Improving Human Action Recognition through Hierarchical Neural Network Classifiers

Pavel Zhdanov, Adil Khan
Institute of Robotics,
Innopolis University, Russia
Email: {p.zhdanov,a.khan}@innopolis.ru

Adín Ramírez Rivera
Institute of Computing,
University of Campinas, Brazil
Email: adin@ic.unicamp.br

Asad Masood Khattak
College of Technological Innovation,
Zayed University, UAE
Email: asad.khattak@zu.ac.ae

***Abstract*—Automatic understanding of videos is one of the complex problems in machine learning and computer vision. An important area in the field of video analysis is human action recognition (HAR). Though a large number of HAR systems have already been developed, there is plenty of daily life actions that are difficult to recognize, due to several reasons, such as recording on different devices, poor video quality and similarities among actions. Development in the field of deep learning, especially in convolutional neural networks (CNN), has provided us with methods that are well-suited for the tasks of image and video recognition. This work implements a CNN-based hierarchical recognition approach to recognize 20 most difficult-to-recognize actions from the Kinetics dataset. Experimental results have shown that the application of our method significantly improves the quality of recognition for these actions.**

***Index Terms*—action recognition, neural networks.**

I. Introduction

Today, human action recognition (HAR) in video data is one of the important problems of computer vision [1]. The goal of HAR is to define and classify human actions in videos. Development of this technology has a number of useful applications. For example, airports, metro stations, and other public places require constant monitoring. In such places, a large number of video cameras are installed, covering almost every section of the terrain to constantly monitor the occurrence of abnormal situations. To achieve this, it is necessary to distinguish suspicious actions from ordinary ones. Other applications may include the use of HAR methods to monitor and assess the situation in real time in medical institutions, construction sites, child care, etc. [2].

A large number of HAR systems have been developed in the past to recognize different types of actions in videos. These include both simple actions (such as walking, running, jumping) [3], as well as complex actions that may involve interaction among a large number of people and objects [4]. These systems are mainly built using machine learning methods, such as artificial neural networks [5]. To train such systems, so that they can recognize actions in video clips of different quality and content, we need big video datasets. Popular human action datasets containing a large number of different videos are UCF101 [6], HMDB [7], ActivityNet [8], and Kinetics [1].

In the general case, training a HAR system consists of getting a set of frames from a specific clip, combining these frames in a sequence, extracting features from them, and submitting them as input to a classifier [9]. During training, we know exactly what action was submitted to the classifier, so we can make adjustments every time the classifier is wrong, thereby increasing the likelihood of getting a correct answer in the future.

Despite the large body of existing HAR methods, the number of actions that can be recognized with good accuracy by these methods is limited. Attempts to increase the number of recognizable actions by modern systems failed due to various constraints. These include poor-quality of video clips, cluttered background, noise, and the problem of similarity of various actions [10]. For instance, it is often quite difficult to draw a line between playing basketball and shooting basketball.

The Kinetics dataset consists of 400 actions. Earlier attempts [1] to use this data to build HAR methods were unable to achieve high recognition rate for all of these actions. The 20 most difficult-to-recognize actions from this dataset are shown in Table I. One possible reason for the low recognition rates is the use of a single model to recognize all actions [1]. Such complex learning problems can be solved better using hierarchical systems. Dividing the model into modules creates a more flexible one by using existing machine learning methods to improved HAR, even for the difficult actions.

Improving the recognition of the 20 worst actions, reported by Kay et al. [1], is important because it refers to the problem of scaling the recognizable actions as a whole. Particularly, it is challenging due to the similarity of these actions. Our interest is the development of a flexible model that allows us to improve the quality of recognition of actions easily.

Therefore, the main idea of our work is to develop a hierarchical classification model capable of improving the quality of recognition of the 20 most difficult-to-recognize actions from the Kinetics dataset. The hierarchical scheme makes it possible to train separate models for each group of actions, which results in each network learning a feature space well-suited for the corresponding group of actions, instead of learning and using the same set of features for all actions.

Existing methods for classifying actions in HAR use different types of artificial neural networks. The disadvantage of these approaches is the use of a single model for recognition. A single model that handle different and independent types of activities related to different groups (such as cooking, drink-

Table I
TWENTY MOST CHALLENGING ACTIONS FROM THE KINETICS DATASET [1] AND THEIR CORRESPONDING GROUPS USED IN THIS WORK.

Group	Actions
Sport	Throwing ball Shooting basketball
Communication	Answering questions Recording music
Fails	Headbutting Faceplanting
Makeupping	Fixing hair
Hands action	Tossing coin Shaking hands Rock scissors paper Slapping
Cooking	Making cake
Eating	Eating chips Eating doughnuts
Drinking	Drinking Drinking beer Drinking shots
Other	Sniffing Yawling Sneezing

ing, etc.) has problems encoding this information. Often the extracted features for an action can be similar to features from other actions. Instead, the proposed hierarchical classification method simplifies the complexity of one model by dividing the responsibility of encoding and recognition into a set of neural networks. In this way each network is responsible only for one group of actions.

Our method contains two steps. The first stage performs the classification of the input video into a group of activities. The second level classification network, selected by the output of the previous step, re-processes the video and determines the specific action. As for the type of the neural network, we used 3D convolutional (conv3D) neural networks, since they can handle well the spatio-temporal data, such as videos. The classifier on the top level predicts the action groups, that are used to condition the next layer of classifiers to produce the final action. An overview of the proposed method is shown in Fig. 1. The hierarchical classification model was tested on Kinetics [1] dataset, and the results we obtained show an advantage over existing solutions.

II. RELATED WORK

The general pipeline of action recognitions is as follows. The first step is to form a dataset containing as many different videos as possible on different types of actions. The data must be structured to allow for efficient computation [11]. After the dataset is formed or selected, it is necessary to determine whether any preprocessing of the video will be required. For some tasks, video preprocessing may not be needed, but it is often better to clean the video of noise and unnecessary data [12]. Features are extracted [13]–[15], and,

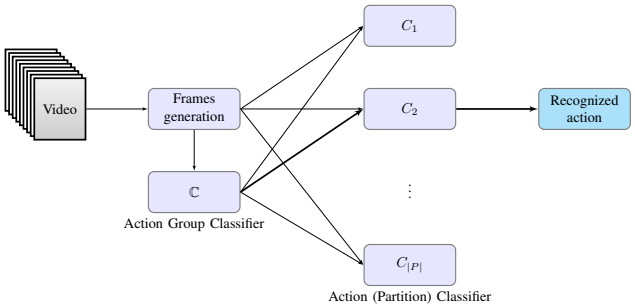

Figure 1. General framework of the proposed method. The video data is processed to extract the high level action group through a high level classifier, $\mathbb{C}$. Then, another classifier, C_g, conditioned on the predicted group, g, is used to predict the final action. (The bold path shows a possible outcome when $\mathbb{C}(x) = g = 2$ for the video input x.)

in some cases, there is a feature selection step to achieve dimensionality reduction [16]. The final stage of the pipeline is classification [17]. A large number of action datasets have emerged over the last few years. Among them Weizmann [18], ASLAN [19], KTH [11], [20], UCF Sports [21], UCF-Youthes [22], Hollywood [23], [24], HMDB51 dataset [25], and Kinetics [1] are the most common. The latter contains 400 different types of actions, which is almost twice as many as its predecessors. This dataset is the focus of our work due to its challenging classes.

Various algorithms can be applied to the input video clip for feature extraction. For example, Dense Sampling Methods [26], Sparse Sampling Methods [27]–[30], histogram of oriented gradients (HOG) [31], [32], HOG3D [33], and optical flow [14], [15]. Some works have also used Convolutional Neural Network (CNN) based descriptors [13], [34]–[36].

As for feature selection, various approaches have been used that aim at reducing the dimensionality [37], [38]. The most commonly used feature selection methods include Principal Component Analysis, Stepwise Linear Discriminant Analysis [39] and Genetic Algorithms [40], etc. Overall, feature selection methods are divided into the following categories [16]:

- Filter methods, which select features based on a performance measure regardless of the employed data modeling algorithm.
- Wrapper methods, which consider feature subsets by the quality of the performance on a modeling algorithm, which is taken as a black box evaluator.
- Embedded and hybrid methods, which perform feature selection during the execution of the algorithm.

There exist several methods that have used Kinetics dataset for HAR. Carreira and Zisserman [41] proposed the I3D method based on a two stream network (optical-flow and raw-data based). The work did not refer to the assessment of the difficult-to-identify actions of the dataset [1]. Rather, they tested the I3D method on the entire Kinetics dataset and reported average recognition rates.

Sparsely Labeled for Action Classification (SLAC) [42] method was developed using Kinetics dataset to automatically

place labels in a video indicating a range containing a human action. The SLAC approach reduces the number of human actions by automatically identifying clips that contain coordinated actions. This method claims to improve the accuracy of HAR based on Kinetics dataset, but the authors did not show the accuracy of recognition for the 20 actions from Table I.

Another Kinetics-based work is Temporal 3D ConvNet (T3D) [43], which proposes to create another time layer in CNN. The results of the experiments show results exceeding the existing classification methods by several percent. However, there are no experiments indicating an improvement in the accuracy of recognition of the similar actions.

One of the closest works [44] to ours, in which the authors showed improvements in the quality of recognition of 9 actions from Table I, implements and tests an Appearance-and-Relation Network (ARTNet). The method was tested on Kinetics [1], UCF101 [4], and HMDB51 [13] datasets. For UCF101 and HMDB51, their results were much better than previous methods. However, in the case of Kinetics dataset the method showed 20% worse results than C3D [45], and I3D [41].

Considerable efforts were concentrated around the application of recurrent neural networks (RNN) to improve the accuracy of the HAR on video. However, Long et al. [46] implement the Attention Clusters (AC) approach based on the studies of local feature integration which showed results superior to the use of RNN. The authors showed the competitiveness of the AC algorithm with existing action classifiers, but did not show the accuracies on the similar actions too.

To conclude, nowadays, Kinetics is one of the most popular and widely used action dataset containing a large number of different types of actions. None of other works that have used this dataset [47]–[59] have focused on evaluating the impact of similar actions on the overall classification accuracy. More importantly, there is no effort on improving the recognition accuracy of these hard-to-classify actions, as reported in the original dataset [1]. The method of hierarchical classification for HAR, developed in this work, would make possible to use Kinetics dataset more favorably by reducing the time for retraining the networks and increasing the accuracy of recognition for the similar actions.

III. Hierarchical Classification

Based on experience from previous works, we propose a model that improves the limitations of previous solutions. For different types of actions, different features may be better suited. Therefore, unlike existing works, which used a single type of features for all considered types of actions, we propose a hierarchical model; a two-level structure for recognizing actions. The overview of the method is shown in Fig 1. First the video is fed to the model trained to recognize a global action group or class. Once the group has been recognized, the video is transferred to the next node that launches the model to recognize the actions inside the global class. Since the videos are of different length, we extract frames to normalize them, see Section IV-A for the details.

Due to the success of other methods on the standard databases, we decided to improve the classification of the 20 most difficult actions from the Kinetics datasets [1] (as shown in Table I). The main difficulty of these classes arises from their similarity, that can be challenging even for humans.

To tackle the similarity problem, we propose to use a two layer classifier (although more layers could be used) to determine a super class of the actions, and then use a set of specialized classifiers to obtain a finer classification label. Thus, given the set of classes to classify, $\mathcal{A}$, there is a partition P over all the actions, such that $\cup_{A\in P} A = \mathcal{A}$ and $A_i \cap A_j = \emptyset$ for all $A_i, A_j \in P$, and let x be a video with action class $a \in A$. We intend to learn a function

$$\mathbb{C}(x) = g, \tag{1}$$

i.e., the action group classifier (at top level), such that it returns the group index, g, that represents a subset $A_g \in P$, that contains x's action, $a \in A_g$. Simultaneously, we intend to learn a set of classifiers

$$\mathcal{C} = \{C_g(x) : 1 \leq g \leq |P|\}, \tag{2}$$

such that C_g classifies the actions on the partition A_g. Hence, we perform the classification of x by finding the optimal $g^* = \mathbb{C}(x)$ as the result of the high level classifier (1), and then using it to select the low level classifier (2), C_{g^*}, that will be used to produce the final action.

We propose a neural network architecture (shown in Fig. 2) to approximate these classifiers. The architecture comprises Conv3D layers alternated with max polling 3D layers. As an activation function on the Conv3D layers, we used rectified linear units. The last layer is a softmax layer that performs the classification and its size is equal to the number of respective classes (that is, number of groups, $|P|$, and classes within each group, g, for each level, $|A_g|$, respectively).

In our network architecture, a 3D kernel with a size of $3\times3\times3$ is used, the efficiency of this kernel was experimentally proved by Tran et al. [45]. Similarly, Tran et al. verified that 3D ConvNet performs consistently better than 2D ConvNets on a large-scale internal dataset, namely I380K [45]. The rest of the network parameters (shown in Fig. 2) were also obtained from the studied literature presented in the related work as well as calibrated to solve our particular problem in the initial experiments.

We use the same network architecture at both levels of the hierarchical system. The same architecture is general enough to tackle the action group (high level) and action (low level) recognition. However, since we are training several instances of the network each one will learn different features, tuned for their particular partition.

IV. Experiments and Results

A. Setup

We conducted experiments on the 20 actions from the Kinetics dataset [1], since these actions had the lowest reported accuracy [60]. By accuracy we mean the probability that a

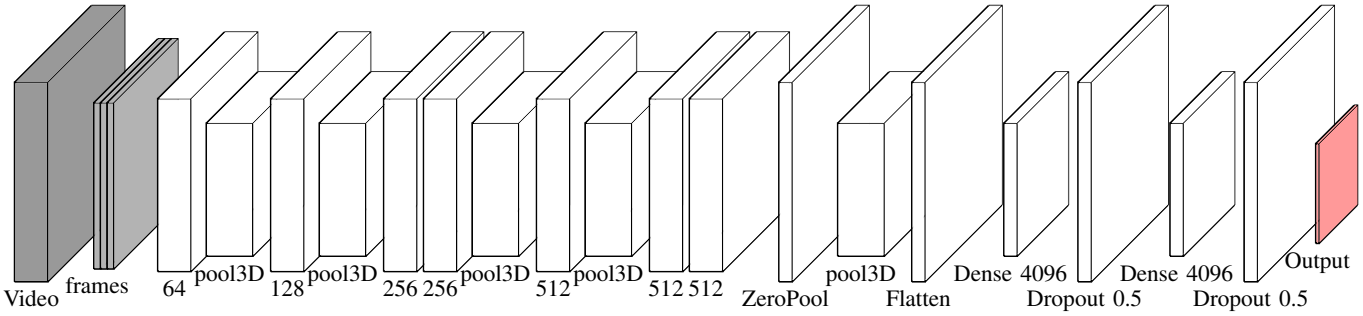

Figure 2. The neural network architecture for action recognition. It has 8 convolution layers of $3\times3\times3$ with 1 stride in both spatial and temporal dimensions, 5 max-pooling layers of $2\times2\times2$, except the first pool which is $1\times2\times2$), and 2 fully connected layers which are have 4096 output units, the last is softmax layer is output layer.

video containing an action that is submitted will be defined by the system as the correct action. Kinetics dataset was split in the ratio of 70% of the video to train, 20% of the video for validation and 10% of the video for the tests. The results displayed in the table II reflect the fact that the detected action will be determined by the system with an accuracy for example: recognizing the recording mucis system to determine with a probability of 34% that this recording music and 66% will be distributed among the remaining actions.

After we have partitioned the dataset into parts, we ran a script that separates each video into a sequence of frames. Firstly, we have a dataset consisting of video clips with duration of 10 seconds or less, depending on the length of the initial video. The original video on YouTube can be of any length; from a few hours to a few seconds. Kinetics dataset contains information about the range within which the target action is located. In accordance with this, video is cut from the YouTube video to get a short video containing only the desired action.

We used `ffmpeg` [61] to cut the video into frames. We can specify the required number of frames that we want to get from the video clip. The number of frames from the video are sampled regularly given a rate. Since Kinetics contains videos of different lengths, as a result of the frame generation step, a different number of frames is obtained for different videos. It is worth noting that most of the videos have a duration of 10 seconds.

In our experiments, we chose 10 second videos, and tested for 20, 40, 60, 80, and 100 as the number of frames. The aim was to determine the optimal number of frames to obtain the best recognition results, which turned out to be 60. Increasing the number of frames greatly slows the learning process, although it can possibly improve the quality of network learning and, as a consequence, the accuracy of the action recognition system. We used the recognition accuracy as the evaluation metric, as was done in the existing works [36].

As for the libraries, we used tensorflow [62] and keras [63] to construct, train, and test the networks. We used the Adam optimizer with learning rate 1×10^{-4}, batchsize of 32, initial learning rate of 0.004, and 50 epochs for training. As for the hardware, we used a computer equipped with Intel® Core™ i7-7700K CPU @ 4.20GHz, and GPU-GeeForce 1080ti with 8 GB of video memory.

Table II
EXPERIMENT'S RESULTS AVERAGE ACCURACY (%).

Experiment	**Avg. Recog. (%)**
Baseline scores [1]	14.0
1st exp.	6.5
2nd exp. with optical flow	4.4
3rd exp. hierarchical classification approach	36.0
4th exp. hierarchical classification with optical flow	34.7

B. Experiments

We do not use preprocessing in this work. For feature extraction, we tested optical flow approach Table II, but did not get a good accuracy. We then tried feature learning and came to the conclusion that feature learning is better that using hand-crafted features. Our findings are consistent with that of a study done by Antipov et al. [64], where they compared these two types of features. The study showed that hand-crafted and learned features perform equally well on small-sized homogeneous datasets. However, learned features significantly outperform hand-crafted ones in the case of heterogeneous and unfamiliar (unseen) datasets.

Hence, to fully evaluate our method, we performed four experiments by varying the type of feature with and without our proposed hierarchical system.

In the first experiment, we evaluated a single model that recognizes the 20 actions (that is the partition $P = \{\mathcal{A}\}$). We used the training and validation data to select the optimal parameters (as described in Section IV-A). Table II shows the average accuracy of recognizing the chosen 20 actions from the Kinetics dataset. The average recognition accuracy was 6.5%, which is less than half of the baseline accuracy as reported in the database's proposal [1]. The difference in results is due to the simplicity of our model in this experiment in contrast to the one used in the baseline [1]. In that work, the authors used a two-stream approach that combines the results of two networks: one working with pre-processed video by the

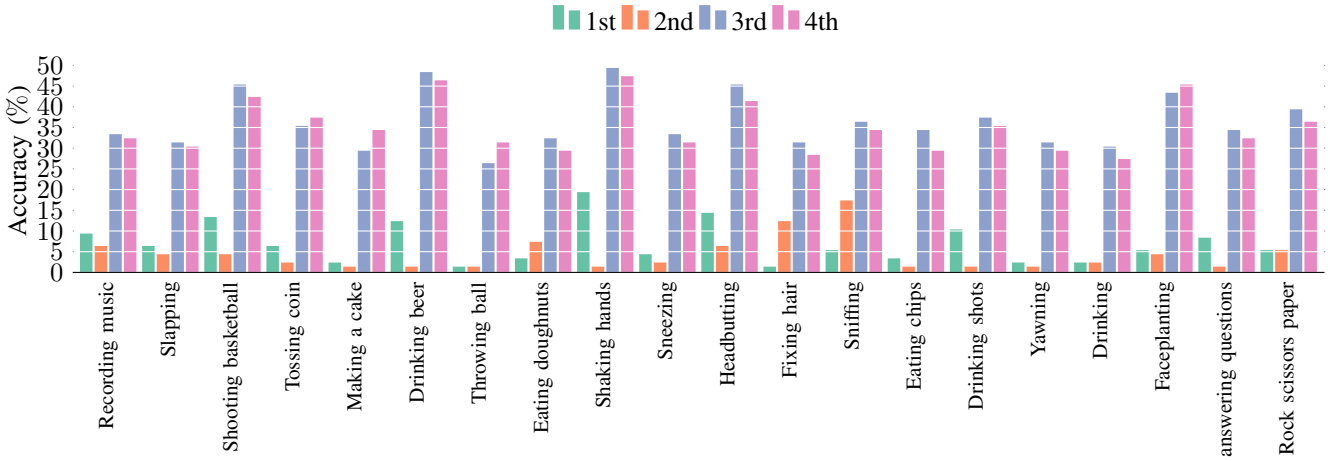

Figure 3. Recognition accuracy of each action in our four experiments.

optical flow algorithm, and a convolutional three-dimensional network working with conventional video.

In the second experiment, we tried to improve the accuracy of this single model by training with optical flow instead of raw data. The average recognition accuracy for all 20 actions with optical flow decreased by 2.1% (as shown in Table II). Although the average recognition accuracy is lower compared with the previous experiment, when examining Fig. 3, we can see that the use of optical flow resulted in a better accuracy for some actions, such as fixing hair and sniffing. With raw data as input, these actions were recognized with an accuracy of less than 5%, but with optical flow, it was over 15%. However, using the optical flow significantly lowered the recognition accuracy of shooting basketball, sneezing and others. Perhaps, this is the reason that the original work [1] used a two-stream approach: some actions are better recognized using raw video as input; whereas for others recognition works better when optical flow is used.

In the third experiment, we evaluated the performance of the proposed hierarchical recognition model. For both first and second levels, the same architecture of the Conv3d model used in the first experiment was used. The purpose of this experiment was to determine whether a hierarchical approach to action recognition would increase the accuracy. The results show an increase in accuracy by six times (cf. Table II and Fig. 3). This demonstrates that the approach is well-suited for recognizing difficult actions, better than the existing work. To determine each action on the lower level, there is a separate conv3D network such that it is trained to recognize only the actions of its subgroup (see Table I). This division of work greatly simplifies the network training, as it does not take into account actions from other groups. At the first level, there is also conv3D, which is not concerned with individual actions, but deals only with the definition of global classes. The division of recognition tasks into two levels simplifies the training of each network, which increases the average recognition accuracy. It is worth noting that the experiment was conducted not on the entire volume of video for each action and training took 50 epochs. Based on the accuracy and loss destruction graphs, it can be judged that the continuation of training on a larger number of epochs will give a result much higher than that presented in this paper. However, it was stopped due to hardware constraints.

In the fourth experiment, we tested the use of optical flow in the proposed hierarchical recognition model. The results of the experiment did not show a significant increase in recognition accuracy. The results of this experiment are only slightly different from the previous experiment. Figure 3 shows that for many actions the results are similar. Hence, we conclude that the application of the two-stream approach to further improve the system might not give significant improvement. It makes more sense to train the model from the previous experiment with a larger number of frames.

As reported earlier, [44] claimed to improve the accuracy for 9 out of the 20 most difficult actions. These include slapping, throwing ball, shaking hands, headbutting, fixing hair, sniffing, drinking, faceplanting, and rock-scissors-paper. The average accuracy of recognition of these actions [44] was 24.1%, which is worse in comparison with the results obtained using our hierarchical classification method, as shown in Table II.

V. Conclusion

In this work we addressed the problem of learning difficult-to-recognize actions due to large similarities among them. We proposed and tested a hierarchical two-level system for recognition. Moreover, our model allows to train the system on new actions without changing the whole model, by training on separate classifiers. For instance, we could train on the whole 400 classes on the dataset by creating the low level classifiers one time. Then, the high level layers could be tunned according to the recognition rates on different partitions. However, due to our limitations on hardware, we could not fully test this hypothesis due to the costly requirements in resources.

The ability to add new activities flexibly creates the need for new datasets containing more data. In our opinion, Kinetics dataset [1] has potential for its expansion due to the popularity of YouTube. Our results are better than existing solutions and this can be a new starting point for the further development of systems working with recognition of actions in a hierarchical way.

ACKNOWLEDGEMENT

This research work was supported by Zayed University Research Cluster Award # R18038, by the São Paulo Research Foundation (FAPESP grant # 2016/19947-6), and by the Brazilian National Council for Scientific and Technological Development (CNPQ grant # 307425/2017-7).

REFERENCES

[1] W. Kay, K. Simonyan, B. Zhang, C. Hillier, J. C. S. Vijayanarasimha and, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zissermank, "The kinetics human action video dataset," *Computer Vision and Pattern Recognition*, vol. 22, pp. 1–22, May 2017.

[2] J. K. Aggarwal and Q. Cai, "Human motion analysis: a review," *Comput. Vis. Image Underst*, vol. 73, pp. 428–440, Mar. 1999.

[3] I. Laptev, "On space-time interest points," *Computer Vision and Pattern Recognition*, vol. 8, Oct. 2005.

[4] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *Computer Vision and Pattern Recognition*, vol. 7, pp. 1–7, Dec. 2012.

[5] M. S. B. Maind and M. P. Wankar, "Research paper on basic of artificial neural network," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, p. 96 – 100, Jan. 2014.

[6] K. Soomro, A. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *Computer Vision and Pattern Recognition*, vol. 7, Dec. 2012.

[7] H. Kuehne, H. J. E. G. T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition.ieee international conference on computer vision," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 8, pp. 1–8, Nov. 2011.

[8] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 10, pp. 1–10, Jun. 2015.

[9] S. Wong and R. Cipolla, "Extracting spatio-temporal interest points using global information," *Computer Vision and Pattern Recognition*, vol. 8, Oct. 2007.

[10] H. Wang, A. Klaser, C. Schmid, and Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Computer Vision and Pattern Recognition*, vol. 34, May 2013.

[11] P. Scovanner, S. Ali, , and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," *Computer Vision and Pattern Recognition*, pp. 357–360, Jan. 2007.

[12] E. Valle, S. de Avila, A. da Luz Jr, F. de Souza, M. Coelho, and A. Araújo, "Content-based filtering for video sharing social networks," *Computer Vision and Pattern Recognition*, Jan. 2011.

[13] J. Donahue, L. A. Hendricks, M. Rohrbach, V. S, and T. D. Sergio Guadarrama, Kate Saenko, "Long-term recurrent convolutional networks for visual recognition and description," *Computer Vision and Pattern Recognition*, vol. 14, May 2016.

[14] A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," *Computer Vision and Pattern Recognition*, vol. 37, Oct. 2003.

[15] Z. Lin, Z. Jiang, and L. Davis, "Recognizing actions by shape-motion prototype trees," *Computer Vision and Pattern Recognition*, vol. 8, Sep. 2009.

[16] A. Jovic, K. Brkic, , and N. Bogunovic, "A review of feature selection methods with applications," *Computer Vision and Pattern Recognition*, May 2015.

[17] H. Chun-Lin and P. Wei, "Study on human action recognition algorithms in videos," *Computer Vision and Pattern Recognition*, vol. 214, pp. 1–214, 2015.

[18] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Computer Vision and Pattern Recognition*, 2005.

[19] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," *Computer Vision and Pattern Recognition*, 2007.

[20] W. Yang, Y. Wang, and G. Mori, "Human action recognition from a single clip per action," *Computer Vision and Pattern Recognition*, vol. 8, Sep. 2009.

[21] M. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," *Computer Vision and Pattern Recognition*, vol. 8, Jun. 2008.

[22] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," *Computer Vision and Pattern Recognition*, vol. 8, Jun. 2009.

[23] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," *Computer Vision and Pattern Recognition*, 2008.

[24] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *In IEEE Conference on Computer Vision and Pattern Recognition*, vol. 9, Jun. 2009.

[25] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," *Computer Vision and Pattern Recognition*, 2011.

[26] C. Xu, R. F. Doell, S. Hanson, C. Hanson, and J. Corso, "A study of actor and action semantic retention in video supervoxel segmentation," *Computer Vision and Pattern Recognition*, vol. 21, Nov. 2013.

[27] X. Wang, L. Wang, and Y. Qiao, "A comparative study of encoding, pooling and normalization methods for action recognition," *Computer Vision and Pattern Recognition*, vol. 14, pp. 572–585, Jun. 2013.

[28] A. Shabani, D. Clausi, and J. Zelek, "Salient feature detectors for human action recognition," *Computer Vision and Pattern Recognition*, vol. 8, May 2012.

[29] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," *Computer Vision and Pattern Recognition*, vol. 11, pp. 124.1–124.11, Sep. 2009.

[30] 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 Pattern Recognition*, vol. 22, May 2016.

[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Computer Vision and Pattern Recognition*, vol. 8, Jun. 2005.

[32] J. Wang, P. Liu, M. She, A. Kouzani, and S. Nahavandi, "Supervised learning probabilistic latent semantic analysis for human motion analysis," *Computer Vision and Pattern Recognition*, vol. 100, pp. 134–143, Jan. 2013.

[33] A. Kl¨aser, M. M. lek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," *Computer Vision and Pattern Recognition*, vol. 10, Sep. 2008.

[34] J. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos. part of: Advances in neural information processing systems 27," *Computer Vision and Pattern Recognition*, vol. 11, Nov. 2014.

[35] S. Ji, W. Xu, M. Yang, , and K. Yu, "3d convolutional neural networks for human action recognition," *Computer Vision and Pattern Recognition*, vol. 35, pp. 221 – 231, Jan. 2013.

[36] M. Baccouche, F. M. C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," *Conference: Human Behaviour Understanding*, vol. 39, Nov. 2011.

[37] B. Kolman and D. Hill, "Elementary linear algebra with applications," *Computer Vision and Pattern Recognition*, vol. 560, 2008.

[38] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," *Computer Vision and Pattern Recognition*, vol. 8, Jun. 2008.

[39] M. H. Siddiqi, S.-W. Lee, and A. M. Khan, "Weed image classification using wavelet transform, stepwise linear discriminant analysis, and support vector machines for an automatic spray control system." *Journal of Information Science & Engineering*, vol. 30, no. 4, 2014.

[40] T. R. D. Saputri, A. M. Khan, and S.-W. Lee, "User-independent activity recognition via three-stage ga-based feature selection," *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, p. 706287, 2014.

[41] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," *Computer Vision and Pattern Recognition*, vol. 10, Feb. 2017.

[42] H. Zhao, Z. Yan, H. Wang, L. Torresani, and A. Torralba, "Slac: A sparsely labeled dataset for action classification and localization," *Computer Vision and Pattern Recognition*, vol. 11, Dec. 2017.

[43] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. V. Gool, "Temporal 3d convnets: New architecture and transfer learning for video classification," *Computer Vision and Pattern Recognition*, vol. 9, Nov. 2017.

[44] L. Wang, W. Li, W. Li, and L. V. Gool, "Appearance-and-relation networks for video classification," *Computer Vision and Pattern Recognition*, vol. 12, Nov. 2017.

[45] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, F. A. Research, and D. College, "Learning spatiotemporal features with 3d convolutional networks," *ICCV*, vol. 9, Dec. 2015.

[46] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen, "Attention clusters: Purely attention based local feature integration for video classification," *Computer Vision and Pattern Recognition*, vol. 11, Nov. 2017.

[47] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *Computer Vision and Pattern Recognition*, vol. 10, Jan. 25 Jan 2018.

[48] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning for video understanding," *Computer Vision and Pattern Recognition*, vol. 10, Dec. 13 Dec 2017.

[49] L. Novak and R. Tailleux, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" *Computer Vision and Pattern Recognition*, vol. 10, Apr. 27 Nov 2017.

[50] ——, "On the local view of atmospheric available potential energy," *Computer Vision and Pattern Recognition*, vol. 40, Nov. 2017.

[51] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," *Computer Vision and Pattern Recognition*, vol. 10, Apr. 2017.

[52] C. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. P. Graf, "Attend and interact: Higher-order object interactions for video understanding," *Computer Vision and Pattern Recognition*, vol. 18, Mar. 2017.

[53] J. Zhu, W. Zou, and Z. Zhu, "End-to-end video-level representation learning for action recognition," *Computer Vision and Pattern Recognition*, vol. 10, Apr. 2018.

[54] M. J. Ferrarotti, S. Decherchi, and W. Rocchia, "Distributed kernel k-means for large scale clustering," *Computer Vision and Pattern Recognition*, vol. 18, Oct. 2017.

[55] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3d residual networks for action recognition," *Computer Vision and Pattern Recognition*, vol. 7, Aug. 2017.

[56] S. G. Lingala, Y. Guo, R. M. Lebel, Y. Zhu, Y. Bliesener, M. Law, and K. S. Nayak, "Tracer kinetic models as temporal constraints during dce-mri reconstruction," *Computer Vision and Pattern Recognition*, vol. 32, Jul. 2017.

[57] Y. Zhang, C. Chen, Z. Gan, R. Henao, and L. Carin, "Stochastic gradient monomial gamma sampler," *Computer Vision and Pattern Recognition*, vol. 10, Jan. 2018.

[58] C. C. Chapman and J.-B. Sallee, "Can we reconstruct mean and eddy fluxes from argo floats?" *Computer Vision and Pattern Recognition*, vol. 120, pp. 83–100, Dec. 2017.

[59] D.-T. Hoang, J. Song, V. Periwal, and J. Jo, "Maximizing weighted shannon entropy for network inference with little data," *Computer Vision and Pattern Recognition*, vol. 5, May 2017.

[60] K. Sozykin, A. Khan, S. Protasov, and R. Hussai, "Multi-label class-imbalanced action recognition in hockey videos via 3d convolutional neural networks," *Computer Vision and Pattern Recognition*, vol. 9, Sep. 2017.

[61] F. Developers, "Webpage ffmpeg," *http://ffmpeg.org/*, 2016.

[62] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Z. G. Research, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *Unknown*, vol. 19, Nov. 2015.

[63] K. Developers, "Keras documentation," *https://keras.io/*, 2015.

[64] G. Antipov, S.-A. Berrani, N. Ruchaud, and J.-L. Dugelay, "Learned vs. hand-crafted features for pedestrian gender recognition," *MM '15 Proceedings of the 23rd ACM international conference on Multimedia*, vol. 3, pp. 1263–1266, Oct. 2015.