An Extensible Deep Architecture for Action Recognition Problem

Isaac Sanou, Donatello Conte, and Hubert Cardot
[1]*LiFAT, EA 6300, Université de Tours,*
64 Avenue Jean Portalis, 37200, Tours, France

Keywords: Human Action Recognition, Deep Learning, 3D Convolution, Model Extensible.

Abstract: Human action Recognition has been extensively addressed by deep learning. However, the problem is still open and many deep learning architectures show some limits, such as extracting redundant spatio-temporal informations, using hand-crafted features, and instability of proposed networks on different datasets. In this paper, we present a general method of deep learning for the human action recognition. This model fits on any type of database and we apply it on CAD-120 which is a complex dataset. Our model thus clearly improves in two aspects. The first aspect is on the redundant informations and the second one is the generality and the multi-functionality application of our deep architecture. Our model uses only raw data for human action recognition and the approach achieves state-of-the-art action classification performance.

1 INTRODUCTION

The recognition of human actions has been a subject of research since the early 1980s, because of its promise in many fields of application (Wang and Schmid, 2013), (Gan et al., 2015). We define gestures like the basic components that describe the meaning of movements. "Raising an arm" and "shaking the supports" are movements in this category. So actions are one-person activities that can be composed of several gestures organized over time. "Walking", "shaking" and "drinking" are examples of simple actions. Then activities are complex sequences of actions performed by many people or objects. "Playing basketball" is an example of activity consisting of actions such as running, shooting, dribbling and objects. Compared with object recognition in images, human action recognition is much more difficult because action contains spatio-temporal informations.

To recognize actions, there are some methods called traditional methods (Laptev, 2005) that consist in three parts: feature extraction, feature coding to generate video-level feature descriptor and the descriptor classification. The main features for action recognition using hand-crafted features are histogram of oriented gradients (HOG), histograms of optical flow (HOF), motion boundary histogram (MBH). Hand-crafted features have achieved great success in the task of action recognition. However, over the last few years, deep convolutional neural networks (CNN) has become the most popular method and achieved the state-of-the-art performance on several datasets. The CNNs offer us the possibility to extract the features instead of the classical method commonly used in computer vision problems. Impressive results have been obtained using convolution networks which is an extension of 2D CNN (LeCun et al., 1998) for the recognition of human action in video. Precisely 3-dimensional convolutional neural networks are used. The usage of 3D convolutions allows to capture spatial information in time from the video data stream by taking consecutive video frames into account. There is a lot of work for human action recognition using 3D CNNs as two-stream convolutional networks (Simonyan and Zisserman, 2014), stacked convolutional independent subspace analysis (Le et al., 2011), stratified pooling based deep convolutional neural networks (Yu et al., 2017), trajectory-pooled deep convolutional descriptors (Wang et al., 2015), etc. In contrast to image classification, there are more aspects to be consider when using video data. Firstly, the video-based CNN weights must be trained on video dataset from scratch while to image-based CNN can profit from transfer learning. Secondly the semantic understanding becomes more complex for example when the camera moves or when the background changes in time. Thirdly, the network takes more time to train depending on the architecture. The work in (Wang et al., 2014) shows good results on the CAD-120 dataset but the model is not extensible on another dataset where they get relatively low score. Still in the work (Zolfaghari et al., 2017), the authors use many

Sanou, I., Conte, D. and Cardot, H.
An Extensible Deep Architecture for Action Recognition Problem.
DOI: 10.5220/0007253301910199
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 191-199
ISBN: 978-989-758-354-4

inputs for their models but these inputs are sometimes using hand-crafted features. Theses methods can have temporal informations but have an impact on the overall performance. In practice, due to the difficulty in data collection and annotation, publicly available action recognition datasets (e.g. UCF101 (Soomro et al., 2012), HMDB51 (Kuehne et al., 2011)) remain limited, in both size and diversity.

To tackle the above problems, we study:

- How to design an effective and efficient video-level framework for learning video representation that is able to capture long-range temporal structure without using, in addition, hand-crafted features;
- How to learn a model given limited training data;
- How to generalize a model to any databases.

Our proposition consists in a new deep architecture for human actions recognition by video analysis, precisely with RGB images. To achieve this task, it is important to have both spatial and temporal informations. We linearly segment our video to avoid having a lot of redundant information. Our model uses 3D convolutions layers and max pooling to perform operations through a stack of images, movement can be captured in the resulting entities. In summary, we build our method on top of the successful Temporal Segment Networks (Wang et al., 2016) and Reconfigurable Convolutional Neural Networks (Wang et al., 2014) while tackling the problems mentioned above. Our method proposed a new deep architecture for human action recognition which transforms frame-level features to video-level feature descriptor. Firstly, we extract short snippets over a long video sequence by linearly segmenting each video called clique. Secondly, a clique is a part of our final architecture and contains 3D CNN which includes some fully-connected layers to extract features. Thirdly, features from each clique are concatenate to form a final video-level feature descriptor. Finally, we use full connection layers for action classification.

The main contributions in this study are summarized as follows:

(i) The uniform temporal sampling method preserves enough video information for action recognition. Furthermore, the network does not need hand-crafted features dataset during the training process just raw data for each clique. Then we obtain significant improvement on CAD-120 database, which only contains 120 video clips.

(ii) The proposed method allows an arbitrary number of cliques so any number of features to be aggregated into video-level feature descriptor.

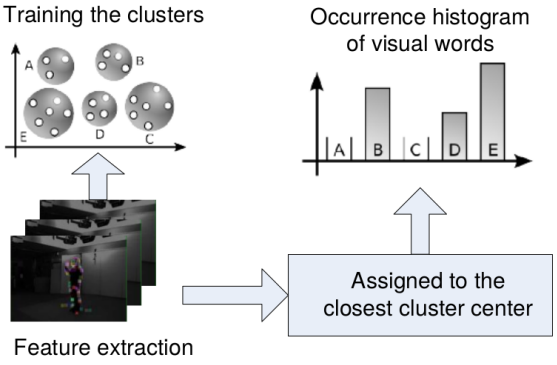

Figure 1: Pipeline for Human action recognition using traditional methods.

(iii) Our approach can be adapt to a large dataset and can be generalized to apply on multi-channel video frames.

The rest of the paper is structured as follows: Section 2 introduce the state-of-the-art methods for Human Action Recognition. Section 3 describes our method for action recognition. In Section 4, we demonstrate the efficiency and practicality of the proposed method on a public dataset which achieve state-of-the-art action classification performance. Finally, Section 5 concludes the paper.

2 RELATED WORKS

A batch of works on human action recognition mainly focus on developing robust and descriptive features (Scovanner et al., 2007; Xia and Aggarwal, 2013; Ni et al., 2013). Typical examples of local feature descriptors include histograms of oriented 3D spatio-temporal gradients (HOG3D) (Laptev et al., 2008), speeded up robust features (SURF) (Klaser et al., 2008), dense trajectory (IDT) (Wang et al., 2013) and motion boundary histogram (MBH) (Bay et al., 2006). In order to transform local descriptors into video feature descriptor most commonly used algorithm is bag of word (see Figure 1 (Fei-Fei and Perona, 2005)). The hand-crafted features based action recognition achieve good performance, but these features are not optimized for visual representation and lack discriminative capacity when encounter background clutter, large intra-class variations videos for action recognition and redundant informations.

More recently, impressive results have been obtained using convolution networks. Specifically, CNN 3D (Rahmani et al., 2018; Zhou et al., 2014) was used to extract spatio-temporal features from raw sequence data for action recognition. Through their hierarchical representation of entities, deep networks learn to

capture localized features as well as temporal information for 3D CNN as well as context index and can exploit high-level information from large-scale video datasets. Other work also focused on handcrafted features and combine with 3D CNNs (Hou et al., 2018) to expect for increasing classification accuracy. With these impressive results of deep learning, our work is naturally in this direction. Recent research showed that most of these approaches are not only computationally expensive, but they also fail on capturing context, high-level information, spatial, temporal, and interactions information. (Wang et al., 2014) shows that learning a model with a sequence of frames, can bring more temporal informations and can reduce over-fitting during training phase. In the work (Wang et al., 2016), the temporal segmentation of each video without overlapping reduces redundant information and results in solid feature for inference. (Simonyan and Zisserman, 2014) proposed a model with different inputs such as RGB, optical flow but this can be complicated when running the algorithm and takes a lot of time too. These methods, however, remain unable to take into account a learning pattern to model the temporal structure. Our proposed linear segmentation, while underlining this principle, constitutes the first framework of the temporal structure without external information (such optical flow or similar) that could be harmful during the training phase. The performance of recognition depends on the fact that the model pays attention to the region concerned, but discriminative localization is a problem for video. Also there is a lot of problems on the recognition of an inter-action object/human and background, the position of the camera, etc. So our deep structured model can be viewed as an extension and improvement of these existing architectures composed of many cliques that represent a part of our architecture and a part of sampling informations. Also we replace hand-crafted features to gray-scales images for inputs of each clique.

3 THE PROPOSED ARCHITECTURE

We will introduce the structure of our deep model and explain how it can handle large intra-class variance.

3.1 3D CNNs Spatio-temporal

Motivated by (Wang et al., 2014), the deep learning model that we present is like a spatio-temporal convolutional neural network, shown in Figure 2. We define a clique as a subpart of the network stacked up

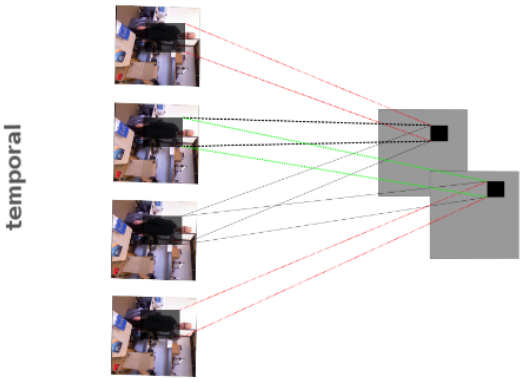

Figure 2: Extraction of multiple features from frames. Multiple 3D convolutions can be applied to contiguous frames to extract multiple features.

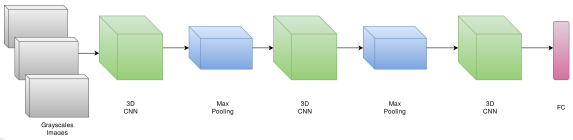

Figure 3: Illustration of a clique of our neural network. A clique is a part of our architecture and each clique has, as input, a short snippets from videos.

for several layers. Each clique extracts features from one decomposed video segment and an illustration is highlighted in Figure 3. Particularly, for each clique, three 3D convolutional layers are first built upon the raw input (i.e. gray-scale data), which is made up of at most one video image. Note that a max-pooling operator is applied on each 3D convolutional layer making our model robust to local different changes and noises. By the way, the convolution results generated by different cliques are concatenated into a long vector of features, on which we build three full connected layers to associate with the activity label. In the following, we introduce the detailed definitions for these components of our model.

1. 3D Convolutional Layer

   In 2D CNNs, convolutions are applied on the 2D feature maps to compute features from the spatial dimensions only. When applied to video analysis problems, it is better to capture the motion information encoded in multiple contiguous frame (Ji et al., 2013). We know that w,h represent the width and height of each frame and w',h',m' respectively represents the width, height and temporal length of the 3D CNN. We can obtain a feature map via performing 3D convolutions across the s-th to the $(s+m'-1)$-th frames. The response for the position (x,y) in the feature map can be repre-

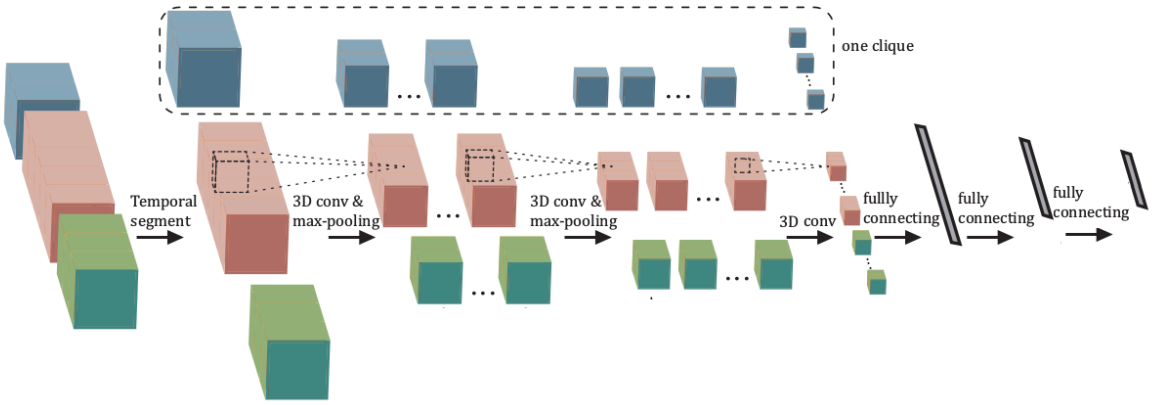

Figure 4: Example of our general deep architecture modified from (Wang et al., 2014)). Here we have three cliques (blue, red, green). These cliques extract information and are concatenated during the final classification stage.

sented as

$$u_{xys} = \tanh\left(b + \sum_{i=0}^{w'-1}\sum_{j=0}^{h'-1}\sum_{k=0}^{m'-1} \omega_{ijk} * P(x+i)(y+j)(s+k)\right) \quad (1)$$

where $p(x+i)(y+j)(s+k)$ is the input pixel value at position $(x+i, y+j)$ in the $(s+k)$-th frame, ω_{ijk} is the parameter for the convolutional kernel, and b is the bias for the feature map. It results in $(m-m'+1)$ feature maps after the first 3D CNN and for each set of feature maps, we successively further perform 3D convolutions and generate another set of feature maps on a deeper layer

2. Max-pooling Operator
   A max-pooling is applied after each 3D convolution result in order to obtain deformation and shift invariance (Yu et al., 2011). Given a set of feature maps, the max-pooling operator reduces its dimensionality and allows for assumptions to be made about features contained in the sub-regions.
3. Full Connection Layer
   The different sets of feature maps from the M model cliques, are concatenated into a long feature vector and then we apply three fully connected layers. Note that the number of the output neurons is K, which is the same as the number of categories of activities. Each neuron represents the probability of an activity hypothesis and to normalize the probabilities of output labels, we apply the softmax function on them.

3.2 Our Approach

We linearly segment each video (Wang et al., 2016) to avoid repeated information so as to improve classification during the learning phase. The number of

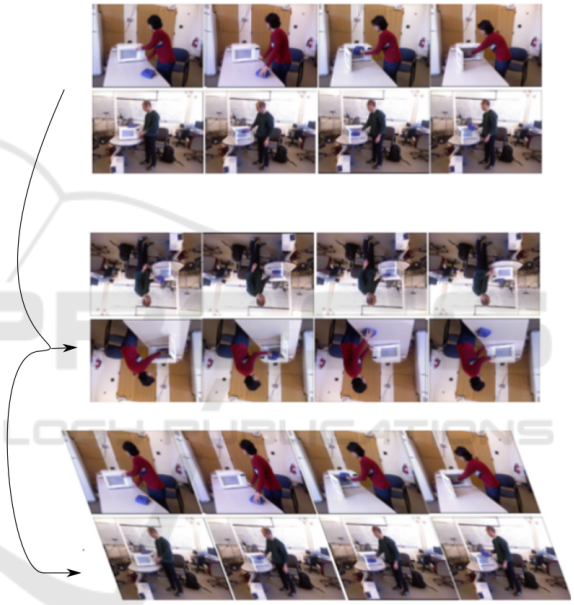

Figure 5: Illustration of the different transformations for data augmentation.

cliques is chosen according to the dataset which is important for the recognition of human action to maintain data consistently. The choice of number can also be used to increase the size of data, in place of using overlapping windows. We will demonstrate an example of cliques in section 4 on the CAD-120 dataset. The last fully connected layer is important because it is also chosen according to the extracted characteristics. More we want to have relevant information, more the number of cliques is needed, and more the fully connected layer is raised. Our work shows that with a small dataset it is empirically sufficient to choose a number of clique between 2 to 4. For a large dataset the number of cliques can be increase to 8. Once the different cliques have extracted the spatio-temporal information (Zolfaghari et al., 2017), we

need to concatenate these features then applied two fully connected layers and the last one corresponds to the layer related to the number of labels (actions). In the Figure 4 we show an example of our architecture.

3.3 Inference

The inference task is to recognize the category of the activity given a video X across all the labels y ($1 \leq y \leq K$, K the number of classes). To do this, we divide the video in N images sequences called clique. For each clique $C_i \in \{C_1, C_2, \ldots, C_N\}$ the classifier gives a label y_i and an acceptance probability $F_{y_i}(C_i, \omega)$ where ω are the learned parameters of our deep architecture. The final action label of the video is the label of the clique with the highest acceptance probability (Eq. 2).

$$\hat{y} = \arg\max_{y_i} F_{y_i}(C_i, \omega) \tag{2}$$

4 EXPERIMENTAL RESULTS

Following (Wang et al., 2014), we have conducted three types of experiments using the CAD-120 activity dataset (Koppula et al., 2013). The CAD-120 dataset contains 120 RGB-D activity sequences of ten categories. Some samples can be saw in Figure 6. This dataset is frequently used in 3D human activity recognition. These activities were done by four different subjects, and each activity was repeated three times by the same actor. The insufficiency of RGB-D data in human activities and the large variance in object appearance, human pose, and viewpoint are common challenges in human action recognition. Most papers on the human actions recognition using CAD-120 work with depth images (Sung et al., 2012a; Wang et al., 2014), but in our case we will just use the RGB images to face with this challenge.

4.1 Architecture Implementation Details

For the sake of reproducibility, we give here all the details of our deep architecture.

We will not use the depth images so we do not use the pose estimation for the recognition of human action. Increasing the data is a solution we used to fill the information gap that the dataset offers us and to remedy the non-use of the depth images. We do a data augmentation with well-known image processing techniques like rotation, translation etc. (Figure 5).

In the first experiment, the number of decomposed video segments (i.e. actions) is $M = 4$, and the length of the maximum number of input frames of each segment is $m = 9$. We scale the size of the input frame to $w = 80$ and $h = 60$ in our experiments and for each clique the number of the first 3D convolutional kernels is 7 with a kernel size of $9 \times 7 \times 3$, where each number represents the width, height and temporal length. In the second layer, the number of 3D kernels is 5, with the size $7 \times 7 \times 3$. We apply $3 \times 3 \times 3$ max-pooling operator over the 3D convolutions. Then in the last 3D convolutional layer, the number of kernels is 4 kernels with size of $1 \times 6 \times 4$. Hence we can obtain 700 features maps as the output for each network clique, and we merge the feature maps together into a vector of $700 \times 4 = 2800$ dimensions. Each unit in this vector is linked firstly to a fully-connected layer of 128 neurons, then to a fully-connected 64 layer. The last layer is connected to the output layer whose size is the number of the activity labels. We can see our full deep Architecture used for this work on CAD-120 in Figure 7. We have four entries in our network because the length of our video is quite short. If we continue on this logic we choose a temporal length a little above the limit necessary for a recognition of human action by deep learning. This choice $m = 9$ (Simonyan and Zisserman, 2014) is due to our dataset which contains different objects in the same label. We have a kernel $1 \times 6 \times 4$ in the last layer of 3D CNN and it behaves like a 2D CNN because we need at this level more spatial information than temporal ones.

4.2 Fine-tuning the CNN on the Target Dataset

The architecture is based on the AlexNet deep model (Wang et al., 2014; Krizhevsky A, 2012). Generally, a CNN configuration contains many parameters. If the model is trained from scratch on video dataset, it is easy to have an over-fitting phenomenon and it needs several weeks to train depending on the architecture (Yu et al., 2017). We introduce a pre-training scheme to optimize our model. It means that we train our model in a first step with the augmented data, and we save the weights. We call these weights the latent variables (Niebles et al., 2010). Then, we load the latent variables to initialize our deep architecture by changing the objective function and the hyper-parameters and train with the small set of data.

4.3 Inference

For inference, we train our network with the video of three people and the video of the last person is used as a test phase. It is important to note that the network never sees the test data and is able to predict

Figure 6: Example of samples extracted from (Wang et al., 2014). Several samples frames and depth maps are presented from the same class.

the class. This strategy is called leave-one-person-out cross-validation and it is commonly used for human action recognition (Gao et al., 2010).

4.4 Results

Regarding experimentation, in a first test we trained our network and in a second test we make a fine tuning and transfer learning on our network. We apply these methods because the weights and the bias are initialized randomly and we apply these techniques for a rather fast convergence. We studied the evolution of our network by learning firstly on three labels and then four labels. We can notice how our model scores well on three labels and even on four labels, we can see it in Table 1, Table 2, and Table 3. Even if mixing the order of the labels we always obtain excellent results. If we train the network on all the labels of our database we notice in the Table 4 that the overall precision decreases but remains always better than the best state-of-the-art methods. In the Table 4 we also see that we have a higher result by using the methods of fine tuning and transfer learning.

Table 1: Accuracy from label 0 to label 2.

Action	Our method	Our method + Fine-tuning
arranging-objects	88.3%	**95.8**%
cleaning-objects	86.6%	**96.3**%
having-meal	85.2%	**93.4**%
Overall Accuracy	86.7%	**95.1**%

Table 2: Accuracy from label 3 to label 5.

Action	Our method	Our method + Fine-tuning
making-cereal	91.8%	**97.3**%
microwaving-food	84.1%	**96.5**%
picking-objects	89.0%	**90.1**%
Overall Accuracy	88.3%	**94.6**%

Table 3: Accuracy from label 6 to label 9.

Action	Our method	Our method + Fine-tuning
stacking-objects	90.9%	**96.0**%
taking-food	92.0%	**92.3**%
taking-medicine	97.2%	**97.9**%
unstacking-objects	87.7%	**95.1**%
Overall Accuracy	92.0%	**93.8**%

Table 4: Accuracy of our deep architecture on CAD-120 with all activities.

Method	Accuracy
Our method	86.4%
Transfer learning	88.9%
Our method + Fine-tuning	**93.6**%

4.5 Comparison with State of Art

On this dataset, we adopt four state-of-the-art methods for comparison (Sung et al., 2012b; Koppula et al., 2013; Xia and Aggarwal, 2013; Ji et al., 2013). As show in Table 5 our approach obtains the average accuracy of 86.4%, distinctly superior than results generated by the competing methods. With fine tuning, our method is even higher. In Table 5, we report also the detailed accuracies for each class, compared with the method based on hand-crafted feature engineering (Xia and Aggarwal, 2013), and the deep architecture of convolutional neural networks (Wang et al., 2014): in almost all cases we obtain the best results even without fine tuning.

Table 6 summarize the average accuracy on CAD 120 dataset against the considered best four existing methods. It is important to highlight that these methods use also information taken from depth images. It appears that our model outperforms previous works although we do not use depth images. This is a great advantage because in practical applications, depth images are almost never available. Remark that for all methods we train the models using the same data annotation and we use the same evaluation protocol. The main contribution of our model is the fact it works directly on the raw data, i.e on the RGB ima-

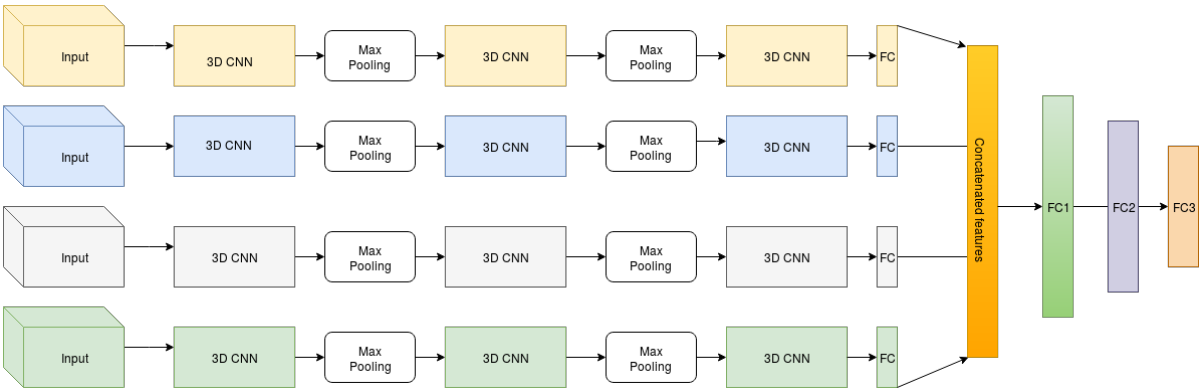

Figure 7: Our deep architecture used to train CAD-120. We have four cliques, 128 and 64-sized fully connected layers. Then the last fully connected layer is 10 for all activities.

Table 5: Comparative table between our proposed method and other methods using the same dataset and using deep learning. Best scores (and second best scores) are filled in dark gray (light gray). Our approach achieves state-of-the-art action classification performance.

Action	(Xia and Aggarwal, 2013)	(Wang et al., 2014)	Our method	Our method + Fine-tuning
arranging-objects	75.0%	82.3%	85.2%	92.1%
cleaning-objects	68.3%	79.7%	80.1%	95.6%
having-meal	41.7%	71.0%	75.9%	91.9%
making-cereal	76.7%	91.5%	90.8%	96.5%
microwaving-food	36.7%	85.3%	85.6%	93.4%
picking-objects	75.0%	97.2%	90.9%	92.7%
stacking-objects	75.0%	61.0%	82.4%	90.1%
taking-food	83.3%	93.5%	95.0%	96.3%
taking-medicine	58.3%	96.8%	97.5%	97.2%
unstacking-objects	33.3%	54.0%	81.2%	90.6%
Overall Accuracy	62.3%	81.2%	86.4%	93.6%

Table 6: The average accuracy on the CAD-120 database. Best score (and second best score) is filled in dark gray (light gray).

Method	Accuracy
(Scovanner et al., 2007)	59.8%
(Koppula et al., 2013)	80.2%
(Wu et al., 2013)	62.1%
(Ji et al., 2013)	64.3%
Ours	86.4%
Ours + Fine-tuning	93.6%

ges.

Our experiments are executed on a server with two multi-core processors, 32GB RAM and two Nvidia GTX 1080 Ti. For model learning, we set the learning rate at 0.001 value, for applying the stochastic gradient descent algorithm. The training times on CAD-120 dataset takes 5 hours for all videos including data augmentation. For inference, it only takes around 0.6 seconds to complete recognition on a given video.

5 CONCLUSIONS

We presented an extensible deep architecture ables to model long-term temporal structure and practice it on a complex dataset for human action recognition task. The proposed architecture is able to outperform existing methods, even with less data, in particular not using depth images and it is able to extract information of the sub-actions related to a main action.

Our model can still be improved and for that, we anticipate as future work to combine the descriptors extracted with our approach in a more complex architecture to extract the semantic of an action.

ACKNOWLEDGEMENTS

We gratefully acknowledge Region Centre-Val de Loire (France) for its support on this work by DANIEAL2 project funding.

REFERENCES

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE.

Gan, C., Wang, N., Yang, Y., Yeung, D.-Y., and Hauptmann, A. G. (2015). Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2568–2577.

Gao, Z., Chen, M.-Y., Hauptmann, A. G., and Cai, A. (2010). Comparing evaluation protocols on the kth dataset. In *International Workshop on Human Behavior Understanding*, pages 88–100. Springer.

Hou, Y., Li, Z., Wang, P., and Li, W. (2018). Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(3):807–811.

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3d convolutional neural networks for human action recognition. *IEEE Transactions on PAMI*, 35(1):221–231.

Klaser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association.

Koppula, H. S., Gupta, R., and Saxena, A. (2013). Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970.

Krizhevsky A, Sutskever I, H. G. (2012). Imagenet classification with deep convolutional neural networks. pages 1097–1105. In Advances in neural information processing systems.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563.

Laptev, I. (2005). On space-time interest points. *International journal of computer vision*, 64(2-3):107–123.

Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3361–3368. IEEE.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Ni, B., Pei, Y., Liang, Z., Lin, L., and Moulin, P. (2013). Integrating multi-stage depth-induced contextual information for human action recognition and localization. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE.

Niebles, J. C., Chen, C.-W., and Fei-Fei, L. (2010). Modeling temporal structure of decomposable motion segments for activity classification. In *European conference on computer vision*, pages 392–405. Springer.

Rahmani, H., Mian, A., and Shah, M. (2018). Learning a deep model for human action recognition from novel viewpoints. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):667–681.

Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 357–360. ACM.

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc.

Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sung, J., Ponce, C., Selman, B., and Saxena, A. (2012a). Unstructured human activity detection from rgbd images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 842–849. IEEE.

Sung, J., Ponce, C., Selman, B., and Saxena, A. (2012b). Unstructured human activity detection from rgbd images. In *2012 IEEE International Conference on Robotics and Automation*, pages 842–849.

Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79.

Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE international conference on computer vision*, pages 3551–3558.

Wang, K., Wang, X., Lin, L., Wang, M., and Zuo, W. (2014). 3d human activity recognition with reconfigurable convolutional neural networks. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 97–106, New York, NY, USA.

Wang, L., Qiao, Y., and Tang, X. (2015). Action recognition with trajectory-pooled deep-convolutional descriptors. In *IEEE Conference on computer vision and pattern recognition*, pages 4305–4314.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 20–36, Cham. Springer International Publishing.

Wu, P., Hoi, S. C., Xia, H., Zhao, P., Wang, D., and Miao, C. (2013). Online multimodal deep similarity learning with application to image retrieval. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 153–162.

Xia, L. and Aggarwal, J. (2013). Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu, K., Lin, Y., and Lafferty, J. (2011). Learning image representations from the pixel level via hierarchical sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1713–1720. IEEE.

Yu, S., Cheng, Y., Su, S., Cai, G., and Li, S. (2017). Stratified pooling based deep convolutional neural networks for human action recognition. *Multimedia Tools and Applications*, 76(11):13367–13382.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495.

Zolfaghari, M., Oliveira, G. L., Sedaghat, N., and Brox, T. (2017). Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2923–2932.