As the frame-level object label is not provided by the dataset, the object classifier is trained with the pseudo label provided by an object detector (i.e. Fast-RCNN [85]). Note that, we are not relying on the location information of the object (i.e., only the existing object categories). In the video, multiple objects can appear in a frame, thus, we train the object

classifier with binary cross-entropy loss: $\mathcal{L}_{clip-objects}$. Finally the ORF module outputs a representation for each object-class. We still need to correlate and refine these object representations to explore their interactions and model the actions.

5.4.3 Class-Temporal Module

In order to learn the relations between the extracted object semantics, THORN has a similar graph reasoning module as CTRN. This module sequentially stacks graph convolutional layers and temporal convolutional layers to model the semantic-temporal relation in the video clip. The architecture of this model has been provided in Sec. 5.3.3. This module can help to refine the node representation from the related nodes and can also capture the correlation between the nodes.

5.4.4 Prediction

The predictions are based on the learned nodes and adjacency matrix. In the evaluated datasets, fine-grained actions are composed of verbs and nouns. For this reason, we use the learned adjacency matrix for predicting the verb and the learned nodes for noun prediction. This is because the adjacency carries more information about how different objects interact with each others, while the nodes carry a refined object representations, after been processed through the Class-Temporal Module. As shown in Fig. 5.10, the output of CTM is sent to two classifiers: one projecting the output representation from $\mathbb{R}^{D_2 \times C_o}$ to $\mathbb{R}^{1 \times C_o}$, and the other classifier projecting A'_{C_o} from $\mathbb{R}^{C_o \times C_o}$ into $\mathbb{R}^{1 \times C_v}$, where C_o and C_v stand for the number of object classes and verb classes respectively.
As shown in Fig. 5.10, our objective is a sum of three losses and can be formulated as :

$$\mathcal{L}_{total} = \mathcal{L}_{verbs} + \mathcal{L}_{nouns} + \mathcal{L}_{clip-objects} \tag{5.11}$$

Where $\mathcal{L}_{verbs}$ and $\mathcal{L}_{nouns}$ are the negative log-likelihood losses (since each action is composed of one verb and one noun). As described earlier, the $\mathcal{L}_{clip-objects}$ is the binary cross-entropy loss to ensure the semantic of the object representation.

5.4.5 Experiments

5.4.5.1 Dataset

We evaluate our model on two of the largest and challenging datasets for first-view and human-object interaction action recognition. **EPIC-KITCHENS 55**[1] and **EGTEA**

Gaze+ [115]. In both datasets, each action is a combination of a verb and a noun. Actions are relevant to different steps of preparing food (e.g. *cleaning the kitchen*, *cutting vegetables*, *preparing table*).

5.4.5.2 Implementation

We implement our method using X3D as the visual encoder where $D_1 = 432$, $H' = W' = 7$ and D_2 is 128. We input a clip of 16 RGB frames for EPIC-KITCHENS and 25 frames for EGTEA Gaze+. We use a dropout probability of 0.3. For the Class-Temporal Module, N_{Block} is 5 blocks. We utilise a kernel size of 9 for temporal convolution in Class-Temporal Module. In training phase, we utilized Adam [162] to optimize the model with an initial learning rate of 0.00005. We scaled the learning rate by a factor of 0.1 with the patience of 5 epochs. The network was trained on a 4-GPU machine for 30 epochs. We evaluated our model using Top-1 and Top-5 accuracy on verbs and nouns for EPIC-KITCHENS, while for EGTEA Gaze+ we evaluated directly on actions using top 1 accuracy.

5.4.5.3 Ablation Study

In this section, we validate our model design for the modules in the THORN. The evaluation is conducted on the EPIC-KITCHENS dataset. We propose different settings and see how each setting can improve the performance. The results are shown in table 5.7

Firstly, we compare our baseline model X3D with THORN. Note that, in THORN, the graph nodes can be constructed either using the output of the last layer of X3D (temporal nodes) or using its intermediate layer (spatio-temporal nodes). Here, we first compared X3D with THORN (temporal nodes), i.e., we construct the nodes by the features in shape of $T \times 2048$. In this setting, nodes would serve to predict both verbs and nouns. In this scenario, we improve nouns prediction by **+5.6%**, while, the verbs accuracy increased by **+9.3%** . Proving the importance of the cross-object reasoning, compared to only capturing visual information from 3D-CNNs.

Secondly, we study the importance of the adjacency matrix for predicting the verbs. To do so, we use the adjacency matrix (ADJ-matrix) to predict verbs, while keeping the nodes to predict the nouns. In this setting, the verb prediction improves by **+4.5%** compared to the previous setting and by **+13.8%** to the baseline X3D. This is because the adjacency matrix captures the object interaction, hence, it is more suitable for verb prediction.

Thirdly, we study the effect of changing the temporal nodes with the spatio-temporal nodes. Spatio-temporal nodes are the nodes constructed by the middle layer of X3D which contains the spatial information $T \times 7 \times 7 \times 432$. With spatio-temporal nodes, THORN improves **+1.8%** on nouns. This is because, with spatial dimensions, the ORF can better capture the object relative locations and the size of the object, then embed them in the

	Verbs		Nouns		Actions	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
X3D	46.5	79.8	34.3	65.3	21.0	38.7
THORN/temporal nodes	55.8	82.9	39.9	66.4	26.8	44.0
THORN/temporal nodes + ADJ-matrix	60.3	86.0	41.1	66.9	30.1	47.3
THORN/spatio-temporal nodes + ADJ-matrix	61.0	85.9	42.9	67.9	30.5	47.5

Table 5.7: Ablation study on different settings. This evaluation is on EPIC-KITCHENS dataset. Temporal nodes means using the final output of X3D of size $T \times 2048$ to create nodes, while spatio-temporal nodes means using a mid layer of size $T \times 7 \times 7 \times 432$ with more spatial information. Finally ADJ-matrix stands for using the adjacency matrix for predicting the verbs instead of using only nodes for nouns and verbs.

Faster-RCNN	THORN	Nouns
✓	×	31.5
×	✓	32.8
✓	✓	42.9

Table 5.8: Ablation study on fusing the scores of THORN with the scores from the object detector (Faster RCNN). This evaluation is on EPIC-KITCHENS dataset. Fusing both scores brings significant improvement on top-1 accuracy. For the object detector, we use an average pooling on all the video clip frames object detection scores and add a thresh-hold of 0.3

node representation. As a result, the noun accuracy improves. This setting also brings **+0.7%** improvement on verbs.
Our overall architecture obtains **+13.8%** improvements on verbs and **+8.6%** on nouns w.r.t. vanilla X3D. This reflects the effectiveness of our proposed modules in THORN and how an object-centric method can improve results on human-object interaction actions.

We then study the components for predicting the nouns in our model. In table 5.8, we show that fusing scores of object detection and the scores obtained by the THORN nodes representation works better than using only one of them. We also find that predictions using only our model are better than the object detector itself. This shows that our model can refine the objects represented by the other objects (nodes) using our graph-based module.

5.4.5.4 Comparison with the State-of-the-Art

We then compare our proposed method with the state-of-the-art methods on EPIC-KITCHENS and EGTEA Gaze+ in table 5.9 and 5.10.

In Table 5.9, we compare our results with the state-of-the-art methods. Among these methods, Long Features Bank (LFB) [183] proposes to use global as well as local features for action recognition. To do so, they extract features on both clip and video levels, and

Model	Obj	RGB	Flow	Audio	Verbs	Nouns	Actions
					top1	top1	top1
Baradel[182]	×	✓	×	✓	40.9	-	-
3D-CNN [22]	×	✓	×	×	49.8	26.1	19.0
STO[183]	✓	✓	×	×	51.0	26.6	19.5
LFB[183]	✓	✓	×	×	52.6	31.5	22.8
AssembleNET++ ODF+SDF[184]	✓	✓	✓	×	60.0	37.1	25.2
THORN	✓	✓	×	×	**61.0**	**42.9**	**30.5**

Table 5.9: Comparing THORN model with other state-of-the-art methods on the validation set. Even though some of these comparisons are not fair since these models are using multi-modalities, we still hold the overall best accuracy, which shows the strength of our model

	Two-stream	I3D [22]	TSN [79]	EGO-RNN [185]	LSTA [186]	SAP [187]	**THORN**
ACC %	43.8	54.2	58.0	62.1	62.0	64.1	**67.5**

Table 5.10: Comparing THORN model with other state-of-the-art methods on EGTEA Gaze+ split1. We hold the best accuracy on actions

combine them to have a better understanding of the scene. Nevertheless, this method still lacks accuracy for the objects. Moreover, LFB is a two step method which trains separately an object and verb recognizer modules. For our THORN, we train a single model for predicting both entities. As a result, we have a **+8.5%** improvement on top 1 nouns and a **+4.9%** w.r.t. LFB on action recognition.

Our method achieves the overall best performance. We claim that AssembleNet++ utilizes additional modality such as optical flow in both training and inference time. Even though, we still have the lead in top 1 accuracy for the verbs, nouns and actions, which proves again that having an object-centric and specific reasoning on object interactions is a key solution for having a better action recognition on HOI datasets. Finally, our results prove that using only RGB with an object-centric model achieves better or similar results compared to methods relying on heavy multi-modality reasoning.

In table 5.10, we compare our method with the state-of-the-art on EGTEA Gaze+ dataset. We have the best accuracy w.r.t. the others methods, which shows the generalization and robustness of our model on actions of HOI.

To sum up, compared to other methods, ours is lightly weighted as we use X3D, while other methods rely on heavy 3D-CNNs such as I3D. THORN is trained jointly on nouns and verbs as opposed to other methods such as LFB [183], and we only need RGB frames and pseudo object labels per frame.

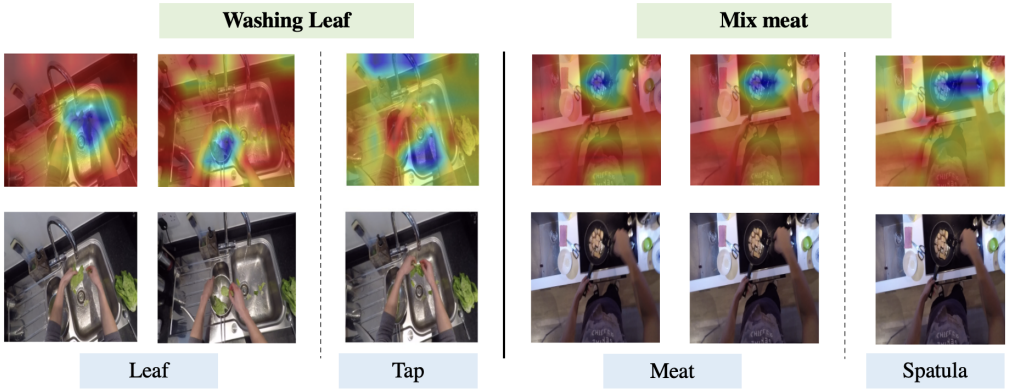


Figure 5.11: Visualization of the Class Activation Mapping for the object extractors' weights. The video name is highlighted in green and the extracted object is highlighted in blue.

Figure 5.12: Visualization of the learned cross-object relations. For the video "*washing knife*": the sampled frames is shown on the left, and the learned adjacency matrix of the graph convolution module is given on the right.

5.4.5.5 Qualitative Study

Firstly, we analyse if the Object Representation Filter can focus on the action related object region, and if it is robust across frames. In figure 5.11, we show two example videos and their class activation map [188] of the object classifier in the object representation filter. For video "*washing leaf*", we find that the object region can be extracted effectively with the proposed module, e.g. object "leaf" and "tap". Also the object are extracted robust across different frames (e.g., "leaf" across two frames). Similar observation can be found in video "*mix meat*".

Secondly, we visualize the learned adjacency matrix. As shown in figure 5.12, the subject is *washing knife*. The graph module highlights the correlation between the action relevant objects, i.e., object "knife" and object "water". Therefore, THORN is able to collect high inter-class relation to recognize the right "verb" and its relevant "objects". Moreover, the irrelevant classes such as "fish", "tap" and "sponge" is not interactive with the other

objects in this example video.

5.5 Conclusion

In this chapter, we propose a generic framework to enhance video representation by modelling the relationships between the different semantics in a video. There are two main steps for this framework: (1) Extraction of the semantic representation from the video. This extraction relies on a series of binary classifiers, each indicates whether a certain semantic exists in the frame or not. (2) After having the semantic representation, we model the relations across different semantics to have a refined video representation. The learned representation is then sent to the prediction head for the objective task.

We have evaluated the effectiveness of our method on two principal components in action detection framework (visual encoder and temporal module). For the visual encoder, we propose THORN, which refines the spatio-temporal representation by modelling inter-object relationships in each video clip. THORN can better represent fine-grained actions relevant to objects than the vanilla visual encoder. For the temporal module, we propose CTRN. This neural network enhances the temporal modelling by modelling the inter-action relationships in untrimmed videos so that CTRN can improve the detection of a series of correlated actions in the videos.

Although this chapter focused on modelling the action-action and object-object interactions of the videos, the framework developed in this context goes beyond this dissertation and can be generalized to other types of semantic relations (e.g. subject-subject interactions for group activity detection).

Chapter 6

Multi-Modal Representation Learning for Action Detection

In this chapter, instead of understanding video using a single modality, we propose two methods that can effectively and efficiently learn the multi-modality video representation. The fusion-based method AGNet is the proposed baseline method for the Toyota Smarthome Untrimmed dataset [43], which utilizes an additional modality to generate the attention weights at multiple temporal scales for improving action detection performance. The distillation-based method has been published in IEEE/CVF International Conference on Computer Vision (ICCV) [49] in 2021. This method encourages the RGB stream to mimic the representation of the additional modality stream in the training phase and avoids using the additional modality at inference time.

6.1 Introduction

Video can be captured or represented in different modalities, such as RGB, optical flow, 3D Pose, etc.. Each modality gives a view of the video which emphasizes an aspect of the information in the video, thus the modalities are usually complementary to each other. For example, RGB focuses more on the appearance of the objects, while optical flow gives more attention to the motion in the video. Thanks to this complimentary nature between modalities, learning representation of different modalities has become an effective manner to represent the video content [42], especially for scenarios requiring high precision. To this end, in the following, we study how to model compact untrimmed video representation by multiple modalities. Since RGB is the modality that contains the greatest amount of information, in this chapter we focus on how to infuse information of other modalities into the RGB branch.

In previous works, to combine multiple modalities, a typical setting, called two-stream

network [42], consists in combining RGB with additional modalities like optical flow [17, 21] or 3D poses [189, 55] to take into account the complementary nature of modalities. To further benefit from multiple modalities, firstly, we learn the multi-modal representation in a fusion manner. In this direction, we propose **Attention Guided Network (AGNet)**, which builds upon the existing temporal model: SSTCN [19]. This network has two input modality branches (e.g. RGB and 3D Poses). The main branch inputs the RGB videos and the attention branch inputs the additional modality. Similar to PDAN, each branch consists of five blocks and each block represents a temporal level. There are attention blocks between the two branches at each temporal level. More specifically, the attention block generates the temporal attention map at each temporal level from the additional modal stream to guide the RGB stream to predict more precise action boundaries. AGNet is proposed as the baseline method in Toyota Smarthome Untrimmed dataset and we evaluate AGNet with RGB and 3D Pose in the datasets. We show that AGNet efficiently infuses the additional modality information into the RGB branch.

Two-stream architecture can effectively combine different modalities and has become a typical setting in video understanding tasks. However, using such setting is contingent upon the availability of multiple modalities and of expensive processing resources. The cost of computing additional modalities could be prohibitive, especially for long untrimmed videos. These constraints limit the usage of multi-modal action detection methods for real-world applications.

Previous studies [190, 191] have shown that cross-modal Knowledge Distillation (KD) is an effective mechanism to avoid the computation of the additional modalities during test time, while preserving the complementary information from the additional modalities. However, most previous works [192, 193, 194] in the video understanding domain have investigated solely the classification of short trimmed videos. In these works, each video corresponds to a single action and the distillation framework infuses the aggregated knowledge of an action instance from one modality into another. Contrary to trimmed videos, untrimmed ones contain rich sequential knowledge with complex temporal relations. Untrimmed videos in real-world scenarios tend to have cluttered background and multiple correlated actions either in sequence [64] or in parallel [13, 2]. Therefore, distillation mechanisms tailored for classification tasks and extended for detection tasks lack in capturing fine-grained details along the temporal dimension. Now the question remains, what should be the right strategy to distillate cross-modal knowledge for action detection in untrimmed videos?

In this work, we propose a distillation framework to combine cross-modal information for detecting actions with high precision and minimal resource. The goal is to reach the two-stream performance while using only the RGB stream at inference time. The proposed distillation framework consists of a traditional teacher-student network archi-

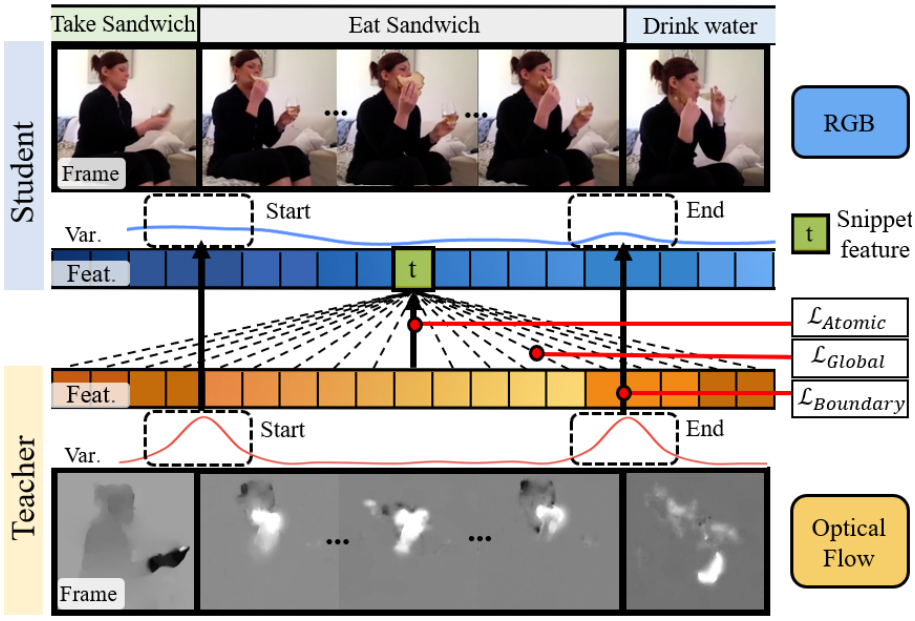


Figure 6.1: Proposed cross-modal distillation framework for action detection. Our distillation framework is composed of three loss terms corresponding to different types of knowledge to transfer across modalities. $\mathcal{L}_{Atomic}$: Atomic KD loss; $\mathcal{L}_{Global}$: Global Contextual Relation loss; $\mathcal{L}_{Boundary}$: Boundary Saliency loss.

tecture which operates in a Seq2Seq fashion [18, 102], thanks to three new distillation losses dedicated to the action detection task as illustrated in Fig. 6.1. The first loss in our formulation is the Atomic KD loss, which enables the RGB student network to mimic the feature representation of every individual snippet from the teacher network in a contrastive manner. This loss-term extends the cross-modal KD mechanism designed for the classification tasks to the temporal domain [195], by transferring the knowledge only between one-to-one corresponding snippets of different modalities. As a snippet is often shorter than the action instance in an untrimmed video, this loss encourages a transfer of sub-representation [119] of the action, for example, "*raising arm*" in the "*drinking*" action. Here, such sub-representation *w.r.t.* the entire video corresponds to an atomic piece of knowledge within the complete action feature distribution. However, an untrimmed video is composed of a sequence of snippets, distilling only the atomic representation is not sufficient for learning discriminative action representations. Thus, distillation mechanisms dedicated to represent specifically an action within an untrimmed video are required.

We therefore introduce two loss-terms for sequence-level KD so as to transfer the cross-snippet relations between different modalities. Firstly, we propose a Global Contextual Relation loss to transfer the contextual information of the sequence between modalities. In our work, contextual information is defined as the embedding of the correlation be-

tween all the snippet features. Thanks to this loss term, every student snippet feature can learn in the latent space from all the correlated teacher snippets within the untrimmed video (Fig. 6.1). With this loss-term, detecting one action in a snippet can benefit from the information in the correlated snippets (corresponding to related actions, e.g. *take and eat sandwich*) across modalities, resulting in better action detection performance. Secondly, we propose another KD loss to distillate the boundary saliency from the teacher to RGB student network, dubbed Boundary Saliency loss. This ensures a more precise action boundary detection of the RGB student which is prone to imprecise action boundary detection due to weak temporal signals. In an untrimmed video, the start and end moments of the action are usually more salient than other parts [196] (see Fig. 6.1). Intuitively, the feature variation across consecutive snippets in the video can reflect such saliency of the action boundaries. Therefore, learning this variation from a modality that can better capture the human movement (e.g. optical flow, 3D poses) which encourages the RGB stream representation to be more sensitive to the action boundaries.

Contributions: In this chapter, based on multi-layer temporal convolution networks, we propose two ways to leverage multiple modalities for action detection task. We first introduce AGNet which leverages an additional modality to generate the temporal region of interest (t-ROI) for the action instance in multiple temporal levels. With these generated attention masks, AGNet can effectively detect the actions in the video. Secondly, following a similar structure, we take a step towards the cross-modal KD for action detection. More specifically, we build a Seq2Seq KD framework for action detection with a novel formulation. This formulation consists of an atomic-level KD loss and two sequence-level KD losses. The three loss terms in our formulation are jointly optimized in an end-to-end fashion. To the best of our knowledge, we are the first to propose a formulation containing sequential KD loss for the action detection task.

6.2 Related Work

In this section, we briefly review the methods for combining multi-modalities and methods for cross-modal distillation in action understanding.

6.2.1 Combining Modalities

With the prevalence of RGB-D sensors, multi-modal video data have become more available for human action recognition and detection. Combining the advantages of privileged modalities in order to make use of their complementary discriminative power has been exploited widely in action recognition domain. Two-stream architectures [42, 197, 22] that learn separate features from optical flow and RGB modalities, outperform single modal-

ity approaches. Towards this direction, Ryoo et al. [198, 199] have proposed a Neural Search Architecture (NAS) to combine both RGB and Optical flow streams. In contrast to these methods, two complementary strategies are adopted to combine RGB and pose modalities. One is fusion of both modalities in feature space [200, 201, 202, 203]. However, these modalities are heterogeneous and must be processed by different kinds of networks to show their effectiveness. Combining these heterogeneous features from different modalities through feature/score fusion introduce noise resulting in a downgraded action recognition performance [204]. The second is Pose-driven attention mechanisms to guide the RGB cues for action recognition as in [205, 206, 50]. In [206, 205], the pose driven attention networks implemented through LSTMs, focus on the salient image features and the key frames. Then, with the success of 3D CNNs, 3D poses have been exploited to compute the attention weights of a spatio-temporal feature map. Then, authors in [50] have proposed a more general spatial and temporal attention mechanism in a dissociated manner. But all the above methods have the following drawbacks: (i) there is no accurate correspondence between the 3D poses and the RGB cues in the process of computing the attention weights; (ii) the attention sub-networks neglect the topology of the human body while computing the attention weights; (iii) the attention weights in provide identical spatial attention along the video. As a result, action pairs with similar appearance like *jumping* and *hopping* are mis-classified. Therefore, Das et al. [55] propose a new spatial embedding to enforce the correspondences between RGB and 3D poses which has been missing in the state-of-the-art methods. The embedding is built upon an end-to-end learnable attention network. The attention network considers the human topology to better activate the relevant body joints for computing the attention weights. Recently, Duan et al. [207] propose to leverage a 3D heatmap stack instead of a graph sequence as the base representation of human skeletons. After that, a Slow-Fast [208] fashioned two-stream network is utilized to model the spatio-temporal relation jointly using RGB and Pose.

However, all the above approaches are designed for action recognition only. The additional modalities should provide clues in long-term temporal modelling which is missing in the short video clip. In action detection, previous methods utilizes the multiple modalities either in the early phase [7, 32] (i.e., input level) or late fusion [17, 18] (i.e., output prediction score level). There are only a few methods [209, 210, 211] study the multi-modal fusion in the feature-level for action detection task and all the methods are designed for the combination of audio and RGB. Consequently, we propose a multi-modal action detection baseline method that enhances the RGB stream by the additional modality-driven attention mechanism at the feature-level. The additional modality can be modalities such as 3D poses or optical flow. Different from the previous methods, our baseline method can generate the temporal region of interest in multiple temporal scales.

6.2.2 Knowledge Distillation

The primary goal of Knowledge Distillation (KD) is to distill the information of a model learned from a teacher network into a student network. Many KD studies [212, 213, 214, 215, 216] explored transferring the knowledge from large complex models to small simpler models, i.e. model compression. In this work, we focus on cross-modal KD, where the difference between the teacher and student models mostly relies on input modalities rather than network architectures. In the video domain, Garcia et al. [193, 194] developed a distillation framework for action classification with a four-step process that hallucinates depth features into RGB frames. Similarly, MARS [192] trains a RGB network in a single step, by back-propagating a linear combination of a OF distillation and classification losses through the entire network. Recently, Luo et al. [195] proposed a Graph Distillation (GD) method that can be applied to the action detection task. This method utilizes sliding windows to process untrimmed videos and distillates the knowledge of every window by minimizing the cosine distance in a mutual learning manner. GD aims at exploiting the privileged modalities and thus relies on a significant number of modalities. In contrast, our framework aims at effectively performing the distillation from the available modalities. Moreover, GD transfers knowledge only between the corresponding snippets (i.e. window), but does not consider the relations across snippets in the distillation, which is critical for handling a sequence of actions. Thus, to better tackle distillation for action detection, in this chapter, we introduce two sequence-level distillation loss terms to transfer the long-range temporal knowledge for action detection. Thanks to our proposed methods, the network can be effective even with few additional modalities.

Fairly recent, given the advance in Transformer architecture, some methods introduce the class or distillation token in Transformer to fuse knowledge of different modalities [217, 218] or distillate knowledge across modalities [219, 220]. How to leverage the Transformer to efficiently learn multi-modal representation to benefit action detection task is our future work.

6.3 Attention-Guided Network (AGNet)

In this section, we introduce an end-to-end baseline method: Attention-Guided Network (AGNet) for action detection which is built upon temporal convolutional networks [5]. An overview of the AGNet is shown in Fig. 6.2. The input is the encoding of a video. The AGNet has two principal components: a stacked dilated temporal convolution network (SD-TCN) and an attention module. In this work, the input to the base-network is always the RGB frames. For attention module, the input is another modality, such as 3D human poses or optical flow. For simplicity, in the following, we consider the 3D poses

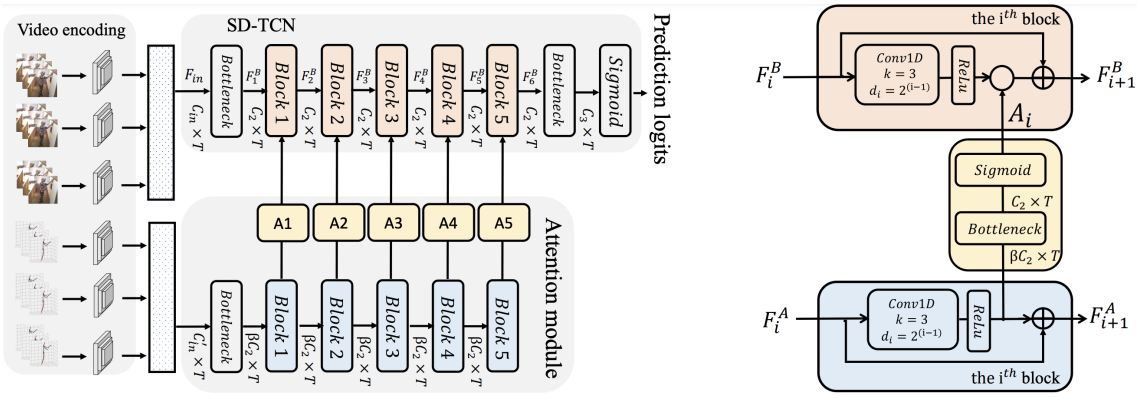


Figure 6.2: On the left, we present the overview of the AGNet. In this figure, `Bottleneck` indicates the 1D convolution that processes the features across time and which kernel size is 1. On the right, we present the computation flow for one block. In each block, `k` is the kernel size and `d` is the dilation.

as the input to the attention module. The SD-TCN and the attention module have both a 5-block structure. These blocks have temporal convolution with increased dilation rates setting, thus the receptive field increases exponentially. The lower-blocks have smaller reception fields while the higher blocks have larger receptive fields. For every block, the pose-attention module generates an attention mask that represents the temporal saliency of human actions in a video. The main contribution is the attention module, which utilizes 3D poses to generate the attention weights at multiple temporal scales. We believe that 3D poses are complementary to the RGB modality as they help filtering the irrelevant context in the RGB frames and providing more weight to the pertinent frames of the video. Below, we detail the video encoding and the model structure of AGNet.

6.3.1 Video Encoding

Similar to most action detection models [17, 18, 5], our model processes the encoding of video segments. In this work, we use state-of-the-art convolution model (i.e. 2D+T CNN or (2+1) D+T GCN) to extract appearance features in the video. The RGB encoding is extracted by a CNN such as Inception [73] or I3D [22]. The pose encoding is extracted by a GCN such as ST-GCN or 2s-AGCN [67]. We fine-tune the 3D convolution model on the training set of TSU to better model the spatial information in this dataset.

Training: To fine-tune the feature extraction model, firstly, we divide the video into 100-frame-long non-overlapping segments. For the RGB modality, to tackle the camera framing challenge, we apply SSD [142] to extract the human crops (i.e. bounding box) of the subject, and resize the crop into 224×224. For 3D poses, the subject would always be re-projected at the center of the screen with a fixed scale by using [135]. We then train the

classification model [22, 67] with the uni-sampled 16 frames for each segment. For the RGB modality, we flip all the images in each segment with a probability of 0.5. The inputs to the RGB or 3D pose convolution model are the RGB human crops and corresponding skeleton of a segment respectively. We optimize the multi-label binary cross-entropy loss [108] to learn the parameters.

Feature extraction: To extract the features, a video is divided into T non-overlapping segments, each segment consisting of 16 frames. These segments of RGB human crops or pose sequences are sent to the fine-tuned spatio-temporal model to extract the segment representation. We stack the segment-level features along the temporal axis to form a $T \times C_{in}$ dimensional video representation where $1 \times C_{in}$ is the feature shape per segment. This video representation denoted as F_{in} is further input to the RGB or pose stream in our architecture.

6.3.2 Model Structure

In this section, we present the structure of the AGNet.

Our stacked-dilated temporal convolution network (SD-TCN) is a TCN-based network. This network has 5 blocks, each block has one 1-dimensional convolution layer, one Hadamard product with the attention weights from the attention module and a residual link. For different blocks, we give different dilation rates to the convolution layer. With these different settings in dilation, we can model local context in the lower block and global context in higher blocks. In our experiment, we set the kernel size (k) to 3 for all convolution layers, dilation (d_i) and padding rate to 2^{i-1}, thus the reception field is up to $2^i + 1$ for the i^{th} block.

In parallel to the SD-TCN, the attention module is another TCN-based model. The attention module has a similar 5-block structure as the SD-TCN, and also the same kernel and dilation setting for the convolution inside the block. Thus, the attention module has the same receptive field as the SD-TCN for each block. However, this module uses significantly lower channel capacity to generate the attention weights. For each convolution layer, it has a ratio of β ($\beta \leq 1$) channels for the SD-TCN. The typical value is $\beta = 1/8$ in our experiments, which is much lower than the SD-TCN. In the attention module, after the convolution layer, we generate the attention map A_i. A bottleneck layer is applied as a transformation to match the channel size to the SD-TCN. Normalizing the high number of T attention weights with softmax leads to extremely low values, which can hamper their effect. To avoid this, we use sigmoid activation to generate the final attention map.

As shown in Fig. 6.2, the input RGB and pose encoding are firstly fed to the bottleneck layers. The output channel size from the bottleneck layers is C_2 and βC_2, corresponding to the SD-TCN and attention module respectively. Then 5 blocks are stacked, the set of

operations in each block can be formulated as follow:

$$F_{i+1}^{A} = F_{i}^{A} + ReLU(Conv1D(F_{i}^{A}, k, d_i)) \tag{6.1}$$

$$A_i = Sigmoid(W_i ReLU(Conv1D(F_{i}^{A}, k, d_i))) \tag{6.2}$$

$$F_{i+1}^{B} = F_{i}^{B} + ReLU(Conv1D(F_{i}^{B}, k, d_i)) \circ A_i \tag{6.3}$$

where F_i^B and F_i^A indicates the input feature map of the i^{th} block of the SD-TCN and attention module respectively. A_i is the attention mask generated from the i^{th} block. $\circ$ indicates the Hadamard product. $W_i \in \mathbb{R}^{C_2 \times \beta C_2}$ are the weights of the bottleneck convolution in attention module.

Finally, we compute the per-frame binary classification score for each class (i.e. prediction logits). The classifier is on top of the SD-TCN, which is another bottleneck convolution with $sigmoid$ activation:

$$P = Sigmoid(W^{'} F_6^B) \tag{6.4}$$

where $P \in \mathbb{R}^{T \times C_3}$ are the prediction logits and $W^{'} \in \mathbb{R}^{C_3 \times C_2}$ are the weights of the bottleneck convolution, C_3 corresponds to the number of action classes. To learn the parameters, we optimize the multi-label binary cross-entropy loss [108].

6.3.3 Comparison with PDAN

Both PDAN (see Sec. 4.4) and AGNet try to decompose the attention map into different temporal levels. However, there are two main differences: firstly, instead of self-attention using only RGB, AGNet explores cross-modality attention. Secondly, AGNet utilises the snippet-level attention, not the kernel-level. we find that the cross-modality model can not provide fine-grained information as kernel-level attention. Thus different from PDAN, we choose to learn the attention outside the kernel, which provides the global temporal region information.

6.3.4 Experiments

As mentioned earlier, AGNet is the proposed baseline of TSU dataset. The goal of these experiments is to verify that the TSU dataset provides novel challenges that are not yet addressed by the other state-of-the-art datasets. For that, we show that the state-of-the-art detection methods perform poorly on TSU and that our AGNet significantly improves the results on TSU as it is designed to address the targeted real-world challenges. To evaluate the effectiveness of the AGNet, we compare it on TSU dataset with 9 detection methods,

			CS	CV
Pose	3D+T	AGCN+Bottleneck [67]	10.1	12.6
		AGCN+LSTM [140]	17.0	14.8
		AGCN+**SD-TCN**	**26.2**	**22.4**
RGB	2D	Inception+Bottleneck [73]	11.5	5.2
		Inception+LSTM [140]	13.2	5.3
		Inception+**SD-TCN**	**22.3**	**12.1**
	2D+T	R-I3D [81]	8.7	-
		I3D+Bottleneck [22]	15.7	9.2
		I3D+Non-local block [28]	16.8	9.6
		I3D+Super event [17]	17.2	10.9
		I3D+LSTM [141]	22.6	12.9
		I3D+Bidirectional-LSTM [139]	24.5	15.1
		I3D+Dilated-TCN [5]	25.1	13.9
		I3D+MS-TCN [19]	25.9	13.1
		I3D+TGM [18]	26.7	13.4
		I3D+**SD-TCN**	**29.2**	**18.3**
RGB+Pose		**AGNet**	**33.2**	**23.2**

Table 6.1: Per-frame mAP (%) on the Fine-grained TSU dataset.

which represent the state-of-the-art on other densely-annotated datasets [2, 13]. We also perform a comparative study between TSU and the challenging Charades dataset for the action detection task to better highlight how real-world challenges are addressed by both datasets.

6.3.4.1 Implementation Details

Video encoding: We use three types of encoders to extract the encoding of the input videos. As described in section 6.3.1, AGCN [67] and I3D [22] are fine-tuned on TSU and then the features are extracted. Moreover, we also evaluate this dataset on per-frame features. We use Inception V1 [73] pre-trained on ImageNet [138] to extract the features. The channel size of I3D and Inception is 1024, the channel size of AGCN is 256.
AGNet: We set $N = 6$ blocks. For I3D and Inception features, the channel size is 1024, for AGCN pose features, the channel size is 256. C_1 is 512 and β is 8. We use Adam optimizer [162] with an initial learning rate of 0.001, and we scale it by a factor of 0.3 with a patience of 10 epochs. The network is trained on a 4-GPU machine for 300 epochs with a mini batch of 32 videos for Charades and 2 videos for TSU. The other baselines' implementation is mentioned in chapter 3.4.1. As mentioned in 3.4.1.2, the Bottleneck used for comparison is a Bottleneck on top of the segment-level features. The improvement over the Bottleneck reflects the effectiveness of modeling temporal information.

	CS			CV		
IoU Threshold (θ)	0.3	0.5	0.7	0.3	0.5	0.7
Bottleneck [22]	5.0	2.5	0.5	2.3	1.1	0.2
Non-local block [28]	4.9	2.2	0.6	1.6	0.7	0.1
Super event [17]	5.7	2.8	0.7	1.8	0.9	0.1
LSTM [140]	11.6	6.4	2.2	6.0	3.2	0.7
Bidirectional-LSTM [139]	13.3	7.9	3.5	9.0	5.4	1.2
Dilated-TCN [5]	12.8	6.9	3.0	5.8	3.3	0.8
MS-TCN [19]	13.2	7.6	3.0	5.3	3.1	0.4
TGM [18]	15.1	9.4	4.2	5.5	3.2	0.4
AGNet	**22.7**	**15.3**	**6.0**	**12.5**	**7.8**	**2.9**

Table 6.2: Event-based mAP (%) for different IoU thresholds for the TSU dataset. The AGNet utilizes both pose and RGB modalities and the other methods utilize only RGB.

6.3.4.2 Experimental Analysis on TSU

In this section, we conduct the ablation and data modality analysis on the fine-grained version of the TSU. In Table 6.1, we firstly compare the three different video encodings: AGCN pose features, inception RGB features and I3D RGB features. We conduct the experiments on the Bottleneck, LSTM and the AGNet. The AGNet is the SD-TCN (RGB) guided by a attention module (pose). On one hand, we observe that using I3D RGB features improves the detection results by up to 11.1% w.r.t. the same method using Inception features. This improvement is intuitive because of the higher ability of the 3D convolutional operations to capture spatio-temporal relations using several datasets for pre-training. On the other hand, we find that, while using the same method, 2D+T RGB features perform better than pose features in Cross-Subject protocol. However, pose features perform better than RGB features in Cross-View protocol (+4.1% for SD-TCN). This reflects that 3D skeleton is more stable while changing viewpoints, which is very helpful in multi-view settings as in TSU. Finally, for the AGNet: SD-TCN (RGB) guided by pose attention module, outperforms RGB and pose SD-TCNs for both CS and CV protocol (+4.0% and +4.9% w.r.t RGB SD-TCN for CS and CV protocol respectively). We also compared AGNet with late fusion SD-TCN (RGB + Pose), our AGNet +0.6% w.r.t. late fusion mechanism on TSU CS protocol. Note that our method lightweights Pose stream while late fusion SD-TCN has regular streams for both RGB and Pose modals.

In table 6.2, we present the event-based evaluation of the detection methods. The AGNet provides more precise predictions than the state-of-the-art methods. However, all these performances are relatively low, indicating that current methods are far from addressing real-world conditions.

Inspired by Charades, to understand the relation between the number of action samples and performance, Fig. 6.3 illustrates AP for each action. In this figure, the action classes are sorted by the number of available samples, together with the name of best per-

Figure 6.3: Average Precision for the actions in TSU. The classes are sorted by their size. The mAP is marked by a red line. We can see that while there is a slight trend for smaller classes to have lower accuracy, many classes do not follow that trend.

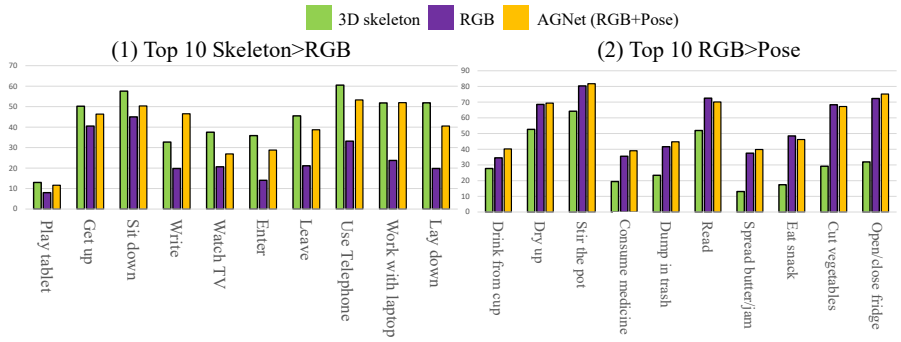


Figure 6.4: Frame-based mAP of the AGNet using different modalities: (1) Top 10 actions where the 3D skeleton stream outperforms the RGB stream for the CV protocol. (2) Top 10 actions where the RGB stream outperforms the 3D Skeleton stream for the CS protocol.

forming classes. The number of samples in a class is primarily decided by the universality of the action (can it happen in any scene), and if it is typical of household environments. It is interesting to notice that, while there is a trend for actions with a higher number of examples to have higher AP, it is not true in general. Activities such as *breakfast*, and *get water* have top-10 performance despite being represented by only few examples.

To understand the advantages of 3D skeleton and RGB modality, in Fig. 6.4, firstly, we select the top 10 actions where 3D skeleton stream outperforms RGB stream in CV protocol. We find that 5 out of the 8 pose-based actions that we defined in Fig. 3.4 (4) are in these top 10 actions. This confirms that 3D skeleton stream has filtered the unnecessary context information in the image, resulting in a better model for the posed-based actions. Secondly, we select the top 10 actions where RGB stream outperforms 3D skeleton stream in CS protocol. We find 7 out of 10 actions are the similar actions with different objects that we defined in Fig. 3.4 (5). This confirms that RGB stream provides the object information lacking in 3D skeleton, which is critical to detect the actions highly correlated with objects. Finally, we show that, while using our attention-based baseline, we can handle both challenges of pose-based actions and similar actions involving different objects.

In Fig. 6.5, we present the attention map of the attention module for 5 layers (on

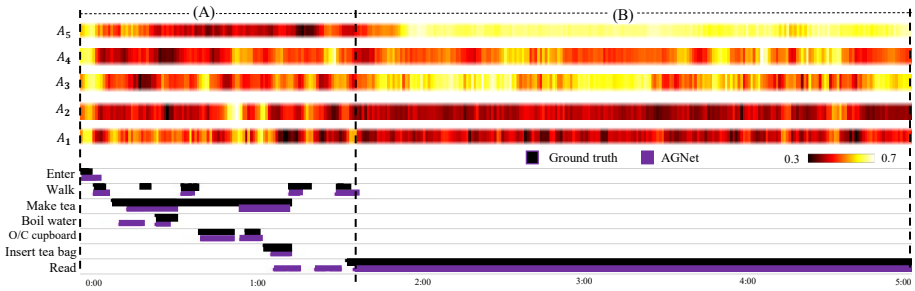


Figure 6.5: Qualitative analysis of the detection result and the attention map. On the top, we visualize the attention map A_i for 5 layers. On the bottom, we present the corresponding ground truth and detection performance for an example video.

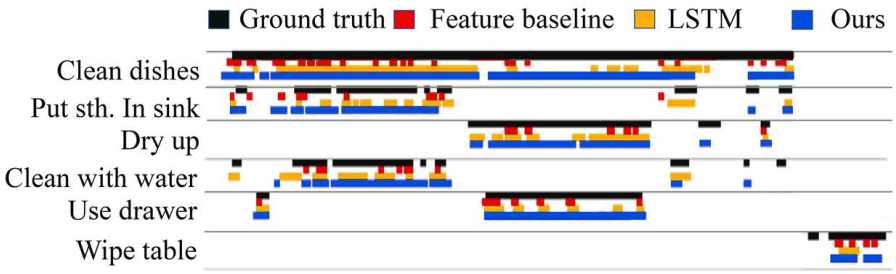


Figure 6.6: Qualitative study

top), and the corresponding ground truth vs. action detection results (on the bottom). On the one hand, in area (A), while detecting short actions, the attention module allocates high attention weights at the lower layer, corroborating that the lower layer is particularly sensitive to short actions. On the other hand, in area (B), with long actions (e.g.*Read book*), only the higher layers allocate high attention weights to the frames in the kernel. This reflects that the higher layers are more sensitive to long-term actions.

In Fig. 6.6, we show qualitative visualization results of three model predictions. In this video, there are one composite long action and 5 elementary actions. We notice that our AGNet can better tackle the long-term temporal relations, detecting the composite action and the related elementary actions simultaneously. Additionally, the AGNet provides better detection for both elementary (e.g. *wipe table*) and composite actions (e.g. *Clean dishes*) compared to I3D and LSTM. However, the detection precision is not sufficient, more work is needed to design better models to detect both composite and elementary actions in untrimmed videos.

In Fig. 6.7, we compare the performance across 4 different action properties of the AGNet and Bottleneck using both RGB and pose modalities (i.e. I3D+AGCN). Bottleneck layer is the baseline reflecting the quality of the feature without temporal processing. Thus, the comparison with the Bottleneck can reflect the improvement from our proposed

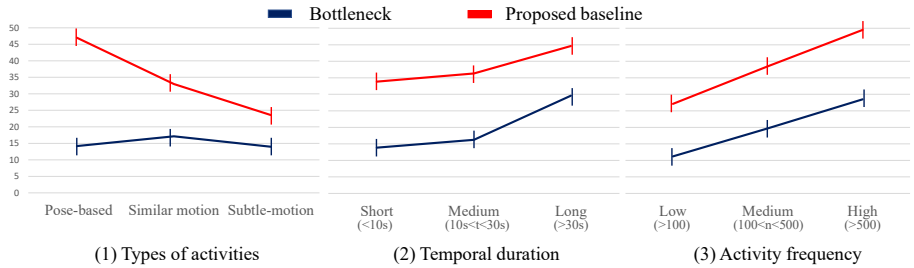


Figure 6.7: We compare the AGNet against the Bottleneck approach across three different action properties using both RGB and Pose modality. Evaluation is provided on frame-based mAP on TSU-CS. The Bottleneck performs poorly on all these types of actions, whereas the AGNet improves the performance on all of them.

methods and the remaining open issues on Fine-grained TSU. In Fig. 6.7 (1), we observe that the AGNet significantly improves the detection of pose-based actions compared to Bottleneck. However, the AGNet does not tackle so well similar motion and subtle motion actions. In Fig. 6.7 (2), we show that longer actions are easier to recognize than shorter ones, similarly to [221]. The consistent performance gain of the AGNet for actions with different temporal duration corroborates its effectiveness to adapt to temporal dynamics. Finally, we show for the AGNet the improvement in the detection of all actions, even of the ones with small numbers of training samples. We are not applying specific measures in the AGNet to handle this issue. Adopting strategies like class-weighting, optimizing through focal loss could be explored in future work.

In summary, we find that the available modalities in TSU are complementary. The AGNet leverages these modalities to address the challenges in TSU such as multi-views, pose-based actions and similar motions.

6.4 Knowledge Distillation for Action Detection

In this section, we first describe the overall architecture of our approach. We then detail the different losses in the proposed framework.

6.4.1 Overall Architecture

An overview of the architecture is shown in Fig. 6.8. In this work, the knowledge transfer occurs between the teacher and student networks. Both networks are composed of a visual encoder and a temporal filter, following the Seq2Seq paradigm. For the visual encoder, we use I3D [22] to encode the spatio-temporal information of a snippet for RGB and Optical Flow (OF). Similar to previous action detection methods [19, 60], sequences of 16 frames are encoded to a single feature vector representation. The encoded feature maps of a video

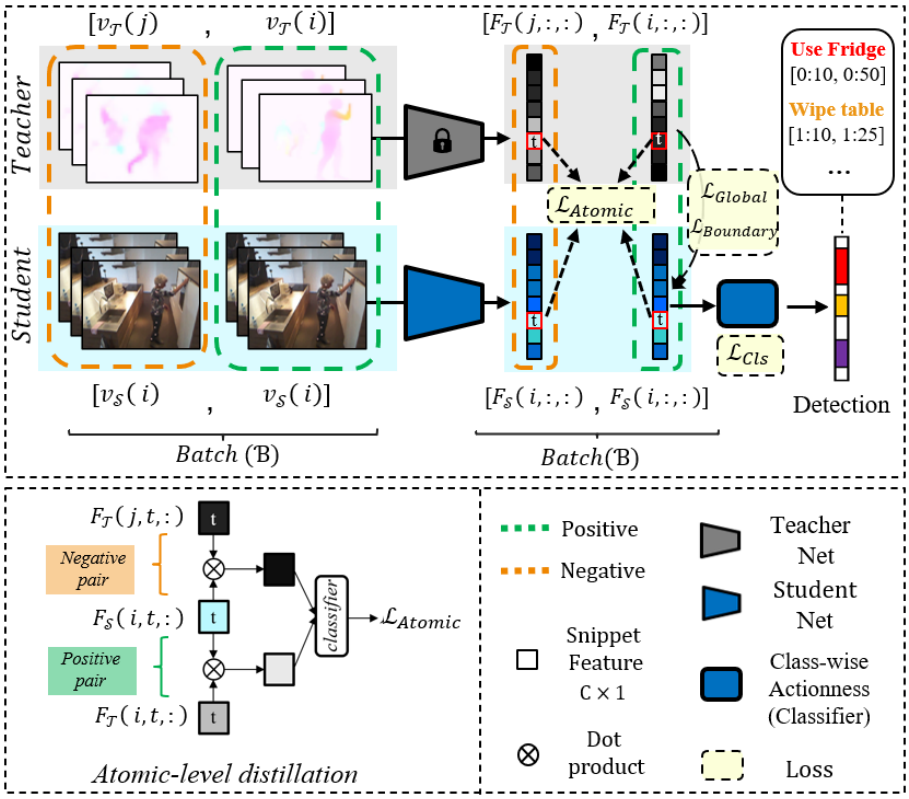


Figure 6.8: The proposed distillation framework. On the top, we present an example of a batch size ($\mathcal{B}$) of 2 untrimmed videos ($\mathcal{V}$) for both student ($\mathcal{S}$) and teacher ($\mathcal{T}$) networks. In this example, the input includes a pair of positive videos and a pair of negative videos. The sequence-level distillation and classification losses are employed only for positive pairs, while atomic-level distillation leverages both positive and negative pairs. On the bottom, we present the atomic-level distillation.

are then fed to the temporal filter. The choice of the temporal filter is flexible, since we can choose any well-known temporal model [18, 5, 26]. Here, we set a 5-layer SS-TCN [19] as default temporal filter, which is based on Dilated-TCN [5]. Both student and teacher have the same type of temporal filter with the same settings (i.e. dilation rate and channel size). In the training phase, the knowledge distillation is performed from the output feature of the teacher network towards the student network. Similar to [18, 102], this output feature map is further classified and grouped as a class-wise actionness detector for detecting the actions.

The input of teacher network is flexible to variant costly modalities (e.g. OF, 3D poses). By default, we chose the teacher network as OF stream, whereas the student network as RGB stream. In the following sections, we express the feature representation of a video indexed i with $F_r(i, t, c)$, where $r \in \{\mathcal{T}, \mathcal{S}\}$ represents the teacher $\mathcal{T}$ and student $\mathcal{S}$; $t \in [1, T]$ represents the snippet index and T the length of the video in snippets; $c \in \mathbb{Z}^C$ represents

the channel index, C is the channel size. This expression can be used for representing feature of a video or a snippet. For example, $F_r(i,:,:)$ and $F_r(i,t,:)$ represent the feature map of a video i and the feature vector of a snippet for video i at time step t, respectively. For an augmented RGB representation, the distillation is performed in two levels. First, we perform distillation at atomic-level to distillate the elementary representation of an action. Second, we perform a sequence-level distillation to distillate (i) the salient relations among the snippets, and (ii) the significant temporal variations across the snippets indicating action boundaries.

6.4.2 Atomic-level Distillation

To transfer the knowledge between two video sequence, firstly, we adapt and integrate the "representation loss" [195] in our overall formulation, dubbed Atomic KD loss. This loss term encourages the student to mimic the feature representation of every individual snippet feature of the teacher network. Our formulation is different from the previous work [195] that minimizes the cosine distance between the snippet features. Inspired by the recent success on contrastive learning [222, 223, 224], we build our model using a contrastive strategy to enhance the atomic-level knowledge imitation.

As shown in Fig. 6.8, let $[F_{\mathcal{S}}(i,t,:),\ F_{\mathcal{T}}(i,t,:)]$ represents a pair of training snippets from same video i at time t but across different modalities for the teacher and student networks. Let $F_{\mathcal{T}}(j,t,:)$ be another snippet representation from a randomly chosen video j of the teacher stream and having a different label. We define the pair $[F_{\mathcal{S}}(i,t,:),\ F_{\mathcal{T}}(j,t,:)]$ as positive when $i=j$, otherwise negative. We aim at pushing closer the representations $F_{\mathcal{S}}(i,t,:)$ and $F_{\mathcal{T}}(i,t,:)$, while pushing apart $F_{\mathcal{S}}(i,t,:)$ and $F_{\mathcal{T}}(j,t,:)$, which can be seen as a binary classification task that tries to maximize the log-likelihood of the mutual information between the student and teacher representations. In practice, the loss is updated by batches with a batch size $\mathcal{B}$. If $\mathcal{N}$ negative pairs exist for each positive pair, then the number of samples in a batch of $\mathcal{P}$ positives is given by $\mathcal{B}=(\mathcal{N}+1)\mathcal{P}$ (see Fig. 6.8). To measure the mutual information between the student and the teacher, we compute:

$$\begin{aligned}\mathcal{L}_{Atomic} = \frac{1}{\mathcal{P}T}\sum_{i=j}\sum_{t=1}^{T}\log[\frac{\exp^{F_{\mathcal{T}}(j,t,:)^\top F_{\mathcal{S}}(i,t,:)}}{\exp^{F_{\mathcal{T}}(j,t,:)^\top F_{\mathcal{S}}(i,t,:)}+\phi}]+\\ \frac{1}{T}\sum_{i\neq j}\sum_{t=1}^{T}[\log(1-\frac{\exp^{F_{\mathcal{T}}(j,t,:)^\top F_{\mathcal{S}}(i,t,:)}}{\exp^{F_{\mathcal{T}}(j,t,:)^\top F_{\mathcal{S}}(i,t,:)}+\phi})]\end{aligned} \tag{6.5}$$

where $\mathcal{P}T$ represents total number of positive snippets, ϕ is the ratio of the negative snippets to the cardinality of snippets in the training set. Note that, this loss term is accompanied by a linear combination with the other distillation losses and the class-wise entropy loss (i.e. supervised learning).

As the length of an action instance is often larger than a snippet, with atomic-level

distillation, the teacher network transfers only the sub-representation of the actions [119]. Next, we propose a novel sequence-level distillation mechanism which has been neglected in the state-of-the-art methods.

6.4.3 Sequence-level Distillation

Sequence-level distillation transfers cross-snippet knowledge between different modalities in an untrimmed video by incorporating contextual information and taking benefit from the variations of cross-modal representation along action boundaries. Consequently, we propose two sequence-level distillation losses: (1) Global Contextual Relation, (2) Boundary Saliency, to improve action detection performance. Note that both sequence-level distillation losses are applied only between positive video pairs, corresponding to $\mathcal{P}$ videos.

6.4.3.1 Global Contextual Relation

For sequence-level distillation, firstly, we propose to transfer contextual knowledge between modalities of the entire video. Intuitively, the detection of one given action could be supported by the detection of other related actions, which may be distant in the untrimmed video. Hence, the representation of an action snippet could benefit from the contextual information across other snippets in the video pertaining to another modality. But the challenge in modeling such contextual relationships is the high complexity of the model for taking into account all the snippets in a video in relation with a single snippet. Therefore, we propose an embedding that projects the student-teacher features in a space where the global contextual relations among all actions are computed.

For the global contextual relation loss, we compute the Channel Covariance Matrix (Cov) of the sequence of snippets which projects the entire video into a compact embedding space. Note that the length of untrimmed videos in the dataset may vary a lot, while the channel size is fixed for all videos. Providing a feature map of the video, Cov encodes the variance within each channel and the covariance between all channels over the whole video. Each element in the matrix reflects the correlation between two channels, which can characterize the specific activation patterns along time of an action class. Thus, the covariance matrix captures the relations between snippets along time and indicates whether a salient relation exists (i.e. which may be related to an action), while being computationally optimal. Here, the Cov is formulated as:

$$Cov_r(i) = \frac{1}{T-1}\sum_{t=1}^{T}[F_r(i,t,:) - \mu_i][F_r(i,t,:) - \mu_i]^T \tag{6.6}$$

such that $r \in \{\mathcal{T}, \mathcal{S}\}$, and μ_i represents the mean value of all the channels in the feature map $F_r(i,:,:)$ of a video i. The covariance matrix $Cov_r \in \mathbb{R}^{C\times C}$ is a symmetric matrix and

thus it is determined by $\frac{C(C+1)}{2}$ values. We apply a filter mask extracting all the entries on and above the diagonal of the covariance matrix. We reshape these values in the form of a vector $G_r(i)$:

$$G_r(i) = mask[Cov_r(i)] \tag{6.7}$$

where $mask(.)$ is the filter mask operation. The obtained feature vector $G_r(i)$ represents the channel covariance of the video. We then enforce a distillation loss in the embedded space from the frozen teacher to the student over the positive video pairs ($\mathcal{P}$). This is performed by minimizing the mean square error, which is formulated as the Global Contextual Relation loss:

$$\mathcal{L}_{Global} = \frac{1}{\mathcal{P}} \sum_{i=1}^{\mathcal{P}} ||G_{\mathcal{T}}(i) - G_{\mathcal{S}}(i)||^2 \tag{6.8}$$

The differential property of equation 6.6 enables to train our teacher-student framework jointly with the other losses.

6.4.3.2 Boundary Saliency

The boundary saliency loss term is used in our formulation to learn comparatively precise boundaries for action detection. In an untrimmed video, we find that the starting and ending of the action are more salient than other parts, that brings us crucial information to detect the transition of an action to another action or background. Intuitively, the sharp variation across consecutive snippets in the video can reflect such saliency of the action boundaries, which is a cross-snippet knowledge. Transferring the knowledge of feature evolution along time encourages the features to be more sensitive at the action start and end, thus assisting the class-wise actionness detector in the student network to detect precise boundaries of the action instances. Such an approach is especially effective when the modality processed at the teacher network provides pertinent boundary information. For instance, modalities which are sensitive to motion (e.g. OF, 3D poses) are able to bring a significant benefit from this loss term. In addition, this loss-term also encourages to retain the temporal consistency across the different modalities.

In practice, we first define the variation between consecutive snippets as $Var(i)$ for video i, which is formulated as:

$$Var_r(i) = \frac{1}{T-1} \sum_{t=1}^{T-1} \sum_{c=1}^{C} [F_r(i, t+1, c) - F_r(i, t, c)] \tag{6.9}$$

where $r \in \{\mathcal{T}, \mathcal{S}\}$. Then, we define the Boundary Saliency loss as the $L1$ distance between the frozen teacher and the student network over the $\mathcal{P}$ positive pairs, which is formulated as:

$$\mathcal{L}_{Boundary} = \frac{1}{\mathcal{P}} \sum_{i=1}^{\mathcal{P}} |Var_{\mathcal{T}}(i) - Var_{\mathcal{S}}(i)| \tag{6.10}$$

With both sequence-level distillation losses, the student network learns two types of cross-snippet information from the other modalities. Below, we summarize the training procedure section.

6.4.4 Training and Testing

To sum up, firstly, we train the teacher networks with the classification (Cls) loss, i.e cross-entropy. The weights of the teacher network is then frozen followed by training the student network. During training, multiple distillation losses are jointly optimized with classification loss for the end task, i.e. action detection. On one hand, the atomic distillation is trained in a contrastive manner (with positive and negative pairs), whereas the sequence-level distillation losses are performed in a non-contrastive manner by utilizing only the positive pairs in a batch. The overall objective is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{Cls} + \alpha_1 \mathcal{L}_{Atomic} + \alpha_2 \mathcal{L}_{Global} + \alpha_3 \mathcal{L}_{Boundary} \tag{6.11}$$

where α_i are the loss weighting factors determined during the validation step. $\mathcal{L}_{Cls}$ represents the cross-entropy classification loss. We call the educated-student network as **Augmented-RGB**. During inference time, we only use RGB videos as input to detect the actions and up-sample the predicted logits to the same temporal resolution as the ground truth to perform the evaluation.

6.4.5 Experiments

To corroborate the effectiveness of our proposed KD framework, we perform an exhaustive experimental analysis for the action detection task.

6.4.5.1 Datasets

We evaluate our framework on five action detection datasets: Charades [114], PKU-MMD [64], TSU [43], THUMOS14 [11], and MultiTHUMOS [13]. These datasets contain videos of different types: (1) sport and daily living videos, (2) short and long videos, (3) densely and sparsely labelled videos. Note: there are two settings on Charades: (a) video-level action classification, (b) frame-level action detection (Charades_v1_localize [114]). We only target the second one in this paper.

All the datasets are evaluated by the mean Average Precision (mAP). We evaluate the per-frame mAP on densely labelled datasets following [13, 114].

6.4.5.2 Implementation Details

For extracting the additional modalities, Optical Flow (OF) is obtained using TVL1 [225], the 3D Poses are extracted using LCRNet++ [134]. In this work, we adapt the 5-layer

	$\mathcal{L}_{Atomic}$	$\mathcal{L}_{Global}$	$\mathcal{L}_{Boundary}$	Charades	PKU-MMD
Teacher-OF	–	–	–	18.6	68.4
Vanilla-RGB	–	–	–	22.3	79.6
Two-stream	–	–	–	24.8	83.4
Atomic	✓	–	–	23.9	82.7
Sequence	–	✓	–	23.8	83.7
	–	–	✓	23.4	83.1
	–	✓	✓	24.2	84.2
Mixture	✓	✓	–	24.4	84.3
	✓	–	✓	24.2	83.7
Total	✓	✓	✓	**24.6**	**85.5**

Table 6.3: Ablation study for the proposed framework on Charades and PKU-MMD (CS) datasets. For PKU-MMD we consider IoU=0.1.

SSTCN [19] as the temporal filter, the output channel size C is 256. While training the teacher-student framework, we use Adam optimizer [162] with an initial learning rate of 0.001, and we scale it by a factor of 0.3 with a patience of 10 epochs. The network is trained for 300 epochs with a mini-batch $\mathcal{B}$ of 16 videos for Charades, 8 videos for PKU-MMD, THUMOS, and 4 videos for the TSU dataset. $\mathcal{N}$ is set to 1, $\mathcal{P}$ as $\frac{\mathcal{B}}{2}$ and α_i=[300, 100, 5]. We use binary cross-entropy for multi-label classification. For sparsely-labelled datasets: THUMOS14 and PKU-MMD, following [195, 102], a post-processing step is performed to generate the action events.

6.4.5.3 Ablation Study

Firstly, we discuss about the effectiveness of the losses proposed in our distillation framework. Tab. 6.3 shows the comparison of action detection performance on Charades and PKU-MMD (IoU=0.1). This table also shows the impact of progressively integrating the KD losses in our distillation framework. The vanilla-RGB is the network trained using only $\mathcal{L}_{Cls}$ without distillation. Compared to vanilla RGB, while training with $\mathcal{L}_{Atomic}$, $\mathcal{L}_{Global}$, $\mathcal{L}_{Boundary}$ independently obtains an improvement of +3.1, 4.1, 3.5% mAP on PKU-MMD respectively. The action detection performance is further improved by the convex combination of any two losses *w.r.t.* their individual counter-parts. This shows the complementary functionalities of the proposed losses. Also, note that the combination of the sequence losses contributes higher than the atomic loss. This observation supports the importance of the sequence-level losses for action detection. Finally, when trained with all the three losses, the student outperforms all the baselines (+2.3%, +5.9% *w.r.t.* vanilla RGB stream on Charades and PKU-MMD). These results show that both our design choices and different losses contribute to the overall performance of our approach.

In Tab. 6.4, we show that our distillation mechanisms perform better at feature-level than at logit-level. The primary reason behind this trend is that we are performing cross-

Table 6.4: Feature-level and logit-level distillation. The student learns from OF stream. For PKU-MMD, we set IoU=0.1.

	Charades	PKU-MMD
Logit	23.7	84.9
Logit+Feature	24.2	85.4
Feature (Ours)	**24.6**	**85.5**

Table 6.5: Comparison with cross-modal KD methods and $\mathcal{L}_{Atomic}$ on Charades and PKU-MMD datasets. For PKU-MMD, IoU=0.1.

	Charades	PKU-MMD
Vanilla-RGB	22.3	79.6
$+\mathcal{L}_{Hall}$ [193]	22.7	81.5
$+\mathcal{L}_{MARS}$ [192]	23.5	81.7
$+\mathcal{L}_{GD}$ [195]	23.3	82.2
$+\mathcal{L}_{Atomic}$(Ours)	**23.9**	**82.7**

	Charades	PKU-MMD	TSU-CS	TSU-CV
Teacher-OF	18.6	68.4	29.4	17.5
Teacher-Pose	9.8	65.0	26.2	22.4
Vanilla-RGB	22.3	79.6	29.2	18.9
Two-stream RGB + Pose	23.0	82.9	32.6	23.7
Two-stream RGB + OF	24.8	83.4	33.5	19.5
Pose Augmented RGB	23.2	84.7	32.4	23.6
OF Augmented RGB	24.6	85.5	32.8	19.3
Pose + OF Augmented RGB	24.9	86.3	33.7	23.8

Table 6.6: Ablation for different modalities on Charades, PKU-MMD (CS), TSU-CS and TSU-CV. For TSU, the reported values are frame-based mAP (%). The IoU threshold for PKU-MMD is 0.1.

modal distillation, where the frozen teacher may under-perform compared to the student network (e.g. OF on Charades and PKU-MMD) *w.r.t.* the different modalities. As the logits represent the classification scores, they may introduce noise from the weak teacher via KD into the RGB student.

6.4.5.4 Analysis of our Distillation Framework

In this section, we further analyze our distillation framework in different aspects.

Comparison with popular cross-modal KD methods: Tab. 6.5 presents a comparison of our extended atomic distillation with state-of-the-art cross-modal KD methods, learning from OF. These baseline methods [226, 192, 193, 227] using traditional losses like MSE and cosine distance are actually designed for classification tasks. For the comparative analysis with our $\mathcal{L}_{Atomic}$, we adapt them following [195] for the task of action detection. $\mathcal{L}_{Atomic}$ consistently outperforms all the baseline methods on Charades and PKU-MMD datasets (+1.6%, +3.1% *w.r.t.* vanilla RGB stream on Charades and PKU-MMD).

Analyzing our framework with different modalities: In Tab. 6.6, we validate that our proposed method is generic and can be effective with different modalities. For experimentation, we perform distillation from OF and 3D Poses. For 3D poses, the teacher consists of 2s-AGCN [228] as visual encoder followed by the temporal filters for detecting actions. In datasets like Charades, most actions involve human-object interactions with prominent motion patterns and in datasets like PKU-MMD, most actions have similar appearance with

Stream	SH (CS)	NTU-60 (CS)	NTU-60 (CV)	NTU-120 (CS_1)	NTU-120 (CS_2)	N-UCLA ($V^3_{1,2}$)
#training samples	8.8k	34.7k	37.6k	52.9k	52.2k	1k
RGB	53.4	85.5	87.3	77.0	80.1	86.0
3D Poses	51.5	85.8	93.8	79.6	81.1	78.2
RGB+3D Pose (Late Fusion)	63.0	87.7	94.8	81.1	83.3	87.1
Ours	67.1	90.8	93.8	85.1	87.6	89.1

Table 6.7: Top-1 accuracy of RGB, 3D Poses, and the Augmented-RGB on 4 datasets.

variant motion over time. Thus, OF stream provides more salient information than Pose stream on these datasets. Whereas 3D Poses are robust to the change of the view-points and thus, significantly improves the action detection performance in cross-view settings (see Tab. 6.6). Furthermore, with a multi-teacher network with OF and Poses, the RGB stream now dubbed as **Pose + OF Augmented RGB** learns some additional information (+2.6%, +6.7%, +4.5%, +4.9% *w.r.t.* vanilla RGB stream on Charades, PKU-MMD, TSU-CS, TSU-CV respectively).

Can $\mathcal{L}_{Atomic}$ generalized to action recognition? As the proposed atomic-level loss is close to action recognition task. We also conduct the experiment for studying if this loss-term can benefit the action recognition task. In practice, we distillate the knowledge from 3D poses to RGB stream, where RGB backbone is an I3D model [22] and Pose backbone is a Graph Convolutional Network [178]. The distillation occurs at the feature-level between the outputs of the two backbones. In Table 6.7, we compare our distillation model with uni-modal models and their combinations on SH [50], NTU-60 [62], NTU-120 [229] and N-UCLA [63] datasets. Following the state-of-the-art trends, RGB and Poses are combined using score level fusion (i.e., late fusion). Our method significantly outperform the individual modalities. With our contrastive distillation ($\mathcal{L}_{Atomic}$), the Augmented-RGB outperforms the late fusion strategy of combining RGB and Poses on all the datasets except NTU-60 (CV protocol). This experiment shows the generalization and robustness of our atomic-level distillation method. We have also extexded our Video Pose Network [55] by this atomic-level distillation loss and further outperform SoTA on multi-modal action recognition task [54].

Inference time & Complexity: Fig. 6.9 shows the precision vs inference time per video on Charades dataset. The inference time includes the time of extracting the additional modalities and the processing time of the visual encoder and temporal filter. We find that Two stream RGB + TVL1 [225] achieves high precision on Charades, but at the expense of a high computational cost. In this work, we use TVL1 to obtain the OF modality. Although there are methods [230, 231] that generate OF at higher speed, these methods perform significantly worse than TVL1 [192]. Similarly, the computation of accurate 3D Poses is not real-time, and hence doubles the video processing time [134]. With these

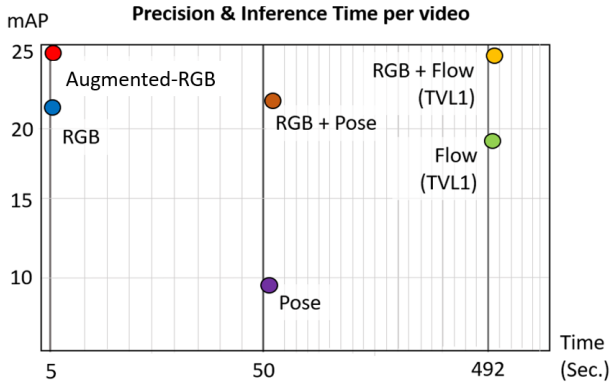


Figure 6.9: Precision vs Inference time per video on Charades.

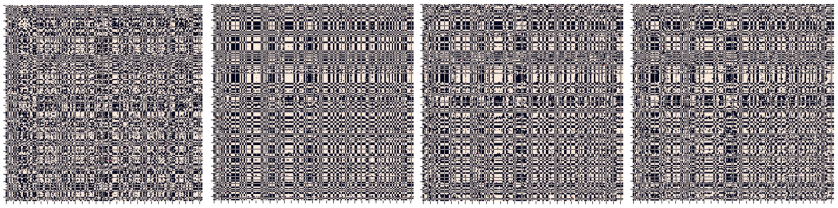


Figure 6.10: Channel Covariance. We visualize the Covariance matrix of a video for the vanilla RGB, vanilla OF, the two-stream RGB+OF, and the Augmented-RGB ($\mathcal{L}_{Global}$). For better visualization, we normalize the matrix to [0,1] and set a threshold of 0.5.

modalities in training phase, our proposed framework avoids estimating these modalities at test time while keeping the performance of two-stream network. The processing speed at the inference phase (I3D+SSTCN) is about 140 fps using 4 GPUs, thus can be seen as a real-time processing.

Concerning complexity, as we have the same type of temporal filter and encoder for teacher and student, the Augmented-RGB stream retains the same number of parameters as the vanilla RGB stream at inference time, whereas two stream network doubles the number of parameters often causing over-fitting [232].

6.4.5.5 Qualitative Analysis

With Global Contextual Relation loss, the student learns the relationships among the action instances of the teacher network along with retaining the student's individual representation. As shown in Fig. 6.10, with only $\mathcal{L}_{Global}$, the channel covariance representation of Augmented-RGB is closer to the one of RGB+OF. Hence, the Augmented-RGB achieves performance close to the one of the two-stream network.
We also compare the performance of RGB stream with Boundary Saliency distillation and vanilla RGB stream. In Fig. 6.11, we find that the network with $\mathcal{L}_{Boundary}$ detects tighter

Figure 6.11: Action boundary detection: (1) Ground truth indicates if it is action or background at this frame. (2) The boundaries detected without $\mathcal{L}_{Boundary}$, (3) The boundaries detected with $\mathcal{L}_{Boundary}$.

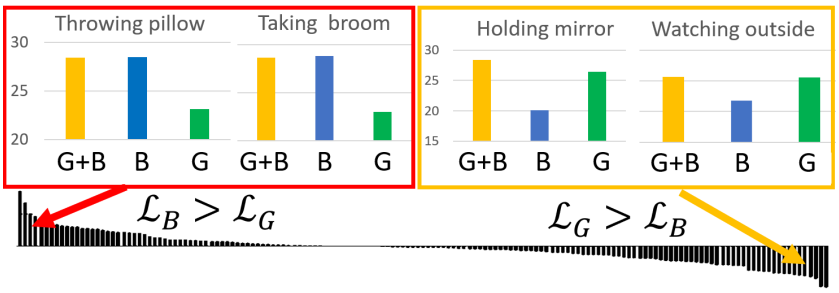


Figure 6.12: Difference of Average Precision for two sequence-level distillation losses on Charades dataset. G: $\mathcal{L}_{Global}$, B: $\mathcal{L}_{Boundary}$.

temporal boundaries of the actions compared to the vanilla network. To further show how two sequence-level distillation losses are complementary, we compare APs for a student that is trained with only $\mathcal{L}_{Global}$ or $\mathcal{L}_{Boundary}$ on Charades in Fig. 6.12. We find that $\mathcal{L}_{Boundary}$ improves more the actions with high variation across time (e.g. *Throw pillow*), $\mathcal{L}_{Global}$ improves more the actions with relatively longer duration (e.g. *Holding mirror*). While learning from $\mathcal{L}_{Global}+\mathcal{L}_{Boundary}$, the student improves all action types, reflecting how these two loss-terms complement each other.

Fig. 6.13 shows the class-wise actionness result of the vanilla-RGB and Augmented-RGB in a densely labelled video along with the action detection results. We notice that the Augmented-RGB detects tight action boundary *w.r.t.* the vanilla-RGB, e.g. *use cupboard, walk*. Thanks to our distillation methods, the Augmented-RGB now predicts the *use drawer* action which is miss detected in vanilla-RGB.

6.4.5.6 Comparison with the State-of-the-Art

In Tab. 6.8, we compare other action detection methods with our Augmented-RGB on PKU-MMD. Recall that our distillation mechanisms are build on SSTCN. While one method [236] using Poses achieves very high performance, this method is skeleton-based and applicable only for specific datasets (i.e. NTU-RGBD [62], PKU-MMD [64]), where high quality 3D Poses are available. In contrast, our method is generic and does not rely on Poses at inference time while being more effective compared to other RGB based SoA methods, such as Graph Distillation [195] (+2.6%, +2.4%, +4.6% for 0.1, 0.3, 0.5 IoU),

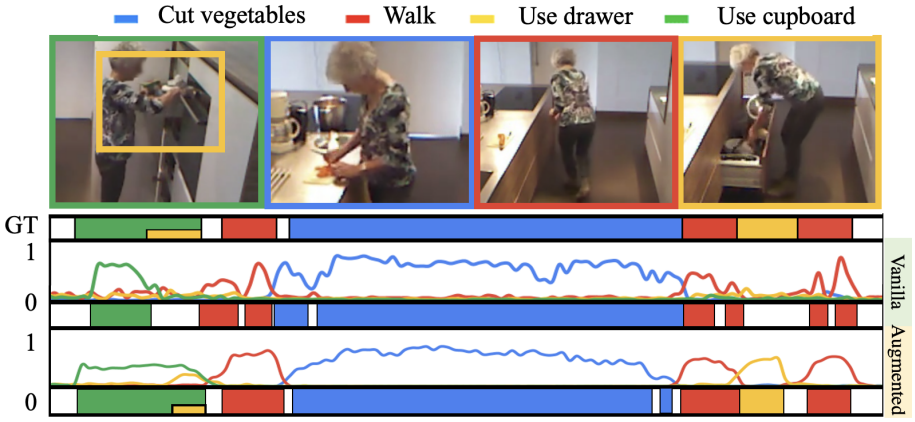


Figure 6.13: Class-wise actionness with the detection results.

		Method	mAP@tIoU (θ) 0.1	0.3	0.5
Test modality	Poses	JCRRNN [233]	45.2	–	32.5
		Convolution Skeleton [64]	49.3	31.8	12.1
		Skeleton boxes [234]	61.3	–	54.8
		Wang and Wang [235]	84.2	–	–
		Li et al. [236]	92.2	–	90.4
	RGB	Deep RGB [64]	50.7	32.3	14.7
		Qin and Shelton [237]	65.0	51.0	29.4
		GRU+GD [195]	82.4	81.3	74.3
		SSTCN+GD	83.7	82.1	76.5
		Augmented-RGB	**86.3**	**84.5**	**81.1**

Table 6.8: Event-based mAP on PKU-MMD (CS) dataset. Only the last five rows utilize RGB at inference time. Note that Graph distillation (GD) learns from more than 4 modalities while our method learns from OF and Pose.

which utilizes the same temporal filter but more modalities (e.g. depth) at training time compared to our method.

To show the generalization of our method, we also evaluate our distillation framework on Charades and TSU-CS, MultiTHUMOS and THUMOS in Table 6.9. For all these comparisons, the student network is distilled with teacher pre-trained with OF in the training phase, as Poses are not always available. For a fair comparison with our Augmented-RGB, Vanilla-RGB and Two-stream networks are implemented using SSTCN. In this table, we find that, anchor-based methods (e.g. AFNet) perform decently on sparsely-labelled datasets, while failing on densely labelled datasets due to the combinatorial explosion of proposals. On the other hand, Seq2Seq architectures are stable on both types of dataset. With the help of our proposed distillation method, the Augmented-RGB achieves the competitive Two-stream performance on all the datasets (+2.3, 3.6, 6.8, 7.2 % *w.r.t.* vanilla-RGB on Charades, TSU, MultiTHUMOS, THUMOS14 respectively). We observe that the performance improvement on THUMOS which consists of sport videos, is significant due

Type	Model	Dense			Sparse
		Charades	TSU-CS	MultiTHUMOS	THUMOS14
Anchor	R-C3D [81]	12.7	8.7	—	28.9
	TAL [20]	—	—	—	42.8
	G-TAD [21]	—	—	—	40.2
	AFNet [92]	13.1	—	—	49.5
Seq2Seq	TAN [102]	17.6	—	33.3	46.8
	WSGM [163]	18.7	—	—	32.8
	TGM [18]	21.5	26.7	44.3	53.5
	Vanila-RGB [19]	22.3	29.2	37.8	46.1
	Two-stream	24.8	33.5	44.4	53.7
	Augmented-RGB	24.6	32.8	44.6	53.3

Table 6.9: Comparison with State-of-the-Art action detection methods. Our method learns only from OF. The cells in white are the two stream results (RGB+OF), while the cell in orange represents using only RGB at Inference time. We report frame-based mAP and event-based mAP for the dense and sparse labelled datasets respectively. The IoU is 0.5 for THUMOS14.

to strong motion patterns resulting to an effective OF based teacher network. Thus, Augmented-RGB while using only RGB at inference, performs on par with Two-stream network for the task of action detection.

6.5 Conclusion

In this chapter, we have introduced two frameworks for learning cross-modal representation for the action detection tasks. Both frameworks are built upon a temporal convolution network. Firstly, we propose AGNet, which is the baseline of TSU dataset. This model is designed to address many real-world challenges existing in TSU. For instance, for dealing with large temporal variance, the attention module generates attention masks at different temporal scales to help detect actions with different temporal lengths. For multi-view challenge, we use both RGB and 3D skeleton to better tackle the view variance problem. This is because 3D skeleton is robust to different view points. We show that our baseline outperforms the state-of-the-art on all the evaluation protocols of TSU. As a continuation of AGNet, we build a distillation framework for action detection, which leverages only RGB at inference time. This distillation framework encourages the RGB stream to learn three types of knowledge to better benefit from the cross-modal information in untrimmed videos. Thanks to this framework, we can improve the performance of vanilla RGB networks and make it possible to detect actions in real-time with high precision, even in case of densely labelled datasets. Experiments show that the proposed method can efficiently infuse different modalities into RGB. For instance, the Augmented-RGB network achieves a performance similar to the Two-stream network while using only RGB at inference time.

Chapter 7

Discussion and Future Work

In the final chapter, we summarize the contributions of this thesis and depict the future work directions.

7.1 Contribution Summarization

In this thesis, our work revolves around temporal action detection tasks in real-world videos (see Fig. 7.1). We have a special focus on analysing the videos with dense action occurrences and videos with fine-grained actions. Firstly, we introduce a real-world indoor dataset: **Toyota Smarthome Untrimmed** (chapter 3). As the name of the dataset suggests, we aim at providing a large indoor dataset to detect human behaviours in an ordinary smart-home. This is an important application for action detection with the societal objective of helping older people to live longer in their preferred environment. To this end, we recorded the daily life activities of 18 older people and we extensively annotated all the human actions that appear in those videos. As the recording process is unscripted and without constraints, this dataset features many properties that lie in the "real world" but that are overlooked by the existing datasets. For instance, these properties include composite actions, concurrent actions, high camera framing, and so on. We compared this dataset with the current state-of-the-art indoor action detection datasets and we showed the contribution of our dataset in terms of new challenges. We believe that releasing videos with those properties could help design better action detection methods for smart-home applications.

Besides the dataset, we introduced multiple approaches for action detection. These approaches can be divided into three topics, which aim at solving real-world challenges in action detection in different manners.

(1) **Temporal relational reasoning** (chapter 4): As the action detection model takes as input untrimmed videos, one of the primary focuses of this dissertation is on the temporal

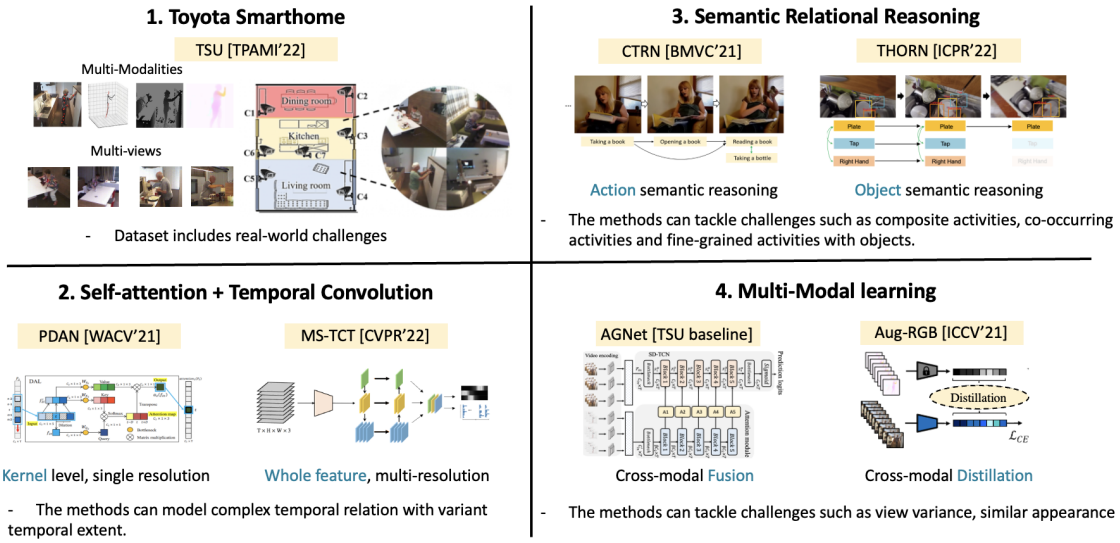


Figure 7.1: Summary of Thesis.

modelling of the video. We proposed three different temporal networks. SA-TCN features a temporal encoder and decoder structure. The attention module on top of the shrunk temporal feature in the middle stage enables the model to handle long-term temporal dependencies. However, the shrunk temporal feature may fail to capture the short-term temporal dependencies. To model both short-term and long-term temporal dependencies and complex temporal relations in videos, we propose PDAN and MS-TCT. PDAN is a temporal convolutional network, with temporal kernels which are adaptive to the input data. MS-TCT is a ConvTransformer network that leverages both temporal convolutional layers and multi-head attention layers at multiple temporal scales. Both networks can capture the different levels of temporal dependencies in a video.

(2) **Semantic relational reasoning** (chapter 5): Knowledge of action relations or object relations can be critical to detect actions in a video, especially for complex actions with long-term dependencies. In this thesis, we proposed a general semantic reasoning framework that can extract the semantic representations (e.g. object, action) from a video and which uses a graph convolutional network to learn the relations across different semantics. We evaluated this framework to model the object-object relations in the visual encoder and action-action relations in the temporal module. As a briefly recap: the aforementioned visual encoder and temporal module are two main components of the sequence-to-sequence action detection framework: (a) the visual encoder encodes a series of video frames into features, and (b) temporal module models the temporal dependencies among the temporal features. The experiments show that our semantic reasoning framework can effectively extract the semantic representations and enhance the video representation for recognizing and detecting fine-grained actions.

(3) **Multi-Modality Representation Learning** (chapter 6): The last contribution was to propose to enhance RGB representation by learning from other modalities in action detection. We firstly proposed AGNet which utilises an additional modality to generate attention masks at multiple temporal scales. Each mask indicates the region of interest of actions on a certain temporal scale, therefore it can help the RGB model to better detect the action. However, AGNet still relies on the additional modality in the testing phase. As a continuation, we proposed a knowledge distillation framework that can transfer the knowledge from the additional modality and use only RGB at inference time. The experiments show that our method can effectively transfer knowledge from the representation of the additional modality to the RGB model.

7.2 Limitations and Perspectives

We then analyse the limitations of the current methods and we outline the future work directions.

7.2.1 Visual Encoding

The recent action detection methods' [81, 21, 15] performance is not satisfying on the popular benchmarks, especially for the datasets with dense action occurrences [92, 18], such as Charades and TSU. Although the proposed temporal models are effective on other temporal reasoning tasks [107, 140, 103], due to the limitation of an unoptimized spatio-temporal visual encoder, the final action detection results are still low.

Firstly, the issue lies in the window approach of temporal feature extraction. The current visual encoders, such as 3D Convolutional Networks [22, 57, 23] and video Transformers [76, 157, 24], are all designed for pre-segmented videos, where each video represents a complete action instance. The video snippet with the same label should be represented similarly. However, in practice, the visual encoder extracts video features from video snippets (i.e., no-overlapping small windows), not from the complete action instances. Each snippet contains only a tiny part of action information and can be taken from anywhere in the action instance. These incomplete snippets increase extremely the data diversity at inference time resulting in an over-fitting issue for the current models. There are some attempts [238, 239, 240] for introducing additional completeness or boundary detection sub-tasks in the pre-training phase. However, those methods are tailored for sparsely annotated videos and cannot handle videos with dense action regions. As a future direction, utilizing masked auto-encoder [241], which encourages the visual encoder to learn robust action instance representation from randomly masked instances, may mitigate the over-fitting problem.

Another way to enhance the visual encoder is to utilise additional object trackers. Currently, the input to the temporal module is a one-dimensional representation where each time step corresponds to a single feature vector. This design makes the temporal module difficult to capture the spatial or semantic information from the video. To tackle this, object trackers can extract the object semantics from the video and enrich the input of the temporal module. For example, we can input the two-dimensional ($Object \times Time$) feature map to the temporal module to explore the object-temporal relations. However, current object detection datasets are not generalized enough for full semantic action understanding. This is because many objects involved in fine-grained actions [2] may not be included in existing large object detection or image classification datasets. Moreover, the imperfect object detection performance, especially in the case of low-resolution videos, make it hard to detect the object precisely from the video.

Furthermore, the disassociation of the visual encoder and temporal module may lead the visual encoder can not effectively extract features for the final objective task (i.e., not end-to-end training). In other words, this disassociation leaves the visual encoding sub-optimal and restricts the action detection performance. Current visual encoder and temporal module are not optimized jointly in our current networks, due to hardware limitations. To link these two modules, a possible approach is to add a momentum memory bank [29, 30] in-between the visual encoder and temporal module. With this dynamic bridge between both modules, the temporal module could gradually access the spatial information of the video. As a result, the visual encoder and temporal module can be trained end-to-end. Note that the previous approach can be seen as utilizing a frozen memory bank of the extracted snippet feature, while this new manner provides the model with a memory bank which is updated dynamically.

7.2.2 Other Challenges

Besides the limitations in visual encoder, we are interesting in tackling other challenges in action detection tasks.

Firstly, all the methods introduced in this thesis are fully-supervised action detection methods which require the complete annotation of all action instances (i.e., temporal boundaries and categories) in training videos. However, such supervised learning strategy is very time-consuming and costly. To eliminate the need for exhaustive annotations in the training phase, limited supervision is required. Contrary to full supervision, in limited supervision, the annotations are unavailable or partially available. In the future, we want to build a weakly supervised framework for action detection using only the video-level labels. With video-level labels, the network like STPN [242] can leverage T-CAM to provide guidance to locate the action instance in the video. T-CAM is a one dimensional

class-specific activation map in the temporal domain. As T-CAM quality highly relies on the temporal dependencies, it is possible to extend our architecture to propose a weakly supervised learning strategy.

Secondly, how to handle long-tailed data for action detection? In this dissertation, our methods do not have a specific design for the action categories which have only a few samples. Therefore, the results are not balanced for the "head" categories and the "tail" ones. To tackle this issue, a possible approach is similar to few-shot learning [133] that learns the action representation more efficiently with only a few samples. Another direction can be to pre-train the action representation from similar large datasets and to transfer the knowledge to the domain of the target dataset. As multiple actions can occur at the same time, the mix-up augmentation [243] for different action instances during the pre-training phase can also help to learn the co-occurring action representation.

Finally, how to detect actions that involve multiple subjects? In this thesis, our detection focuses on actions performed by a single subject. Although we also evaluate our methods on sport datasets such as MultiTHUMOS, the annotation is subject-agnostic. To detect more complex activities that involve multiple subjects (e.g. actions in basketball games), a framework is needed to explore the relations across different subjects. Our future work is to extend the current semantic reasoning framework: not only modelling the object-object (i.e., THORN) and action-action (i.e., CTRN) relations but also modelling the subject-subject relations in the videos. In other words, we want to construct a novel hierarchical model for complex action detection by combining different types of semantic reasoning modules.

Bibliography

[1] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *European Conference on Computer Vision (ECCV)*, 2018. (Cited on pages xi, 10, 27, 28, 34, 37 and 107.)

[2] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *European Conference on Computer Vision(ECCV)*, 2016. (Cited on pages xi, 2, 10, 19, 24, 26, 32, 33, 34, 38, 46, 55, 58, 60, 114, 122 and 142.)

[3] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding," *arXiv preprint arXiv:1703.07475*, 2017. (Cited on pages xi and 11.)

[4] A. Richard, H. Kuehne, and J. Gall, "Weakly supervised action learning with rnn based fine-to-coarse modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 754–763, 2017. (Cited on pages xii, 65 and 66.)

[5] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156–165, 2017. (Cited on pages xii, 4, 7, 17, 21, 47, 48, 54, 58, 61, 62, 64, 65, 66, 67, 76, 78, 91, 95, 97, 98, 118, 119, 122, 123 and 127.)

[6] F. Negin, A. Goel, A. G. Abubakr, F. Bremond, and G. Francesca, "Online detection of long-term daily living activities by weakly supervised recognition of sub-activities," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2018. (Cited on pages xii, 65 and 66.)

[7] P. Tirupattur, K. Duarte, Y. Rawat, and M. Shah, "Modeling multi-label action dependencies for temporal action localization," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. (Cited on pages xviii, 2, 21, 25, 26, 56, 60, 78, 82, 83, 85, 86, 94, 101, 102 and 117.)

[8] A. Diba, M. Fayyaz, V. Sharma, M. Paluri, J. Gall, R. Stiefelhagen, and L. V. Gool, "Large scale holistic video understanding," in *European Conference on Computer Vision*, pp. 593–610, Springer, 2020. (Cited on page 1.)

[9] R. Michael, L. Ivan, M. Greg, and O. Sangmin, "Tutorial on human activity recognition," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014. (Cited on page 1.)

[10] C. Schmid and S.-F. Chang, "ActivityNet: large scale activity recognition challenge." http://activity-net.org/challenges/2017/index.html, 2014. (Cited on page 2.)

[11] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "THUMOS challenge: Action recognition with a large number of classes." http://crcv.ucf.edu/THUMOS14/, 2014. (Cited on pages 2, 4, 19, 26, 27, 32, 34, 36, 60 and 131.)

[12] H. Zhao, A. Torralba, L. Torresani, and Z. Yan, "Hacs: Human action clips and segments dataset for recognition and temporal localization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8668–8678, 2019. (Cited on pages 2, 4, 19, 26, 32, 34 and 36.)

[13] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, "Every moment counts: Dense detailed labeling of actions in complex videos," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 375–389, 2018. (Cited on pages 2, 4, 19, 21, 24, 26, 27, 32, 34, 36, 46, 54, 55, 58, 70, 76, 77, 100, 114, 122 and 131.)

[14] U. Yamaguchi, F. Saito, K. Ikeda, and T. Yamamoto, "Hsr, human support robot as research and development platform," in *The Abstracts of the international conference on advanced mechatronics: toward evolutionary fusion of IT and mechatronics: ICAM 2015.6*, pp. 39–40, The Japan Society of Mechanical Engineers, 2015. (Cited on page 3.)

[15] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *ICCV*, 2019. (Cited on pages 4, 17, 19, 20, 93, 97 and 141.)

[16] K. Kahatapitiya and M. S. Ryoo, "Coarse-fine networks for temporal activity detection in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8385–8394, 2021. (Cited on pages 4, 26, 85 and 86.)

[17] A. Piergiovanni and M. S. Ryoo, "Learning latent super-events to detect multiple activities in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. (Cited on pages 4, 6, 17, 26, 47, 48, 50, 59, 68, 76, 77, 85, 91, 95, 101, 102, 104, 114, 117, 119, 122 and 123.)

[18] A. Piergiovanni and M. S. Ryoo, "Temporal gaussian mixture layer for videos," in *International Conference on Machine Learning (ICML)*, 2019. (Cited on pages 4, 17, 21, 26, 47, 48, 59, 68, 76, 77, 78, 85, 86, 95, 102, 104, 115, 117, 119, 122, 123, 127, 138 and 141.)

[19] Y. A. Farha and J. Gall, "Ms-tcn: Multi-stage temporal convolutional network for action segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3575–3584, 2019. (Cited on pages 4, 17, 21, 26, 47, 48, 54, 58, 67, 76, 114, 122, 123, 126, 127, 132 and 138.)

[20] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1130–1139, 2018. (Cited on pages 6 and 138.)

[21] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, "G-TAD: Sub-graph localization for temporal action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10156–10165, 2020. (Cited on pages 6, 17, 20, 114, 138 and 141.)

[22] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, IEEE, 2017. (Cited on pages 6, 17, 18, 46, 47, 48, 50, 56, 59, 60, 64, 67, 77, 78, 95, 110, 116, 119, 120, 122, 123, 126, 134 and 141.)

[23] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, (Washington, DC, USA), pp. 4489–4497, IEEE Computer Society, 2015. (Cited on pages 6, 17, 18, 47 and 141.)

[24] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," *arXiv preprint arXiv:2103.15691*, 2021. (Cited on pages 7, 18, 60 and 141.)

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021. (Cited on pages 7, 18, 56 and 60.)

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997. (Cited on pages 7, 47, 54 and 127.)

[27] M. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova, "TokenLearner: Adaptive Space-Time Tokenization for Videos," *Advances in Neural Information Processing Systems*, vol. 34, 2021. (Cited on pages 7 and 60.)

[28] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018. (Cited on pages 7, 18, 47, 48, 59, 69, 74, 122 and 123.)

[29] F. Cheng and G. Bertasius, "Tallformer: Temporal action localization with long-memory transformer," *arXiv preprint arXiv:2204.01680*, 2022. (Cited on pages 7 and 142.)

[30] C. Zhang, A. Gputa, and A. Zisserman, "Temporal query networks for fine-grained video understanding," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. (Cited on pages 7, 60 and 142.)

[31] P. Lee, Y. Uh, and H. Byun, "Background suppression network for weakly-supervised temporal action localization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 11320–11327, 2020. (Cited on page 7.)

[32] P. Lee, J. Wang, Y. Lu, and H. Byun, "Weakly-supervised temporal action localization by uncertainty modeling," in *AAAI Conference on Artificial Intelligence*, vol. 2, 2021. (Cited on pages 7 and 117.)

[33] L. Fan, P. Xiong, W. Wei, and Y. Wu, "Flar: A unified prototype framework for few-sample lifelong active recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15394–15403, 2021. (Cited on page 8.)

[34] J. Munro and D. Damen, "Multi-modal Domain Adaptation for Fine-grained Action Recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2020. (Cited on page 8.)

[35] V. G. T. da Costa, G. Zara, P. Rota, T. Oliveira-Santos, N. Sebe, V. Murino, and E. Ricci, "Dual-head contrastive domain adaptation for video action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1181–1190, 2022. (Cited on page 8.)

[36] G. V. Horn and P. Perona, "The devil is in the tails: Fine-grained classification in the wild," *CoRR*, vol. abs/1709.01450, 2017. (Cited on pages 8 and 44.)

[37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017. (Cited on page 8.)

[38] Y. Zhang, P. Tokmakov, M. Hebert, and C. Schmid, "A study on action detection in the wild," *arXiv preprint arXiv:1904.12993*, 2019. (Cited on pages 8 and 44.)

[39] M. Petrovich, M. J. Black, and G. Varol, "Action-conditioned 3d human motion synthesis with transformer vae," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10985–10995, 2021. (Cited on page 8.)

[40] G. Varol, I. Laptev, C. Schmid, and A. Zisserman, "Synthetic humans for action recognition from unseen viewpoints," *International Journal of Computer Vision*, vol. 129, no. 7, pp. 2264–2287, 2021. (Cited on page 8.)

[41] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017. (Cited on pages 8, 31 and 46.)

[42] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, pp. 568–576, 2014. (Cited on pages 11, 17, 113, 114 and 116.)

[43] R. Dai, S. Das, S. Sharma, L. Minciullo, L. Garattoni, F. Bremond, and G. Francesca, "Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022. (Cited on pages 14, 15, 31, 78, 113 and 131.)

[44] R. Dai, L. Minciullo, L. Garattoni, G. Francesca, and F. Bremond, "Self-attention temporal convolutional network for long-term daily living activity detection," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–7, IEEE, 2019. (Cited on pages 14, 15 and 53.)

[45] R. Dai, S. Das, L. Minciullo, L. Garattoni, G. Francesca, and F. Bremond, "Pdan: Pyramid dilated attention network for action detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2970–2979, 2021. (Cited on pages 14, 15, 53 and 85.)

[46] R. Dai, S. Das, K. Kahatapitiya, M. S. Ryoo, and F. Brémond, "Ms-tct: Multi-scale temporal convtransformer for action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20041–20051, June 2022. (Cited on pages 14, 15 and 53.)

[47] R. Dai, S. Das, and F. Bremond, "CTRN: Class Temporal Relational Network For Action Detection," in *BMVC 2021 - The British Machine Vision Conference*, (Virtual, United Kingdom), Nov. 2021. (Cited on pages 14, 15 and 91.)

[48] M. Guermal, R. Dai, and F. Brémond, "THORN: Temporal Human-Object Relation Network for Action Recognition," in *26TH International Conference on Pattern Recognition (ICPR)*, 2022. (Cited on pages 14, 15 and 91.)

[49] R. Dai, S. Das, and F. Bremond, "Learning an augmented rgb representation with cross-modal knowledge distillation for action detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13053–13064, October 2021. (Cited on pages 14, 15 and 113.)

[50] S. Das, R. Dai, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond, and G. Francesca, "Toyota smarthome: Real-world activities of daily living," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. (Cited on pages 15, 32, 34, 39, 40, 46, 117 and 134.)

[51] G. Francesca, L. Minciullo, L. Garattoni, S. Das, R. Dai, and F. Bremond, "Method for recognizing activities using separate spatial and temporal attention weights," 2021. EU Patent WO2021069945A1. (Cited on page 15.)

[52] D. Yang, R. Dai, Y. Wang, R. Mallick, L. Minciullo, G. Francesca, and F. Bremond, "Selective spatio-temporal aggregation based pose refinement system: Towards understanding human activities in real-world videos," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2363–2372, 2021. (Cited on pages 15, 18 and 39.)

[53] L. Minciullo, L. Garattoni, G. Francesca, R. Dai, S. Das, and F. Bremond, "Method and system for detecting an action in a video clip," 2022. EU Patent EP3995992A1. (Cited on page 15.)

[54] S. Das, R. Dai, D. Yang, and F. Bremond, "Vpn++: Rethinking video-pose embeddings for understanding activities of daily living," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. (Cited on pages 15 and 134.)

[55] S. Das, S. Sharma, R. Dai, F. Bremond, and M. Thonnat, "Vpn: Learning video-pose embedding for activities of daily living," in *European Conference on Computer Vision*, pp. 72–90, Springer, 2020. (Cited on pages 15, 114, 117 and 134.)

[56] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. (Cited on pages 17 and 18.)

[57] C. Feichtenhofer, "X3D: Expanding Architectures for Efficient Video Recognition," 2020. (Cited on pages 17, 18, 86, 105 and 141.)

[58] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," *CoRR*, vol. abs/1812.02707, 2018. (Cited on pages 17, 59 and 69.)

[59] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2914–2923, 2017. (Cited on pages 17, 19, 20 and 76.)

[60] S. Zhang, H. Peng, J. Fu, and J. Luo, "Learning 2d temporal adjacent networks formoment localization with natural language," in *AAAI*, 2020. (Cited on pages 17, 20 and 126.)

[61] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012. (Cited on page 18.)

[62] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. (Cited on pages 18, 27, 31, 40, 64, 134 and 136.)

[63] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, "Cross-view action modeling, learning, and recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2649–2656, June 2014. (Cited on pages 18, 31, 40 and 134.)

[64] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "Pku-mmd: A large scale benchmark for skeleton-based human action understanding," in *Proceedings of the Workshop on Visual Analysis in Smart and Connected Communities*, VSCC '17, (New York, NY, USA), pp. 1–8, ACM, 2017. (Cited on pages 18, 26, 27, 32, 33, 34, 38, 46, 114, 131, 136 and 137.)

[65] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1110–1118, 2015. (Cited on page 18.)

[66] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018. (Cited on page 18.)

[67] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12026–12035, 2019. (Cited on pages 18, 46, 47, 119, 120 and 122.)

[68] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 143–152, 2020. (Cited on page 18.)

[69] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot, "Skeleton-based online action prediction using scale selection network," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 6, pp. 1453–1467, 2019. (Cited on page 18.)

[70] H. Duan, Y. Zhao, K. Chen, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2969–2978, 2022. (Cited on page 18.)

[71] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3163–3172, 2021. (Cited on pages 18, 21 and 60.)

[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. (Cited on pages 18, 59 and 64.)

[73] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015. (Cited on pages 18, 46, 47, 119 and 122.)

[74] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *proceedings of the IEEE International Conference on Computer Vision*, pp. 5533–5541, 2017. (Cited on page 18.)

[75] N. Hussein, E. Gavves, and A. W. Smeulders, "Timeception for complex action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 254–263, 2019. (Cited on pages 18 and 75.)

[76] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?," *arXiv preprint arXiv:2102.05095*, 2021. (Cited on pages 18, 60 and 141.)

[77] M. Nawhal and G. Mori, "Activity graph transformer for temporal action localization," *arXiv preprint arXiv:2101.08540*, 2021. (Cited on pages 19, 20 and 92.)

[78] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 961–970, 2015. (Cited on pages 19, 26, 32, 34, 36 and 60.)

[79] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016. (Cited on pages 19 and 110.)

[80] J. Gao, K. Chen, and R. Nevatia, "Ctap: Complementary temporal action proposal generation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 68–83, 2018. (Cited on page 19.)

[81] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 5783–5792, 2017. (Cited on pages 19, 20, 47, 76, 85, 102, 122, 138 and 141.)

[82] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *Proceedings of the IEEE international conference on computer vision*, pp. 3628–3636, 2017. (Cited on pages 19 and 20.)

[83] H. Kuehne, A. Arslan, and T. Serre, "The language of actions: Recovering the syntax and semantics of goal-directed human activities," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 780–787, 2014. (Cited on pages 19, 21, 24, 26, 27, 34, 37, 54 and 64.)

[84] S. Stein and S. J. McKenna, "Combining embedded accelerometers with computer vision for recognizing food preparation activities," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 729–738, 2013. (Cited on pages 19, 21, 26, 34, 37 and 54.)

[85] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3339–3348, 2018. (Cited on pages 19, 20 and 106.)

[86] Y. Xiong, Y. Zhao, L. Wang, D. Lin, and X. Tang, "A pursuit of temporal accuracy in general activity detection," *arXiv preprint arXiv:1703.02716*, 2017. (Cited on page 20.)

[87] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018. (Cited on page 20.)

[88] D. Zhukov, J.-B. Alayrac, I. Laptev, and J. Sivic, "Learning actionness via long-range temporal order verification," in *European Conference on Computer Vision*, pp. 470–487, Springer, 2020. (Cited on page 20.)

[89] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "Bmn: Boundary-matching network for temporal action proposal generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3889–3898, 2019. (Cited on page 20.)

[90] T. Yu, Z. Ren, Y. Li, E. Yan, N. Xu, and J. Yuan, "Temporal structure mining for weakly supervised action detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5522–5531, 2019. (Cited on page 20.)

[91] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. (Cited on page 20.)

[92] G. Chen, C. Zhang, and Y. Zou, "Afnet: Temporal locality-aware network with dual structure for accurate and fast action detection," *IEEE Transactions on Multimedia*, 2020. (Cited on pages 20, 138 and 141.)

[93] J. Tan, J. Tang, L. Wang, and G. Wu, "Relaxed transformer decoders for direct action proposal generation," *arXiv preprint arXiv:2102.01894*, 2021. (Cited on pages 20, 56 and 60.)

[94] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020. (Cited on pages 20, 56 and 60.)

[95] R. Su, W. Ouyang, L. Zhou, and D. Xu, "Improving action localization by progressive cross-stream cooperation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12016–12025, 2019. (Cited on page 21.)

[96] Y. Huang, Y. Sugano, and Y. Sato, "Improving action segmentation via graph-based temporal reasoning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14024–14034, 2020. (Cited on pages 21, 54, 93 and 97.)

[97] S. Das, M. Thonnat, and F. Bremond, "Looking deeper into time for activities of daily living recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 498–507, 2020. (Cited on page 21.)

[98] R. De Geest and T. Tuytelaars, "Modeling temporal structure with lstm for online action detection," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1549–1557, IEEE, 2018. (Cited on pages 21 and 58.)

[99] F. Carrara, P. Elias, J. Sedmidubsky, and P. Zezula, "Lstm-based real-time action detection and prediction in human motion streams," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27309–27331, 2019. (Cited on pages 21 and 58.)

[100] L. Ding and C. Xu, "Weakly-supervised action segmentation with iterative soft boundary assignment," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on pages 21, 27, 54, 61, 65 and 66.)

[101] P. Lei and S. Todorovic, "Temporal deformable residual networks for action segmentation in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6742–6751, 2018. (Cited on page 21.)

[102] X. Dai, B. Singh, J. Y.-H. Ng, and L. Davis, "Tan: Temporal aggregation network for dense multi-label action recognition," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 151–160, IEEE, 2019. (Cited on pages 21, 58, 76, 77, 102, 115, 127, 132 and 138.)

[103] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016. (Cited on pages 21 and 141.)

[104] G. Hinton, "What is wrong with convolutional neural nets?," in *AAAI keynote speech*, 2017. (Cited on page 21.)

[105] F. Yi, H. Wen, and T. Jiang, "Asformer: Transformer for action segmentation," *arXiv preprint arXiv:2110.08568*, 2021. (Cited on page 21.)

[106] J. Zhao, Y. Zhang, X. Li, H. Chen, B. Shuai, M. Xu, C. Liu, K. Kundu, Y. Xiong, D. Modolo, *et al.*, "Tuber: Tubelet transformer for video action detection," in *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13598–13607, 2022. (Cited on page 21.)

[107] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017. (Cited on pages 21, 56, 59, 62, 68, 69, 78, 80 and 141.)

[108] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classificationârevisiting neural networks," in *Joint european conference on machine learning and knowledge discovery in databases*, pp. 437–452, Springer, 2014. (Cited on pages 22, 47, 71, 82, 120 and 121.)

[109] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, pp. 303–338, June 2010. (Cited on page 24.)

[110] G. Vaquette, A. Orcesi, L. Lucat, and C. Achard, "The daily home life activity dataset: a high semantic activity dataset for online recognition," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 497–504, IEEE, 2017. (Cited on pages 24, 27, 32, 34, 38, 54 and 64.)

[111] Y. Lei, U. Dogan, A. Binder, and M. Kloft, "Multi-class svms: From tighter data-dependent generalization bounds to novel algorithms," *Advances in neural information processing systems*, vol. 28, 2015. (Cited on page 25.)

[112] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE transactions on knowledge and data engineering*, vol. 23, no. 7, pp. 1079–1089, 2010. (Cited on page 25.)

[113] X.-Z. Wu and Z.-H. Zhou, "A unified view of multi-label performance measures," in *International Conference on Machine Learning*, pp. 3780–3788, PMLR, 2017. (Cited on page 25.)

[114] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta, "Asynchronous temporal fields for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 585–594, 2017. (Cited on pages 26, 27, 46, 50, 76, 94, 100 and 131.)

[115] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. (Cited on pages 27, 28 and 108.)

[116] "Department of economic and social affairs population of united nations, the world population aging report." https://www.un.org/en/development/desa/population/. Accessed Feb. 28th, 2020. (Cited on page 31.)

[117] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016. (Cited on pages 32, 34 and 36.)

[118] P. Weinzaepfel, X. Martin, and C. Schmid, "Human action localization with sparse spatial supervision," *arXiv preprint arXiv:1605.05197*, 2016. (Cited on pages 32, 34 and 36.)

[119] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik, "Ava: A video dataset of spatio-temporally localized atomic visual actions," *Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018. (Cited on pages 32, 34, 36, 115 and 129.)

[120] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," *arXiv preprint arXiv:1906.03327*, 2019. (Cited on pages 32, 34, 36 and 37.)

[121] "Kitware. the multiview extended video with activities (meva) dataset." http://mevadata.org/. Accessed Feb. 23th, 2020. (Cited on pages 34 and 36.)

[122] DARPA and Kitware, "Virat video dataset." http://www.viratdata.org/. Accessed Feb. 28th, 2020. (Cited on pages 34 and 36.)

[123] R. Sanabria, O. Caglayan, S. Palaskar, D. Elliott, L. Barrault, L. Specia, and F. Metze, "How2: a large-scale dataset for multimodal language understanding," in *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*, NeurIPS, 2018. (Cited on pages 34, 36 and 37.)

[124] Y. Tang, D. Ding, Y. Rao, Y. Zheng, D. Zhang, L. Zhao, J. Lu, and J. Zhou, "Coin: A large-scale dataset for comprehensive instructional video analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1207–1216, 2019. (Cited on pages 34 and 37.)

[125] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 2847–2854, IEEE, 2012. (Cited on pages 34 and 37.)

[126] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, "Charades-ego: A large-scale dataset of paired third and first person videos," *arXiv preprint arXiv:1804.09626*, 2018. (Cited on pages 34 and 37.)

[127] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *CVPR 2011*, pp. 3281–3288, IEEE, 2011. (Cited on pages 34 and 37.)

[128] D. Damen, H. Doughty, G. M. Farinella, , A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Rescaling egocentric vision," *CoRR*, vol. abs/2006.13256, 2020. (Cited on pages 34 and 37.)

[129] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele, "Recognizing fine-grained and composite activities using hand-centric features and script data," *International Journal of Computer Vision*, pp. 1–28, 2015. (Cited on pages 34 and 37.)

[130] K. Hema Swetha, G. Rudhir, and S. Ashutosh, "Learning human activities and object affordances from rgb-d videos," in *IJRR*, 2013. (Cited on pages 34 and 38.)

[131] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele, "Coherent multi-sentence video description with variable level of detail," in *German conference on pattern recognition*, pp. 184–195, Springer, 2014. (Cited on page 37.)

[132] N. Hu, G. Englebienne, Z. Lou, and B. Kröse, "Learning latent structure for activity recognition," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1048–1053, IEEE, 2014. (Cited on page 38.)

[133] M. Guo, E. Chou, D.-A. Huang, S. Song, S. Yeung, and L. Fei-Fei, "Neural graph matching networks for fewshot 3d action recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 653–669, 2018. (Cited on pages 38 and 143.)

[134] G. Rogez, P. Weinzaepfel, and C. Schmid, "LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. (Cited on pages 39, 131 and 134.)

[135] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. (Cited on pages 39 and 119.)

[136] M. P. Institute, "Tla software: Elan." `https://tla.mpi.nl/tools/tla-tools/elan/`. Accessed Oct. 30th, 2019. (Cited on page 41.)

[137] G. A. Sigurdsson, O. Russakovsky, and A. Gupta, "What actions are needed for understanding human actions in videos?," in *International Conference on Computer Vision (ICCV)*, 2017. (Cited on page 43.)

[138] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009. (Cited on pages 46 and 122.)

[139] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005. (Cited on pages 47, 48, 122 and 123.)

[140] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. (Cited on pages 47, 48, 122, 123 and 141.)

[141] B. Mahasseni and S. Todorovic, "Regularizing long short term memory with 3d human-skeleton sequences for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3054–3062, 2016. (Cited on pages 47 and 122.)

[142] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *ECCV 2016*, 2019. (Cited on pages 50, 61 and 119.)

[143] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "SST: Single-stream temporal action proposals," in *CVPR*, 2017. (Cited on page 54.)

[144] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao, "A multi-stream bi-directional recurrent neural network for fine-grained action detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1961–1970, 2016. (Cited on pages 54 and 58.)

[145] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *CVPR 2011*, pp. 3281–3288, IEEE, 2011. (Cited on page 54.)

[146] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. (Cited on pages 56, 60 and 89.)

[147] S. d'Ascoli, H. Touvron, M. Leavitt, A. Morcos, G. Biroli, and L. Sagun, "Convit: Improving vision transformers with soft convolutional inductive biases," *arXiv preprint arXiv:2103.10697*, 2021. (Cited on page 56.)

[148] J. Guo, K. Han, H. Wu, C. Xu, Y. Tang, C. Xu, and Y. Wang, "Cmt: Convolutional neural networks meet vision transformers," *arXiv preprint arXiv:2107.06263*, 2021. (Cited on page 56.)

[149] M. A. Islam, S. Jia, and N. D. Bruce, "How much position information do convolutional neural networks encode?," *arXiv preprint arXiv:2001.08248*, 2020. (Cited on pages 56 and 80.)

[150] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *arXiv preprint arXiv:1906.05909*, 2019. (Cited on page 59.)

[151] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019. (Cited on page 59.)

[152] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng, "Deepvit: Towards deeper vision transformer," *arXiv preprint arXiv:2103.11886*, 2021. (Cited on page 60.)

[153] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," *arXiv preprint arXiv:2102.12122*, 2021. (Cited on page 60.)

[154] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pvtv2: Improved baselines with pyramid vision transformer," *arXiv preprint arXiv:2106.13797*, 2021. (Cited on page 60.)

[155] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *arXiv preprint arXiv:2105.15203*, 2021. (Cited on page 60.)

[156] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *arXiv preprint arXiv:2107.06278*, 2021. (Cited on page 60.)

[157] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," *arXiv preprint arXiv:2106.13230*, 2021. (Cited on pages 60 and 141.)

[158] Y. Zhang, X. Li, C. Liu, B. Shuai, Y. Zhu, B. Brattoli, H. Chen, I. Marsic, and J. Tighe, "Vidtr: Video transformer without convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13577–13587, 2021. (Cited on page 60.)

[159] E. Kazakos, J. Huh, A. Nagrani, A. Zisserman, and D. Damen, "With a little help from my temporal context: Multimodal egocentric action recognition," in *British Machine Vision Conference (BMVC)*, 2021. (Cited on page 60.)

[160] M. Xu, Y. Xiong, H. Chen, X. Li, W. Xia, Z. Tu, and S. Soatto, "Long short-term transformer for online action detection," *Advances in Neural Information Processing Systems*, vol. 34, 2021. (Cited on page 60.)

[161] A. Chan-Hon-Tong, C. Achard, and L. Lucat, "Deeply optimized hough transform: Application to action segmentation," in *International Conference on Image Analysis and Processing*, pp. 51–60, Springer, 2013. (Cited on page 65.)

[162] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. (Cited on pages 72, 83, 101, 108, 122 and 132.)

[163] B. Fernando, C. Tan, and H. Bilen, "Weakly supervised gaussian networks for action detection," in *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020. (Cited on pages 76 and 138.)

[164] P. Ghosh, Y. Yao, L. S. Davis, and A. Divakaran, "Stacked spatio-temporal graph convolutional networks for action segmentation," *arXiv preprint arXiv:1811.10575*, 2018. (Cited on pages 76, 77 and 94.)

[165] J. Yuste Ramos, "Using deep learning for fine-grained action segmentation," *Treballs Finals de Grau Universitat de Barcelona*, 2021. (Cited on page 77.)

[166] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962. (Cited on page 78.)

[167] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological cybernetics*, vol. 20, no. 3, pp. 121–136, 1975. (Cited on page 78.)

[168] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3560–3569, 2021. (Cited on page 81.)

[169] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, 2018. (Cited on page 81.)

[170] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019. (Cited on page 82.)

[171] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018. (Cited on pages 93 and 94.)

[172] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018. (Cited on page 93.)

[173] T. Lan, Y. Zhu, A. R. Zamir, and S. Savarese, "Action recognition by hierarchical mid-level action elements," in *Proceedings of the IEEE international conference on computer vision*, pp. 4552–4560, 2015. (Cited on page 93.)

[174] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 399–417, 2018. (Cited on page 94.)

[175] Y. Zhang, X. Li, and I. Marsic, "Multi-label activity recognition using activity-specific features and activity correlations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14625–14635, 2021. (Cited on page 94.)

[176] L. B. Almeida, "C1. 2 multilayer perceptrons," *Handbook of Neural Computation C*, vol. 1, 1997. (Cited on page 96.)

[177] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016. (Cited on pages 97, 98 and 100.)

[178] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12026–12035, 2019. (Cited on pages 97 and 134.)

[179] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5177–5186, 2019. (Cited on pages 99 and 100.)

[180] Z. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Learning graph convolutional networks for multi-label recognition and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. (Cited on pages 99 and 100.)

[181] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Thirty-Second AAAI conference on artificial intelligence*, 2018. (Cited on page 100.)

[182] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori, "Object level visual reasoning in videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 105–121, 2018. (Cited on page 110.)

[183] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 284–293, 2019. (Cited on pages 109 and 110.)

[184] L. Wang and P. Koniusz, "Self-supervising action recognition by statistical moment and subspace descriptors," in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 4324–4333, 2021. (Cited on page 110.)

[185] S. Sudhakaran and O. Lanz, "Attention is all we need: Nailing down object-centric attention for egocentric activity recognition," *arXiv preprint arXiv:1807.11794*, 2018. (Cited on page 110.)

[186] S. Sudhakaran, S. Escalera, and O. Lanz, "Lsta: Long short-term attention for egocentric action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9954–9963, 2019. (Cited on page 110.)

[187] X. Wang, Y. Wu, L. Zhu, and Y. Yang, "Symbiotic attention with privileged information for egocentric action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12249–12256, 2020. (Cited on page 110.)

[188] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016. (Cited on page 111.)

[189] M. Zolfaghari, G. L. Oliveira, N. Sedaghat, and T. Brox, "Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2904–2913, 2017. (Cited on page 114.)

[190] S. Gupta, J. Hoffman, and J. Malik, "Cross modal distillation for supervision transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. (Cited on page 114.)

[191] F. Hafner, A. Bhuiyan, J. F. Kooij, and E. Granger, "A cross-modal distillation network for person re-identification in rgb-depth," *arXiv preprint arXiv:1810.11641*, 2018. (Cited on page 114.)

[192] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, "MARS: Motion-Augmented RGB Stream for Action Recognition," in *CVPR*, 2019. (Cited on pages 114, 118, 133 and 134.)

[193] N. C. Garcia, P. Morerio, and V. Murino, "Modality distillation with multiple stream networks for action recognition," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), (Cham), pp. 106–121, Springer International Publishing, 2018. (Cited on pages 114, 118 and 133.)

[194] N. C. Garcia, S. A. Bargal, V. Ablavsky, P. Morerio, V. Murino, and S. Sclaroff, "Distillation multiple choice learning for multimodal action recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2755–2764, January 2021. (Cited on pages 114 and 118.)

[195] Z. Luo, J.-T. Hsieh, L. Jiang, J. Carlos Niebles, and L. Fei-Fei, "Graph distillation for action detection with privileged modalities," in *The European Conference on Computer Vision (ECCV)*, September 2018. (Cited on pages 115, 118, 128, 132, 133, 136 and 137.)

[196] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Learning salient boundary feature for anchor-free temporal action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3320–3329, 2021. (Cited on page 116.)

[197] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pp. 1933–1941, IEEE, 2016. (Cited on page 116.)

[198] M. S. Ryoo, A. Piergiovanni, M. Tan, and A. Angelova, "Assemblenet: Searching for multi-stream neural connectivity in video architectures," in *International Conference on Learning Representations*, 2020. (Cited on page 117.)

[199] M. S. Ryoo, A. Piergiovanni, J. Kangaspunta, and A. Angelova, "Assemblenet++: Assembling modality representations via attention connections," in *ECCV*, 2020. (Cited on page 117.)

[200] A. Shahroudy, G. Wang, and T.-T. Ng, "Multi-modal feature fusion for action recognition in rgb-d sequences," in *2014 6th International Symposium on Communi-*

cations, Control and Signal Processing (ISCCSP), pp. 1–4, May 2014. (Cited on page 117.)

[201] G. Liu, J. Qian, F. Wen, X. Zhu, R. Ying, and P. Liu, "Action recognition based on 3d skeleton and rgb frame fusion," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 258–264, Nov 2019. (Cited on page 117.)

[202] H. Rahmani and M. Bennamoun, "Learning action recognition model from depth and skeleton videos," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5833–5842, Oct 2017. (Cited on page 117.)

[203] J.-M. Perez-Rua, V. Vielzeuf, S. Pateux, M. Baccouche, and F. Jurie, "Mfas: Multimodal fusion architecture search," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. (Cited on page 117.)

[204] S. Das, M. Thonnat, kaustubh Sakhalkar, M. Koperski, F. Brémond, and G. Francesca, "A new hybrid architecture for human activity recognition from rgb-d videos," *MultiMedia Modeling. MMM 2019*, 2019. (Cited on page 117.)

[205] F. Baradel, C. Wolf, and J. Mille, "Human activity recognition with pose-driven attention to rgb," in *The British Machine Vision Conference (BMVC)*, September 2018. (Cited on page 117.)

[206] F. Baradel, C. Wolf, and J. Mille, "Human action recognition: Pose-based attention draws focus to hands," in *proceedings of the IEEE International Conference on Computer Vision WorkshopS*, pp. 604–613, 2017. (Cited on page 117.)

[207] H. Duan, Y. Zhao, K. Chen, D. Shao, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," *arXiv preprint arXiv:2104.13586*, 2021. (Cited on page 117.)

[208] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," *CoRR*, vol. abs/1812.03982, 2018. (Cited on page 117.)

[209] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Epic-fusion: Audio-visual temporal binding for egocentric action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5492–5501, 2019. (Cited on page 117.)

[210] A. Bagchi, J. Mahmood, D. Fernandes, and R. K. Sarvadevabhatla, "Hear me out: Fusional approaches for audio augmented temporal action localization," *arXiv preprint arXiv:2106.14118*, 2021. (Cited on page 117.)

[211] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Slow-fast auditory streams for audio recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 855–859, IEEE, 2021. (Cited on page 117.)

[212] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. (Cited on page 118.)

[213] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 742–751, 2017. (Cited on page 118.)

[214] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3967–3976, 2019. (Cited on page 118.)

[215] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1365–1374, 2019. (Cited on page 118.)

[216] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2604–2613, 2019. (Cited on page 118.)

[217] A. Miech, J.-B. Alayrac, I. Laptev, J. Sivic, and A. Zisserman, "Thinking fast and slow: Efficient text-to-visual retrieval with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9826–9836, 2021. (Cited on page 118.)

[218] V. Gabeur, C. Sun, K. Alahari, and C. Schmid, "Multi-modal transformer for video retrieval," in *European Conference on Computer Vision*, pp. 214–229, Springer, 2020. (Cited on page 118.)

[219] C. You, N. Chen, and Y. Zou, "Mrd-net: Multi-modal residual knowledge distillation for spoken question answering," in *IJCAI*, 2021. (Cited on page 118.)

[220] Z. Yang, J. Liu, J. Huang, X. He, T. Mei, C. Xu, and J. Luo, "Cross-modal contrastive distillation for instructional activity anticipation," *arXiv preprint arXiv:2201.06734*, 2022. (Cited on page 118.)

[221] G. A. Sigurdsson, O. Russakovsky, and A. Gupta, "What actions are needed for understanding human actions in videos?," in *Proceedings of the IEEE international conference on computer vision*, pp. 2137–2146, 2017. (Cited on page 126.)

[222] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *International Conference on Learning Representations*, 2020. (Cited on page 128.)

[223] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020. (Cited on page 128.)

[224] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *Adv. Neural Inform. Process. Syst.*, 2018. (Cited on page 128.)

[225] J. Sanchez Perez, E. Meinhardt-Llopis, and G. Facciolo, "TV-L1 Optical Flow Estimation," *Image Processing On Line*, vol. 3, pp. 137–150, 2013. (Cited on pages 131 and 134.)

[226] J. Hoffman, S. Gupta, and T. Darrell, "Learning with side information through modality hallucination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 826–834, 2016. (Cited on page 133.)

[227] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5005–5013, 2016. (Cited on page 133.)

[228] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *CVPR*, 2019. (Cited on page 133.)

[229] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2684–2701, 2019. (Cited on page 134.)

[230] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding and classification for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2593–2600, 2014. (Cited on page 134.)

[231] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8934–8943, 2018. (Cited on page 134.)

[232] W. Wang, D. Tran, and M. Feiszli, "What makes training multi-modal networks hard?," *CoRR*, vol. abs/1905.12681, 2019. (Cited on page 135.)

[233] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *European Conference on Computer Vision*, pp. 203–220, Springer, 2016. (Cited on page 137.)

[234] Bo Li, Huahui Chen, Yucheng Chen, Yuchao Dai, and Mingyi He, "Skeleton boxes: Solving skeleton based action detection with a single deep convolutional neural network," in *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 613–616, 2017. (Cited on page 137.)

[235] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1290–1297, IEEE, 2012. (Cited on page 137.)

[236] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 597–600, IEEE, 2017. (Cited on pages 136 and 137.)

[237] Z. Qin and C. R. Shelton, "Event detection in continuous video: An inference in point process approach," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5680–5691, 2017. (Cited on page 137.)

[238] H. Alwassel, S. Giancola, and B. Ghanem, "Tsp: Temporally-sensitive pretraining of video encoders for localization tasks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3173–3183, 2021. (Cited on page 141.)

[239] M. Xu, J.-M. Pérez-Rúa, V. Escorcia, B. Martinez, X. Zhu, L. Zhang, B. Ghanem, and T. Xiang, "Boundary-sensitive pre-training for temporal localization in videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7220–7230, 2021. (Cited on page 141.)

[240] M. Xu, J.-M. Perez-Rua, X. Zhu, B. Ghanem, and B. Martinez, "Low-fidelity end-to-end video encoder pre-training for temporal action localization," *arXiv preprint arXiv:2103.15233*, 2021. (Cited on page 141.)

[241] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv preprint arXiv:2111.06377*, 2021. (Cited on page 141.)

[242] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6752–6761, 2018. (Cited on page 142.)

[243] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *International Conference on Machine Learning*, pp. 6438–6447, PMLR, 2019. (Cited on page 143.)