A New Hybrid Architecture for Human Activity Recognition from RGB-D videos

Srijan Das, Monique Thonnat, Kaustubh Sakhalkar, Michal F Koperski, Francois Bremond, Gianpiero Francesca

A New Hybrid Architecture for Human Activity Recognition from RGB-D videos

Srijan Das[1], Monique Thonnat[1], Kaustubh Sakhalkar[1], Michal Koperski[1], Francois Bremond[1], and Gianpiero Francesca[2]

[1] INRIA, Sophia Antipolis, 2004 Rte des Lucioles, 06902, Valbonne, France
name.surname@inria.fr
[2] Toyota Motor Europe, Hoge Wei 33, B - 1930 Zaventem
gianpiero.francesca@toyota-europe.com

Abstract. Activity Recognition from RGB-D videos is still an open problem due to the presence of large varieties of actions. In this work, we propose a new architecture by mixing a high level handcrafted strategy and machine learning techniques. We propose a novel two level fusion strategy to combine features from different cues to address the problem of large variety of actions. As similar actions are common in daily living activities, we also propose a mechanism for similar action discrimination. We validate our approach on four public datasets, CAD-60, CAD-120, MSRDailyActivity3D, and NTU-RGB+D improving the state-of-the-art results on them.

Keywords: activity recognition · RGB-D videos · data fusion.

1 Introduction

Action Recognition has been a popular problem statement in the vision community because of its large scale applications. In this paper, we focus on Activities of Daily Living (ADL) which can be used for monitoring hospital patients, smarthome applications and so on. We propose a new architecture aiming to be effective and efficient for ADL recognition from RGB-D videos. ADL recognition includes challenges such as viewpoint changes, occlusions, same environment and similar actions. Over time, with the development of technology, features used for action recognition have taken new strides from computing simple SIFT features to deep CNN features. The emergence of deep learning, inspired the authors in [13, 10] to use CNN features for modeling the appearance of actions in video sequences. The introduction of cheap kinect sensors motivated the researchers to use 3 dimensional information of human poses to exploit the human skeleton geometry [23, 16]. Our approach leverages the advantages of using handcrafted features along with features from deep networks. Compared to object detection, action recognition involves encoding object information involved in the action, pose information of the subject performing the action and their motion. Time is also an important factor in this problem domain. Spatio-temporal contextual association is an important challenge to be explored. The diversity of actions

in ADL makes the problem of action recognition complex. This problem can be solved by using different visual cues as in [17, 24] where each cue is responsible for modeling actions of specific categories. Current approaches using multiple visual cues fail to achieve high performance rate and consistency in modeling the actions.
In this work, we propose an answer to the following questions:

1. Which visual cue is effective for which action?
2. How these visual cues should be combined in order to mitigate the disadvantages of each cue?
3. How to disambiguate similar actions?

In the following we will focus on three types of visual cues: appearance, pose and short-term motion. We propose **a novel two-level fusion strategy** to combine the features in a common feature space to appropriately model the actions. We also address the challenge of recognizing similar actions in daily living activities by proposing a mechanism for **similar action discrimination**.

2 Related Work on Action Recognition

Handcrafted Approaches- Earlier approaches on action recognition are based on extracting handcrafted features frame by frame and aggregating them to form a global representation of the video. Wang et al. in [19] propose to compute local descriptors around the dense trajectories to recognize actions and further improve the technique in [20] by subtracting the camera motion. These local descriptors are used with fisher vector encoding so as to have fixed size video descriptors. Handcrafted approaches demand resources in terms of time and expertise but at the same time they successfully capture the local temporal structure of the actions in the videos.

CNN based Approaches- Following the breakthrough of convolutional neural networks (CNN) on object recognition [13], it is natural to extend them for videos. Early models extract CNN features from video frames and aggregates them with pooling for classifying by SVM. The authors in [5, 8] use different body part patches to extract features from a convolutional network in order to recognize actions. The requirement to introduce spatio-temporal relationship in videos motivated the authors in [4] to use 3D convolutions. They use convolutional inflation in 2D networks expanding it to 3D. Such deep architectures successfully model the appearance but fail to model long-term motion. This motivates us to use such architectures to encode the color statistics.

RNN based Approaches- RNNs being sequential models capture temporal information. In [9] temporal information is encoded using input from fc6 layer of convolutional network. With the advancement in camera technologies now, it is possible to get more accurate information from the scene including depth of the scene with the help of cameras like RGB-D sensors along with skeleton joints information. This motivates the authors in [16, 7, 23] to utilize 3D human geometry of the subject performing action using RNNs. LSTMs (special kind of RNN) being capable of understanding the human dynamics can model the

pose based motion in a video. Such sequence models including variants like [16, 7] have shown to successfully encode long-term temporal information which is an important aspect for recognizing ADL.

Multi-stream Fusion based Approaches- It can be concluded from the aforementioned approaches that we need pose based motion, short term motion as well as appearance information for robust action recognition. The strategy of combining appearance and motion features in an early stage before classification as in [17, 5, 24] has been popular. This is because appearance and motion are complementary and their early fusion utilizes the correlation between features from different modalities. Thus making them more discriminative in common feature space rather than their individual feature space. The use of different modalities via a Markov chaining is proposed in [24]. The authors in [24] use pose, appearance and motion, fusing them in order to have a sequential refinement of action labels. But the drawback of such chaining models includes mutual dependence of the visual cues used for action classification. The existing studies on action recognition show the diversity of approaches and information used. This gives us a hint of different visual cues for modeling the actions along with eliminating the mutual dependence among them. Understanding the pose, appearance and motion of the subject performing the action in a video is important for action recognition. Thus, we focus on combining the pros of different visual cues with a learning strategy optimized for modeling ADL.

3 Feature Relevance depending on Action types

ADL consists of high variation of actions categories ranging from actions with similar poses like *stacking and unstacking objects, rubbing two hands and clapping,* actions with low motion like *typing keyboard, relaxing on couch,* and actions having temporal evolution of body dynamics like *walking, falling down* and so on. For optimizing action recognition it is important to establish a proper relationship between the nature of features and action categories to be modeled. For ADL, features corresponding to mainly three types of visual cues are widely used in the literature, say

- **appearance** modeling the spatial layout of the action videos from convolutional neural networks.
- **short-term motion** which is often computed through optical flow for instantaneous motion or based on short-term tracklets as in dense trajectories [19, 20].
- **pose based motion** obtained from recurrent neural networks modeling the temporal evolution of 3D human body dynamics.

In table 1, we show the importance of appearance based features for action recognition. We use the average number of local features of some actions from [11] to describe the motion of the actions. The 3^{rd} column in table 1 shows the difference in classification accuracy using appearance and short-term motion features (where $D = Accuracy$(Appearance)-$Accuracy$(Motion)). In fig. 1, we show a

comparison of action recognition accuracy for some actions using short-term and pose based motion. For dense trajectories, we do not use the HOG features (for this figure only) in order to neglect appearance and have a fair comparison with pose based motion features from LSTM. In spite of both features modeling the motion, the statistics in fig. 1 shows the complementary nature of both the features and their relevance with temporal dynamics of the subject performing action.

Now, the remaining question is how to combine the features to take advantages from each visual cue? Early fusion is preferred when all the features characterize the actions because the correlation between them materialize in a precise level. If not, it is better to compute late fusion in order to balance the feature weights at the latest stage. So, in the next section we propose a two level fusion strategy to combine features at the most appropriate level depending on action categories.

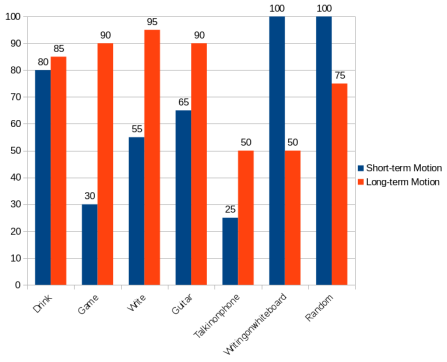

Fig. 1: Comparison of action recognition accuracy using short-term and pose based motion. Short-term motion is modeled by dense trajectories [19] and pose based motion is modeled by LSTM [7].

Action	Number of features	D
Relaxing on couch	1346	+100 %
Working on computer	1356	+50%
Still	1510	+75%
Talking on couch	2060	+50%
Drinking water	3079	-50%
Cooking (chopping)	4448	0%
Cooking (Stirring)	4961	0%
Brushing teeth	5527	-25%

Table 1: Comparison of action recognition based on appearance and motion. The table shows average number of detected features using Dense Trajectories [19] taken from [11]. Third Column shows clear importance of appearance with little motion.

4 Proposed Architecture for Action Recognition

In the following first, we describe the two level fusion strategy then we explain how to disambiguate similar actions. Fig. 2 shows the overall architecture for the testing phase.

4.1 Two-level Fusion Strategy

The first level of fusion (early) is intended to combine features in a balanced way to address actions which are characterized by most of the features. The second level of fusion (late) puts more emphasize on selection of features which are characterizing specific actions in a prominent manner.
For early fusion, we concatenate appearance (F_1) and short-term motion (F_2) leading to $F_x = [F_1, F_2]$ because they are often highly correlated. For late fusion, we put more importance on pose based motion because this feature is very complementary to the previous ones. Temporal information from poses is not discriminative for all the actions, so fusing temporal information at an early stage adds noise to the classifier. For actions like *relaxing on couch*, *talking on phone*, *writing on whiteboard* and so on temporal information may not be important. Thus encoding the vector which is representative of time in a video to a common feature space along with appearance and motion leads to common feature space where the actions are not discriminative. Thus we propose to fuse the pose based motion (F_3) features using a late fusion strategy where the fusion focuses on the individual strength of modalities.
In the two-level fusion strategy, the fused representation of appearance and

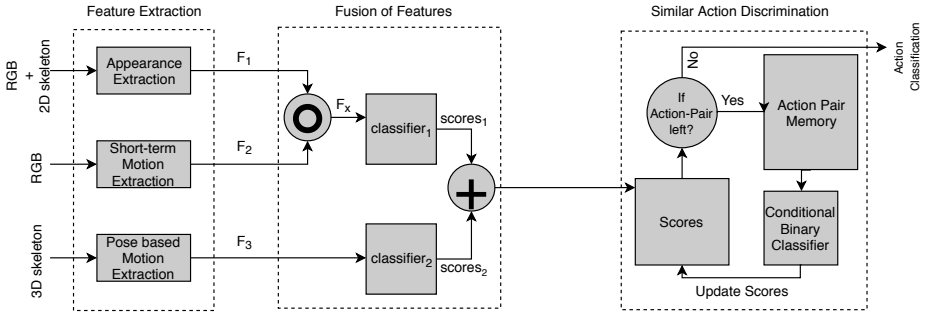

Fig. 2: Big picture of the architecture proposed to combine the features with two-level fusion strategy for the testing phase. The action-pair memory module keeps track of action pairs with high similarities. Such action pairs are forwarded to binary classifier to disambiguate the similar actions.

motion of a video F_x and the pose based motion representation of a video F_3 is input to two linear SVM classifiers. Classifiers clf_1 and clf_2 learn the mapping $\mathbb{X} \rightarrow \mathbb{Y}$, where $F_x \in \mathbb{X}$ for clf_1, $F_3 \in \mathbb{X}$ for clf_2 and $y \in \mathbb{Y}$ is a class label. For a given SVM parameter θ, the algorithm performs a parameter search on a large number of SVM parameter combinations to obtain the optimal value θ^*. So, θ_1^* and θ_2^* are the optimal SVM parameter of clf_1 and clf_2 respectively. The second level of fusion is performed on the test set by fusing the classification

scores of the respective classifiers. For this, we introduce a fusion parameter α to balance the visual cues; α ranging between [0,1]. Let $scores_1 = P(y|F_x, \theta_1^*)$ and $scores_2 = P(y|F_3, \theta_2^*)$ be the classification scores computed by clf_1 and clf_2 respectively (see fig. 2). Then the second level of fusion is performed by computing the action classification score s.

$$s = \alpha P(y|F_x, \theta_1^*) + (1 - \alpha)P(y|F_3, \theta_2^*) \tag{1}$$

A small value of α means that the temporal information is the dominant visual cue. Thanks to the fusion strategy, an optimized pool of features is extracted to feed the classifiers dedicated to the different action categories. See section 6.2 for hyper-parameter α setting.

4.2 Similar Action Discrimination

Daily living action datasets contain similar actions like *stacking, unstacking objects; cleaning objects, taking food* and so on. Thus the classifier misclassifies similar action types and degrades its performance. So, we propose a mechanism for similar action discrimination consisting of a memory module and a binary classifier. The objective is to disambiguate similar actions by exploiting their predicted scores from the fusion phase. In the training stage, the algorithm checks the confused pair of actions in the fused scores of the cross-validation set. Let C be the confusion matrix of the actions classified in the validation set and a_r represents the action r, then the algorithm checks the false positives from C. If $C(i,j) + C(j,i) \geq \epsilon$ with $i \neq j$, then action a_i and a_j are misclassified. The action pair memory module depicted in fig. 2 keeps a track of these action pairs in descending order of misclassification score in the validation step. The last level of classifier is a binary classifier to classify the actions (a_i, a_j) with similar gestures. Handling ambiguities through binary classifier consists in combining a selection of features dedicated to selection of small set of ambiguous actions which are very similar to each other. Because these actions may have similar motion, pose or temporal dynamics, different combination of features are used to classify the two ambiguous actions. Thus the action-pair memory module keeps track of which features to use or fuse for disambiguating the similar actions in the validation set. The feature or combination of features with maximum classification accuracy in the validation set is recorded in the action pair memory module. In the training phase, the action-pair memory module learns to record the similar action pairs along with the entity of features required to disambiguate them by a greedy approach from the cross-validation. See section 6.2 for hyper-parameter ϵ setting.

In the testing phase, the classification scores are generated from the fusion phase (scores from the late fusion). The video samples with predicted labels from the scores obtained if present in the action pair module, are classified by a conditional binary classifier using the features mentioned in the action-pair memory module. The final classification score is updated from the classification score of the binary classifier and the same process is repeated unless all the confused action pairs undergo binary classification. This finite looping of discriminating

similar actions in a binary classifier is bounded by the number of action-pairs recorded in the action-pair memory module in terms of time complexity. This strategy of employing conditional binary classifier is capable of discriminating similar actions which is a challenge in daily living applications.

5 Implementation Details

Feature Extraction - For *appearance extraction*, we use 2D convolutional features (from ResNet-152 pre-trained on ImageNet) from different body regions (cropped using pose information from Depth) of the subject as in [5]. In the case of availability of large training database, we also use 3D convolutional features from I3D [4] network. We use the strategy of selecting the most salient body part based features by employing a feature selection mechanism as in [7]. For *short-term motion extraction*, we use improved dense trajectories toolbox provided in [20]. Fisher vector representation of a video is obtained from its frame-level features using standard Mixture of Gaussians (MoG) model as described in [12]. For *pose based motion extraction*, we build a 3 layered stacked LSTM framework on the platform of keras toolbox [6] with TensorFlow [1]. Adam optimizer initialized with learning rate 0.005 is used to train the network. Parameters like Dropout, gradient clipping, number of neurons in each LSTM layer for each dataset are used as in [7]. The latent temporal representation of the skeleton sequence is extracted from the trained LSTM which is a concatenated feature vector of the output hidden states of the LSTM from each time step.
Fusion of Features - For $classfier_1$ and $classifier_2$, we use scikit-learn [15] implementation of SVM.
Similar Action Discrimination - This stage of disambiguating similar actions is implemented in *Python* with a scikit-learn [15] implementation of SVM for the binary classifier.

6 Experimental Analysis

6.1 Dataset Description

As discussed in the introduction, we are interested in daily living action recognition due to their application in health care and robotics. So, we have selected 4 public datasets which contain daily living actions to evaluate our architecture.
CAD-60 [18] - contains 60 RGB-D videos with 4 subjects performing 14 actions each. These actions are performed in 5 different environments: office, kitchen, bedroom, bathroom and living room.
CAD-120 [18] - contains 120 RGB-D videos with 4 different subjects performing 10 high level activities. Each action is repeated thrice with different objects. Actions with similar motion in this dataset make it more challenging.
MSRDailyActivity3D [21] - contains 320 RGB-D videos with 10 subjects performing 16 actions.
NTURGB+D [16] - contains 56880 RGB-D videos with 40 subjects performing 60 different actions. Samples are captured from 17 camera setups.
The standard evaluations on these datasets include Cross-Subject evaluation

where the training and testing split is made either by leave-one-person out schema or split mentioned in the dataset (as in NTURGB+D). We are not focusing on Cross-View problem. Hence, we have not evaluated cross-view accuracy on NTURGB+D dataset.

6.2 Hyper-parameter setting

Parameter α responsible for score fusion of classifiers clf_1 and clf_2 is trained in the Fusion of Features phase. This is done by globally searching the best value of α ranging between [0,1] for which the cross-validation data yields maximum action classification accuracy in the training phase. This trained α is used for testing. Parameter ϵ used for selecting confused action-pairs is handcrafted. Its value depends on the action categories present in the training samples. The value of ϵ is set manually in function of the confusion matrix during training of the second level fusion stage. The value of ϵ ranges from 0.1 for NTU-RGB+D to 0.44 for CAD-120.

6.3 Qualitative Results

In this section, we perform a qualitative evaluation of our two-level fusion strategy by visualizing the high dimensional data using t-SNE tool [14]. For instance in fig. 3, we visualize the actions *drink* and *sitdown* using short-term motion, appearance, and their combination. From the figure, it is clear that the action groups are visually more discriminative using their combination. This depicts the effectiveness of using common feature space for appearance and short-term motion.

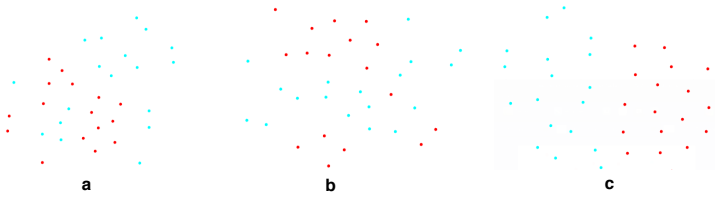

Fig. 3: t-SNE [14] representation of *drink* (in *red*) and *sitdown* (in *blue*) action using (a)short-term motion only (1^{st} column), (b)appearance only (2^{nd} column) and (c)both appearance and short-term motion (3^{rd} column) where the actions are more discriminative as compared to their individual feature space.

6.4 Quantitative Results

In this section, we report the action classification scores of the individual features along with their combination. Table 2 reports the action classification accuracy on three datasets CAD-60, CAD-120 and MSRDailyActivity3D using appearance, short-term and pose based motion. The performance obtained using different features are very data-dependent. For example, we get better results

Dataset	F_1 (2D-CNN)	F_2 (IDT)	F_3 (LSTM)	$F_1 + F_2$	$F_1 + F_2 + F_3$ Early Fusion	Proposed Fusion
CAD-60	**89.70**	72.05	67.64	95.58	70.58	**98.53**
CAD-120	72.58	**79.84**	63.70	83.06	63.70	**87.90**
MSR3D	80.93	81.87	**91.56**	90	91.56	**97.81**

Table 2: Ablation study on how each feature performs individually and with different combination techniques for action classification on CAD-60, CAD-120 and MSRDailyActivity3D. In early fusion, we fused all the features with $l2-$ normalization and proposed fusion is our two-level fusion strategy. $MSR3D$ signifies MSRDailyActivity3D, F_1 is appearance, F_2 is short-term motion and F_3 is pose based motion.

on MSRDailyActivity3D using pose based motion, CAD-120 using short-term motion and CAD-60 using appearance features. Table 2 shows the importance of using the two-level fusion scheme which takes into account the advantages of all features by performing a late fusion of appearance, short-term motion with pose based motion. This is shown by comparing our fusion strategy with naive early fusion of all features. Our proposed fusion outperforms the former as depicted in table 2.

6.5 Effect of using the mechanism of Similar Action Discrimination

This section presents an ablation study on the similar action discrimination mechanism and how the action-pair module works. In table 3, we show the confused actions with their corresponding misclassification rate in CAD-120 for every subject based splits. The action-pair module keeps a track of the confusing actions which are classified separately in a binary classifier which is also a linear SVM. For CAD-120, IDT+FV (short-term motion along with appearance because of presence of the HOG) discriminates the confused action pairs with 100 % accuracy. The drawback of this module includes its thorough dependency on cross-validation set. This drawback is depicted in table 3 where the cross-validation fails to capture confused action pairs like *cleaning objects and taking food* (in 3^{rd} row, left). Table 3 reports the action classification accuracy on all the datasets used before and after applying the action-pair module. This module does not have any effect on CAD-60 and MSRDailyActivity3D on which the actions are already classified with remarkable accuracy.

6.6 State-of-the-art comparison

In this section, we compare our action classification performance with the state-of-the-art. Our proposed two-level fusion along with action-pair module outperforms the existing methods on all the datasets as described in table 4.
NTU-RGB+D is a relatively large dataset and is suitable for using deeper models. In order to show the robustness of our framework, we use I3D [4] to model the appearance instead of using 2D CNN [7] and report **92.2**% accuracy (illustrated by $ProposedMethod + I3D$). This performance boosting is because I3D

split	Action Pairs	$C(i,j)$+ C(j,i)
1	*cleaning object* and *taking food*	0.44
1	*stacking and unstacking objects*	0.67
2	*cleaning object* and *taking food*	0.66
2	*stacking* and *unstacking objects*	0.66
3	*stacking* and *unstacking objects*	0.55
4	*cleaning object* and *taking food*	0.55
4	*stacking* and *unstacking objects*	0.44

Dataset	Acc. before binary classifier	Acc. after binary classifier
CAD-60	98.52 %	98.52 %
CAD-120	87.90%	94.40 %
MSR3D	97.81%	97.81 %
NTU-RGB+D	84.95 %	87.09 %

Table 3: Action-pair memory content for different splits in CAD-120 (on $left$). Each split signifies cross-actor setup for classification evaluation. The second column represents the action pairs confused among each other with their summation of mis-classification accuracy in third column (in validation set). Improvement in action classification accuracy on using conditional binary classifier for all the datasets used (on $right$). $MSR3D$ signifies MSRDailyActivity3D.

can model better appearance information (90.4%) for large available data than 2D CNN architecture.

6.7 Runtime Analysis

The fully automated architecture has been trained on two GTX 1080 Ti GPUs (each for extracting RGB based video descriptors from CNN network and training LSTM on skeleton sequences) and a single CPU (for extracting IDT features with fisher vector encoding) in parallel. IDT being computationally expensive (with a processing speed of less than 4 fps) decides the computational time involved in the feature extraction process. The proposed architecture including the fusion strategy along with the action-pair module only takes as additional cost 10 ms time delay for a forward pass of an image frame on a single CPU.

7 Conclusion

In this paper, we have proposed a new architecture for action recognition mixing a high level fusion strategy and machine learning techniques. The proposed hybrid architecture is fully automated enabling the hyper-paramaters except ϵ to learn themselves. We justify the use of this two-level fusion mechanism by qualitative and quantitative analysis. We also propose an action-pair memory module to disambiguate similar actions. Our proposed effective and efficient action recognition architecture improves the state-of-the-art on four publicly available datasets.

We emphasize the fact that the existing features are quite capable of distinguishing the daily living activities if combined in a strategic way. The quality of recognition rate achieved in this work ranging from 87 % to 98% is satisfactory. A future direction of this work can be to eliminate the handcrafted use of ϵ to record the confused action pairs. This can be done by a technique of regression on the confusion matrix in the training phase.

Method	Accuracy [%]
CAD-60	
Object Affordance	71.40
HON4D	72.70
Actionlet Ensemble	74.70
MSLF	80.36
JOULE-SVM	84.10
P-CNN + kinect + Pose machines	95.58
Proposed Method	**98.52**
CAD-120	
Salient Proto-Objects	78.20
TDD	80.38
SVM + CNN	78.30
STS	84.20
Object Affordance	84.70
MSLF	85.48
R-HCRF	89.80
RSVM + LCNN	90.10
Proposed Method	**94.40**

Method	Accuracy [%]
MSRDailyActivity3D	
Actionlet Ensemble	85.80
RGGP + fusion	85.60
MSLF	85.95
DCSF + joint	88.20
JOULE-SVM	95.00
Range Sample	95.60
DSSCA-SSLM	97.50
Proposed Method	**97.81**
NTU-RGB+D	
Geometric features [23]	70.26
VA-LSTM [22]	79.4
CMN [24]	80.8
STA-hands [2]	82.5
Glimpse Clouds [3]	86.6
Proposed Method	**87.09**
Proposed Method (with I3D)	**92.20**

Table 4: Recognition Accuracy comparison for CAD-60 , CAD-120, MSRDailyActivity3D (Performance of baseline is taken from [8, 12, 7] respectively) and NTU-RGB+D dataset.

References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. Baradel, F., Wolf, C., Mille, J.: Human action recognition: Pose-based attention draws focus to hands. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 604–613 (Oct 2017)
3. Baradel, F., Wolf, C., Mille, J., Taylor, G.W.: Glimpse clouds: Human activity recognition from unstructured feature points. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
4. Carreira, J., Zisserman, A.: 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)
5. Cheron, G., Laptev, I., Schmid, C.: P-cnn: Pose-based cnn features for action recognition. In: ICCV (2015)
6. Chollet, F., et al.: Keras. https://github.com/fchollet/keras (2015)
7. Das, S., Koperski, M., Bremond, F., Francesca, G.: A Fusion of Appearance based CNNs and Temporal evolution of Skeleton with LSTM for Daily Living Action Recognition. ArXiv e-prints (Feb 2018)
8. Das, S., Koperski, M., Bremond, F., Francesca, G.: Action recognition based on a mixture of rgb and depth based skeleton. In: AVSS (2017)
9. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual

recognition and description. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
10. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-Scale Video Classification with Convolutional Neural Networks. In: CVPR (2014)
11. Koperski, M.: Human Action Recognition in videos with Local Representation. Ph.D. thesis, University COTE DAZUR (2017)
12. Koperski, M., Bremond, F.: Modeling spatial layout of features for real world scenario rgb-d action recognition. In: AVSS (2016)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
14. van der Maaten, L., Hinton, G.E.: Visualizing data using t-sne (2008), https://lvdmaaten.github.io/tsne/
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
16. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: 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)
17. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
18. Sung, J., Ponce, C., Selman, B., Saxena, A.: Unstructured human activity detection from rgbd images. In: ICRA (2012)
19. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action Recognition by Dense Trajectories. In: IEEE Conference on Computer Vision & Pattern Recognition. pp. 3169–3176. Colorado Springs, United States (Jun 2011)
20. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: IEEE International Conference on Computer Vision. Sydney, Australia (2013)
21. Wu, Y.: Mining actionlet ensemble for action recognition with depth cameras. In: CVPR (2012)
22. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
23. Zhang, S., Liu, X., Xiao, J.: On geometric features for skeleton-based action recognition using multilayer lstm networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 148–157 (March 2017)
24. Zolfaghari, M., Oliveira, G.L., Sedaghat, N., Brox, T.: Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 2923–2932. IEEE (2017)