Human Action Recognition with 3D Convolutional Neural Network

Tiago Lima
Polytechnic School of Pernambuco,
University of Pernambuco,Brazil
E-mail: tabl@ecomp.poli.br

Bruno Fernandes
Polytechnic School of Pernambuco,
University of Pernambuco, Brazil
E-mail: bjtf@ecomp.poli.br

Pablo Barros
Knowledge Technology, Department of
Informatic, University of Hamburg
E-mail: barros@informatik.uni-hamburg.de

Abstract—**In the last decade, there was a development of technologies that allowed the possibility of storing and processing large amounts of data. Due to this, there was a considerable increase in the use of video cameras. Areas such as surveillance, traffic control, and entertainment, presented a greater demand for the development of techniques for analysis and automatic classification of videos. Within those areas of application, human activities recognition is considered one of the major problems and is discussed in the scientific environment due to related challenges, such as blurred images, point view changed confusion with background and low resolution. Recently, the Convolutional Neural Networks (CNN) have made considerable advances in several areas of research, improving state of the art in many cases, including images and videos classification problems. Thus, this work aims to develop a 3D CNN for the human actions recognition, as well as a study of the influence of the resolutions of entries in the network. After choosing the model are compared with other works in the area. The results obtained by the model surpassed the state-of-the-art in the bases evaluated and are discussed in this document.**

I. Introduction

The human action recognition in video has been one of the areas explored by the researchers of Computer Vision, whose objective is to recognize automatically the actions present in the videos [1]. This is a major problem of Computer Vision for several reasons, among them, the existence of smooth, different perspectives and cluttered background.

In the literature, there are several works that propose the recognition of human actions, the work of Aggarwal and Ryoo (2011) explains a survey of the main techniques. The first approaches used to integrate human actions were based on the human joints trajectories (Campbell and Bobick, 1995; Niyogi and Adelson, 1994). These methods need specific techniques to detect parts of the body or to track them in each image. Another approach that has already been explored is based on Bag-of-Words (BoW) [2], this type of proposal requires a large storage space for the less frequent features, besides the need to combine with other techniques for classification and extraction of Features.

The use of descriptors is frequent in the literature. Other works use descriptors to extract information from the most important parts of the frames, including Histogram Oriented Gradient (HOG) [3], Histogram Oriented Flow (HOF) [3], Bounds Motion Histogram (MBH) [4]. On the other hand, when the images are large or very small, the results of the extractions may contain a large amount of noise or low representativeness of the standards. This is because the descriptors are not optimized for the visual representation and they do not have good visual representativity which can lead to a low recognition of actions.

978-1-5090-5105-2/16/$31.00 ©2017 IEEE

With the success of Deep Networks, in special the AlexNet in 2012 [5], the deep models are being explored recently by different researchers. Among the deep architecture models, Convolutional Neural Networks (CNNs) gained attention because of their ability to learn contextual relationships between features [5]. This type of architecture has already achieved great results in domains such as digit recognition [6], speech recognition [7], emotion recognition [8].

Some authors have tried to apply the CNN for action recognition in videos. In the work of Ji et al. [9], they proposed the use of Convolutional Neural Network with two flows for the recognition of the actions, the first is the raw frame and the second is the optical flow with the temporal information of the movement between the frames.

In the work of Wang et al. [10], they have developed a two-channel Convolutional network based on RGB frames. The first input of the model is the raw image and the second is the optical flow extracted from the motion in order to predict the trajectories. In our proposal, we used a sequence of images to describe the context of the action. In the work of Lin et al. [11], they use a CNN to decompose temporal information by separating groups by sub-actions into RGB-D videos. Unlike the previous work, in this paper we recognize the actions that summarize the type of video. Furthermore, another difference is in the input format of the models since in this research we used the RGB format.

In the work of Chron et al. [12], they proposed a descriptor based on body posture to determine the action in videos. Initially, they calculated the optical flow between frames and motion in each x and y direction. With the information of the directions, they calculate the movement of each part of the body. Finally, they use the raw image and the flow of motion as input to a CNN for action recognition.

Other authors have explored the use of 3D Convolutional Neural Network (3DCNN) for the recognition of human actions in videos. This kind of architecture can process temporal information, what has meaning for applications in videos [13].

In the work of Ji et al. [13], they used gray scale images, gradient and the optical flow along the x and y axes, taking these values as input in a 3DCNN for the recognition of actions in secure videos.

In our work we consider only the images and the quantity of frames as input data for the model. In this way, this paper proposes the use of a 3DCNN to learn temporal spatial information of human actions. We performed some experiments on the influence of input on the model and the number of frames needed to represent an action. To summarize, our contributions in this paper are:

- We evaluate the influence of the number of frames and the resolution of the input images necessary to represent a video of the network.
- We evaluated the processing time when we varied the number of input frames and the resolution.
- We applied our neural network model on the UCF50 database and compared it with other works that followed the same protocol employed in this database.

This paper is organized as follows. In Section 2 we detail our architecture and network used. In Section 3 we explain the experiments performed, the materials and methods. In Section 4 we make a discourse about the results obtained. Finally, we conclude with the final considerations about our architecture and future work.

II. Architecture Design

In this session, we initially discuss our proposed architecture, then we show the Convolutional model applied for video classification. Finally, we show how the neural network was implemented.

A. Architecture

We explained the modeling process for recognition in videos using 3D Convolutional Neural Network. Figure 1 is an overview of the model. Currently, there are several proposals of Neural Networks for video classification: Ji et. al [13], Wang et. al [10], [12]. However, among these collected works, they all used some additional artifice to capture the temporal information. Next, we begin the discussion of our model.

We split the model into two parts: first0 we use the first part to select the model settings. We load a database and apply a pre-processing to fit the data for the model. Then we analyze the influence of the input on the model and its stability. Secondly, we load the template into the classification process. We used the 3DCNN algorithm to sort the videos, which is discussed below.

B. 2D Convolutional Neural Networks

The Convolutional Neural Network (CNN) is biologic-inspired by Multilayer Perceptron [6] and in studies of the Hubel and Wiesel (1968) [14] with the visual cortex of cats. Hubel and Wiesel found that visual cortex contains complex arrangement of light-sensitive cells called *receptive fields*. These cells when act a local filters on input extract patterns that have strong local spatial correlation in the images.

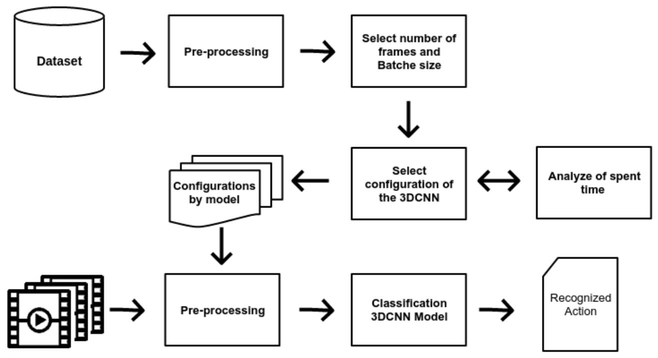

Dataset
Pre-processing
Select number of frames and Batche size
Select configuration of the 3DCNN
Analyze of spent time
Configurations by model
Pre-processing
Classification 3DCNN Model
Recognized Action

Fig. 1. Process for choosing the template settings.

Based on this, the CNNs explore the space-local correlation between the local patterns with the neurons of the adjacent layers. Each layer unresponsive to variations outside of its receptive field with respect to the retina. In this way, the architecture ensures that the filters learned produce the strongest response to a spatially local input pattern. Each filter of the previous layer is represented as visual field for the next layer, this replication of the values share the same parametrization for the feature maps.

A CNN is composed of one or more Convolutional layers (often a Subsampling layer) and then by one or more full connected layer, in the standard version of a Multilayer Neural Network as proposed by LeCun te al. [6]. By form the values of each unit at position (x, y) of *j*th on the map of features of the layer *i*th, denoted by v_{ij}^{xy}, is given by:

$$v_{ij}^{xy} = tanh\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}\right) \quad (1)$$

where tanh(.) is the hyperbolic tangent function, *b*ij is the bias for the features map with m indexes on the feature map set of layer (i-1)th connected to the current layer, w_{ij}^{pq} and h the value of (p, q) of the connected kernel of the features map, and P_i and Q_i are the height and width of the kernel applied, respectively.

In the sub-sampling layer the feature map is reduced by a pooling over the local neighborhood of the feature map to the next layer, the pooling operation can be performed with area overlay or not. CNNs can be constructed by multiple layers of convolution and sub-sampling alternately.

C. 3D Convolutional Neural Networks

3D CNN calculates features for both temporal and spatial dimensions [13]. 3D convolution is achieved through convolutions of a 3D kernel on the cube formed by several frames continuously together. For this construction, the features map in the convolution layer is connected with multiple frames of the next layer, thus capturing the motion information. At the Figure 2, clearly shows the differences between the 2D and 3D models.

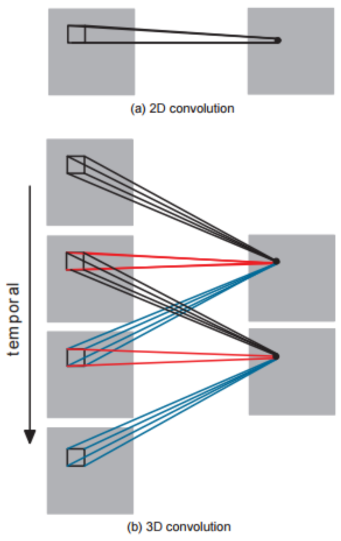


Fig. 2. We compare the 2D (a) and 3D (b) convolutions models. As can be noted in (a) have only spatial information represented by only one block, as in (b) the time kernel is the third dimension. The set of connections in red are share weights. In 3D CNN some kernel is applied with the overlay creating a 3D cube.

Formally the values of the positions (x, y, z) in *j*th of the map of features for each layer *i*th given by:

$$v_{ij}^{xyz} = tanh\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \tag{2}$$

Which is the size of the 3D kernel along the temporal dimension, w_{ijm}^{pqz} is the (p, q, r)th value of the kernel connected in *m*th of the features map of the previous layer.

D. Network implementation

The architecture we used for our experiments contains two hidden layers and four Convolutional, more details are illustrate in Figure 3. We apply the kernel $3x3x3$ to all the convolution layers. For the Pooling layers we use a $2x2x2$ window with no overlap. Between each block we used a Dropout regularization layer with probability of 0.5. Then a full connected layer with 512 output neurons followed by hidden layers Dropout of 0.25. Finally, another Full connected layer with 50 output neurons. The network has approximately $2.3M$ of training parameters. All layers of the template initialized with a normal distribution described in the work of the Xavier [15].

We initially tested the Stochastic Gradient Descend [16] in our model, however, the model always started in a local pinhole, so we chose to use the Adam optimizer proposed by Kingma and Bia [17]. Adam is a first order optimization algorithm based on objective stochastic functions and low order moment adaptive estimators. We used the default settings: learning rate of 1e-3, β_1 = 0.9, β_2 = 0.999, ϵ = $1e-08$ and γ = 0.0.

III. Experimental Evaluation

In this section, we first describe the hardware and software settings used in the runtime environment. Secondly, we talk about the database used in the paper. We explain the process used to select the model settings and finally, we describe the approaches followed for the recognition of human actions in videos.

All experiments were performed on a desktop computer with the Windows 10 Operational System, processor core i7 4790k with 4GHz frequency, Nvidia 970 GTX video card with 4GB of video and 32GB of RAM. All codes used were in the Python language version 3.5 in the Keras framework for Deep Learning [18].

A. Dataset

The UFC50 is a database for recognizing human actions, it consists of realistic videos taken from YouTube, in total the base is composed of 6618 videos [19]. The actions are divided in 25 groups containing an average of 100 videos of per class by group. The videos have challenging characteristics like moving cameras, different viewpoint, large interclass variations, cluttered background, occlusion, low illumination and low quality. In Figure 4 we show examples of videos present in the UCF50.

B. Material and Methods

We show the model divided into two steps, based on Figure 1. In the first part we perform the pre-processing of the UCF50 database to choose the model settings. The images are extracted from videos, and then they are converted to gray scale. The resulting images are resized to the resolutions (16, 32, 48, 64), each interval value is applied to the row and column. We also observed the size of the sequence required to represent a video and defined the intervals (5, 10, 15, 20, 25) of frames. After defined these values, we evaluated how the model behaves using Holdout with 60% (3970 examples) for training, 20% (1324 examples) for validation and 20% for test (1324 examples). The model was trained for 10 epochs, and we calculated the metrics: Accuracy, Precision, and Recall. Finally, we analyzed which configuration obtained the highest rates and the shortest processing time.

The second step, we applied the same pre-processing method of the previous step and load the model with the defined configurations, then we train the model following the proposed protocol Leave-One-Group-Out (LOGO) proposed by Reddy and Shah (2013) [19]. The network is executed 30

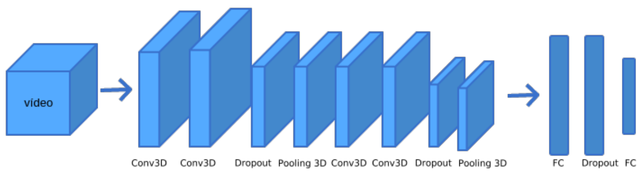


Fig. 3. our model approach.

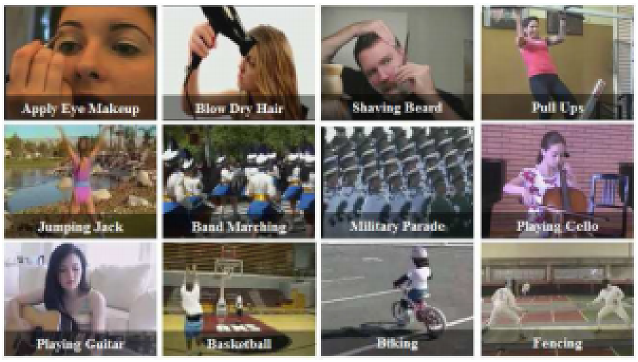


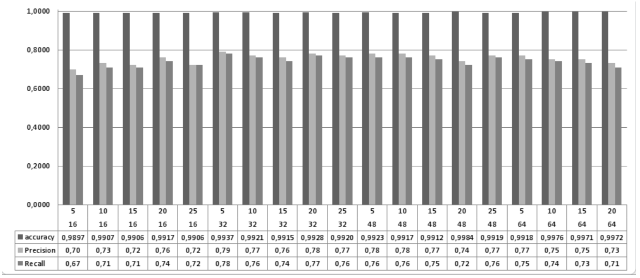
Fig. 4. Screenshots from videos in the UCF50 dataset showing the diverse action categories.

times and we analyze the results compared with other works. Then we applied the Holdout again with the same split of the base of the first stage. The metrics observed were Accuracy, Precision, and Recall for both performance.

IV. Results

A. Model configuration selection

We evaluated the different resolutions of the input images and the influence on the model, as shown in Figure 5. We found that the Accuracy metric yielded better results when the inputs were between $48x48$ and $64x64$, all values were above 98.9% in this range. More specifically, the $64x64$ interval averaged 99.72%. We observed that increasing the input resolution increases the accuracy as well. Due to lack of memory, we were unable to perform the last configuration with 25 frames and $64x64$ input. The 64x64 interval obtained the best results on average with 99.72%. However, the $16x16$ inputs had the lowest values with an average of 99.0%. We note that by increasing the input resolution, the model shows an improvement in accuracy. Due to lack of memory, we were unable to execute the last configuration with 25 frames and $64x64$ input. We noticed another fact: when we have the sequences of images 5 and 10 the model obtained better results of the Accuracy, with execution for the $16x16$ interval. Nonetheless, the Accuracy and Recall metrics average at 74.0% and 74.0% respectively.



	5 16	10 16	15 16	20 16	25 16	5 32	10 32	15 32	20 32	25 32	5 48	10 48	15 48	20 48	25 48	5 64	10 64	15 64	20 64
■ accuracy	0,9897	0,9907	0,9906	0,9917	0,9906	0,9937	0,9921	0,9915	0,9928	0,9920	0,9923	0,9917	0,9912	0,9984	0,9919	0,9918	0,9976	0,9971	0,9972
■ Precision	0,70	0,73	0,72	0,76	0,72	0,79	0,77	0,76	0,78	0,77	0,78	0,78	0,77	0,74	0,77	0,77	0,75	0,75	0,73
■ Recall	0,67	0,71	0,71	0,74	0,72	0,78	0,76	0,74	0,77	0,76	0,76	0,76	0,75	0,72	0,76	0,75	0,74	0,73	0,71

Fig. 5. Influence of image resolution and number of frames in the input.

We observe the time taken to load the database, train the model, evaluate and test. We show in Figure 6 the time spent in each model execution. Let's look at a direct relationship between the input resolution and the number of images used. For example, when applying the $16x16$ resolution there is almost no variation in processing time spent, however, when we increase the variables as mentioned earlier time grows rapidly.

When we compare the results of the groups in Figures 6 and 5. We observe the model with 20 frames and $64x64$ it takes 1200 seconds to performance. On the other hand, with 5 frames and $16x16$ input, we only have 83 seconds. Moderately the same Accuracy, Recall, and Precision. When considering the results of Figure 6 the $32x32$ and 5 frames configuration has the higher metric performance and the fourth best time between all runs, so we chose this configuration for the next step of this work.

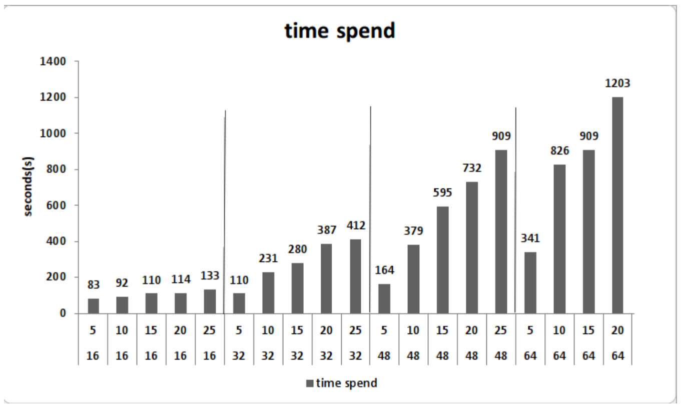


Fig. 6. Time spent for each run.

B. The Network Performance

After the selection of the configuration, we compare with other works that use the protocol proposed by Reddy et. al[19]. In Table I we show this comparison. In the work of Reddy (2013) [19], they combined Optical Flow, 3D-SIFT and PCA to extract as video features, then they apply an SVM. These proposals obtained Precision 76.9%.

In the proposal of Liu et. al (2015) [20], they used Bag of Visual Words to extract features for stock recognition; after that, they applied the concept of multi-view on the features by how to obtain the similarity of data. Finally, they applied the results to linear SVM. They got 87.9% Accuracy with this proposal.

In Wang and Schmid's work (2014) [10], they used Optical Flow and a Dense Trajectory to extract as characteristics of the frames, after which they applied an SVM. They obtained 91.2% Accuracy.

In the proposed Peng et. al (2016) [21], they used the descriptors HOG, HOF, MBH to extract features, after that, they combine as output of the descriptors in the BoW, after this, they use an SVM for recognize the actions. With this approach, they obtained the state-of-the-art with 92.3% in the UFC50. Our proposal obtained 97.61% Accuracy, with a standard deviation of 0.16%.

TABLE I
COMPARISON OUR WITH THE OTHERS METHODS

Method	Accuracy (%)
Reddy and Shah (2013)[19].	76.90
Liu et. al (2015)[20]	87.19
Wang and Schmid (2013)[22]	91.2
Peng et. al (2016)[21]	92.3
Our approach	**97.61**

C. Discussion

Since the other papers analyzed do not carry out a more in-depth analysis, we decided to apply the Holdout mentioned. This approach resulted in 99.21% Accuracy and standard deviation 0.09%. We do some considerations about the execution model for 30 epochs to understand the behavior of the network better. We have a Confusion Matrix in Figure 7. The labels on the vertical axis indicate the true class labels and the horizontal axis indicates the ordered classes. The main diagonal of the matrix represents the cases of predicted correctness examples, and the values outside the diagonal represent the values that the model predicts in the wrong way. We also look in more detail at the confusion matrix for each Confusion Matrix class by traversing the data in the table II.

The Confusion Matrix for the proposed model shown in Figure 7. The labels on the vertical axis indicate the true class labels and the horizontal axis indicates the ordered classes. The main diagonal of the matrix represents the cases of predicted correctness examples and the values outside the diagonal represent the values that the model predicts in the wrong way. As it can be observed that the use of 3DCNN can give good results, a more detailed comparison between the other works was not possible since the authors did not perform the analysis with the Confusion of Matrix. Thus we show a more comprehensive analysis for each class of the confusion matrix.

In the table II we have the Accuracy, Recall F1-score and Supported values. In general, the network achieved good rates for classes with average values of 80% Precision, 80% Recall and F1-score. Upon closer inspection, it is noticed that the Biking, Swing, SalsaSpin, PoleVault, WalkingWithDog, JumpingJack, GolfSwing and Lunges classes were less than 60% of accuracy. The Swing and PoleVault classes had the worst performance on all metrics.

Some of the Swing class videos have background resembling metal frames present in the CleanAndJerk class which can create confusion in the model. Another fact observed was that some examples of Swing class were classified as RockClimbingIndoor, it is observed that the postures of the bodies have some similarity to the movement of the balance. Finally, there was confusion with the PlayGuitar class, when watching the videos, it is noticed that the positioning of people may have influenced since the videos were centralized in the people.

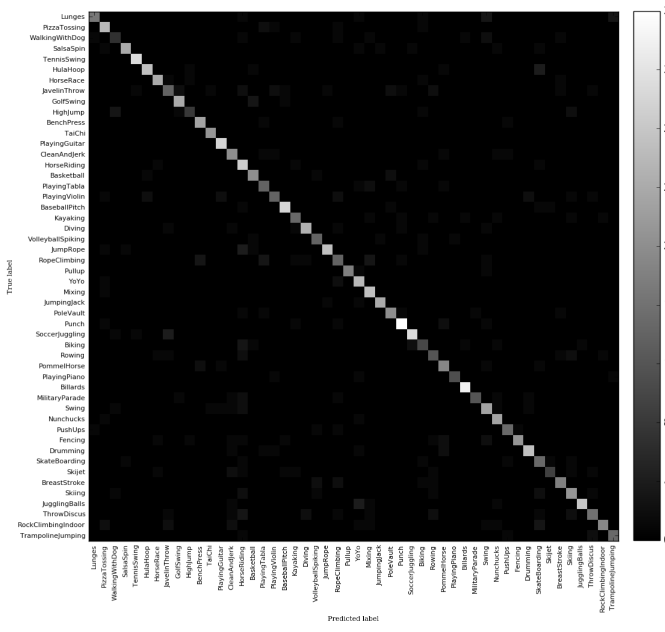


Fig. 7. Matrix confusion.

V. Conclusion

In this article, we analyze the performance by using 3DCNN models for the recognition of human actions. The results obtained show the effectiveness of this type of solution in the classification of videos.

We first explore two aspects of preprocessing for the video classification process: the variation of the input resolution and the amount of frames needed to represent a video. We use this process to select the best settings for the model. The results obtained showed a high Accuracy with a mean of 98.7%, and Recall with a mean of 77.3%. Secondly, we use the configuration obtained and apply the LOGO protocol to compare with other works. Our proposal had the best results, 97.3% Accuracy, while with the state-of-art proposed by Peng et. al [21] with 92.3%. After that, we applied the holdout to evaluate the model in a more detailed way, obtaining 99.21% accuracy. Finally, we analyze the confusion matrix of the model. We found that 3DCNN models have high capacity to process space temporal information.

As future work, we intend to evaluate our model using other databases. We also plan to conduct a study on network salience in video classification. Another point we want to explore, is to classify and tracking objects within the video to aid control areas such as airports, restricted spaces and private areas.

Acknowledgment

This research was partially supported by CNPq, Capes and Facepe.

References

[1] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16:1–16:43, Apr. 2011.

TABLE II
ANALYSIS OF THE CONFUSION MATRIX

Classe	precision	recall	f1-score	support
PommelHorse	0.84	0.81	0.83	32
TaiChi	0.94	0.81	0.87	21
WalkingWithDog	0.61	0.61	0.61	18
PoleVault	0.59	0.59	0.59	27
MilitaryParade	0.88	0.75	0.81	28
Billards	0.88	0.92	0.90	25
Diving	0.79	0.74	0.76	35
Skiing	0.80	0.89	0.84	18
Kayaking	0.78	0.53	0.63	34
SoccerJuggling	0.88	0.78	0.83	37
JugglingBalls	0.83	0.94	0.88	31
Skijet	0.80	0.89	0.84	18
BenchPress	0.92	1.00	0.96	33
SalsaSpin	0.58	0.90	0.71	20
PushUps	0.75	0.75	0.75	20
PlayingTabla	0.72	0.87	0.79	15
JumpRope	1.00	0.87	0.93	30
CleanAndJerk	0.88	0.79	0.83	19
PizzaTossing	0.79	0.79	0.79	19
Drumming	0.94	0.94	0.94	36
PlayingViolin	0.74	0.85	0.79	20
JumpingJack	0.67	0.82	0.74	17
Punch	0.74	0.88	0.80	42
Nunchucks	0.74	0.92	0.82	25
BreastStroke	0.83	1.00	0.90	19
HorseRace	0.95	0.86	0.90	22
YoYo	0.78	0.78	0.78	23
VolleyballSpiking	0.88	0.90	0.89	31
Fencing	0.91	0.97	0.94	31
Basketball	0.97	0.85	0.90	33
HulaHoop	0.80	0.77	0.78	26
BaseballPitch	0.92	0.85	0.88	27
TennisSwing	0.90	0.93	0.92	30
RockClimbingIndoor	0.73	0.59	0.65	27
PlayingPiano	1.00	0.71	0.83	21
RopeClimbing	0.89	0.64	0.74	25
GolfSwing	0.66	0.61	0.63	31
SkateBoarding	0.72	0.81	0.76	26
Rowing	0.76	0.67	0.71	24
Mixing	0.66	1.00	0.79	19
HorseRiding	0.78	0.69	0.73	52
JavelinThrow	0.86	0.67	0.75	27
PlayingGuitar	0.89	0.97	0.93	35
HighJump	0.88	0.81	0.85	27
Swing	0.44	0.44	0.44	27
ThrowDiscus	0.84	0.72	0.78	29
TrampolineJumping	0.71	0.65	0.68	37
Lunges	0.64	0.76	0.70	21
Biking	0.54	0.82	0.65	17
Pullup	0.88	0.88	0.88	17
avg / total	0.81	0.80	0.80	1324

[2] P. Foggia, G. Percannella, A. Saggese, and M. Vento, "Recognizing human actions by a bag of visual words," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2910–2915.

[3] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[4] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *European conference on computer vision*. Springer, 2006, pp. 428–441.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[6] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[7] Y. Qian and P. C. Woodland, "Very deep convolutional neural networks for robust speech recognition," *CoRR*, vol. abs/1610.00277, 2016. [Online]. Available: http://arxiv.org/abs/1610.00277

[8] P. Barros, C. Weber, and S. Wermter, "Emotional expression recognition with a cross-channel convolutional neural network for human-robot interaction," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 582–587.

[9] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[10] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4305–4314.

[11] L. Lin, K. Wang, W. Zuo, M. Wang, J. Luo, and L. Zhang, "A deep structured model with radius–margin bound for 3d human activity recognition," *International Journal of Computer Vision*, vol. 118, no. 2, pp. 256–273, 2016.

[12] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3218–3226.

[13] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[14] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[15] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[19] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, vol. 24, no. 5, pp. 971–981, 2013.

[20] J. Liu, Y. Huang, X. Peng, and L. Wang, "Multi-view descriptor mining via codeword net for action recognition," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 793–797.

[21] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, pp. 109–125, 2016.

[22] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.