3D CNN for Human Action Recognition

Sameh Neili Boualia[1,2]
[1] *University of Tunis El Manar, National Engineering School of Tunis,* 1002, Tunis, Tunisia;
sameh.neili@gmail.com

Najoua Essoukri Ben Amara [2]
[2] *Université de Sousse, Ecole Nationale d'Ingénieurs de Sousse, LATIS-Laboratory of Advanced Technology and Intelligent Systems,* 4023, Sousse, Tunisie;
najoua.benamara@eniso.rnu.tn

2021 18th International Multi-Conference on Systems, Signals & Devices (SSD) | 978-1-6654-1493-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/SSD52085.2021.9429429

***Abstract*—Recognizing different human actions from still images or videos is an important research area in the computer vision and artificial intelligence domains. It represents a key step for a wide range of applications including: human-computer interaction, ambient assisted living, intelligent driving and video surveillance. However, unless the many research works being involved, there are still many challenges ahead including: the high changes in human body shapes, clothing and viewpoint changes and the conditions of system acquisition (illumination variations, occlusions, etc). With the emergence of new deep learning techniques, many approaches are recently proposed for Human Action Recognition (HAR). Compared with conventional machine learning methods, deep learning techniques have more powerful learning ability. The most wide-spread deep learning approach is the Convolutional Neural Network (CNN/ConvNets). It has shown remarkable achievements due to its precision and robustness. As a branch of neural network, 3D CNN is a relatively new technique in the field of deep learning. In this paper, we propose a HAR approach based on a 3D CNN model. We apply the developed model to recognize human actions of KTH and J-HMDB datasets, and we achieve state of the art performance in comparison to baseline methods.**

***Index Terms*—Human Action Recognition, Deep Learning, 3D CNN**

I. Introduction

Deep learning approaches have been investigated since the 1960s [1] but researchers have paid little attention towards them. This was mainly due to the success of shallow models such as SVMs [2] and the unavailability of huge amount of data required for training the deep models. According to the literature, the most wide-spread deep learning approach is Convolutional Neural Networks (CNN/ConvNets). This type of deep learning technique has shown excellent performances for different tasks such as handwritten digit classification [3], pattern recognition [4] and image classification [5]. Actually, the CNN obviate the need for the extensive pre-processing steps that were necessary in conventional approaches. Therefore, the deep learning based algorithms are becoming faster and more computationally efficient. Moreover, they provide several layers of feature extractors that make it easier to learn implicitly the patterns corresponding to each feature. Their hierarchical learning model with multiple hidden layers transforms the input volume into the desired output. As shown in Fig.1, the typical CNN architecture consists of three main types of layers: Convolution and Rectifier Linear Unit layers (CONV + ReLu), pooling layer (POOL), and fully-connected layer (FC). Therefore, this new

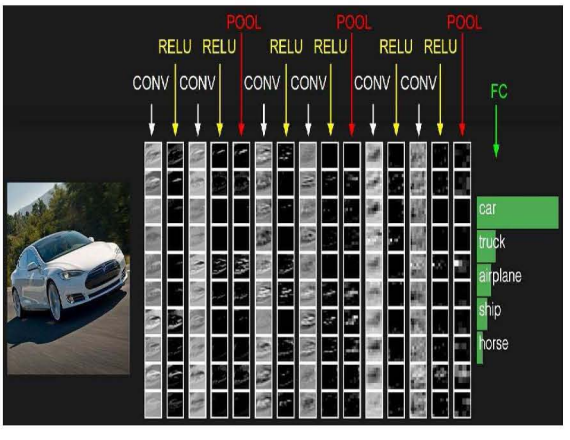

Fig. 1. The typical arrangement of various layers of CNN architecture [6]

type of technology does not need hand-crafted feature anymore. Owing to all these benefits, deep learning methods have had a large impact on a plethora of research areas including object detection and classification [7], [8], pedestrian detection [9]–[11], segmentation [12], pose estimation [13]–[15], gesture detection and localization [16], [17], video action classification [18], [19]. For Human Action Recognition (HAR) (such as 'walking', 'open door', 'sit down'...), many approaches based on deep learning techniques have been proposed. In the following, we present a selection of CNN based research works for HAR which can be classified into two main groups: 2D CNN based and 3D CNN based approaches.

For 2D CNN based HAR approaches, Simonyan et al. [20] implemented a two-stream ConvNets where the spatial stream recognizes the action from still frames and the temporal stream performs recognition from the motion in the form of dense optical flow. This method achieved good results on UCF-101 and HMDB-51 datasets. However, according to authors, the proposed model may not be suitable for real-time applications due to its computational complexity. Moreover, in [21], the authors adapted the successful deep learning architectures to the design of a two-stream ConvNets for action recognition in videos, which they called 'very deep two-stream ConvNets'. They empirically studied both GoogLeNet and VGG-16 for the design of such proposed model. In relation to [20], they presented two novelties: i) they extended the famous Caffe toolbox into Multi-GPU implementation with high efficiency and low memory consumption and ii) they proposed several good practices for the training of ConvNets architecture (learning rate arrangement, data augmentation techniques...). For evaluation, UCF101 dataset was used with which they achieved a recognition accuracy of 91.4%. Later on, Ijjina et al. [22] proposed a new approach for HAR based on genetic algorithms (GA) and CNN. They demonstrated that initializing the weights of a CNN classifier based on solutions generated by GA minimizes the classification error. To demonstrate the efficacy of the proposed classification system, they evaluated their CNN-GA model on UCF50 dataset achieving 96.88% as an average accuracy rate.

Most of the current CNN methods use architectures with 2D convolutions, enabling shift-invariant representations in the image plane. However, the invariance to translations in time axis is also important for HAR since the beginning and the end of the action are generally unknown [23]. Thus, CNN with 3D spatio-temporal convolutions addresses this issue and provides a natural extension of 2D CNN to video. In [24], the authors developed a novel deep model for automatic activity recognition from RGB-D videos. Each human activity was presented as an ensemble of cubic-like video segments, and learned to discover the temporal structures for each category of activities. Their proposed ConvNets model-based consists of 3D convolutions and max-pooling operators over the video segments. Later, Shao et al. [25] mixed appearance and motion features for recognizing group activities in crowded scenes collected from the web. For the combination of the different modalities, the authors applied multitask deep learning. By these means, they were able to capture the intra-class correlations between the learned attributes while they proposed a novel dataset of crowed scene understanding called 'WWW crowd' dataset. Another approach using spatio-temporal features with a 3D convolutional network was proposed in [26]. Experimentally, the authors showed that 3D CNN are more suitable for spatio-temporal features than 2D CNN. Also, they empirically demonstrated that the CNN architecture with small $3 \times 3 \times 3$ kernels is the best choice for spatio-temporal features. Achieving 52.8% accuracy on UCF101 dataset, their model was computationally efficient due to the fast inference of ConvNets. Just recently, Varol et al. [27] proposed LTC-CNN model: a combination of long-term temporal convolutions (LTC) with CNN in order to learn video representations. They have investigated multi-resolution representations of both motion and appearance. They have demonstrated the importance of high-quality optical flow estimation on action recognition accuracy. The model was tested on two recent and challenging human action benchmarks: UCF101 and HMDB51 and has reported state of the art performance. Shou et al. [28] have also designed a novel 3D CNN model named: Convolutional-De-Convolutional (CDC) network where CDC filters were implemented prior to a 3D ConvNets. Shou et al. are the first to combine two reverse operations

(convolution and de-convolution) into a joint CDC filter. The proposed CDC conducted simultaneously down-sampling in space and up-sampling in time to infer both high-level action semantics and temporal dynamics.
In the following, we detail the 3D CNN model proposed for a HAR approach.

II. Deep HAR proposed approach

In 2D CNN, convolutions are applied on the 2D feature maps to compute features from only spatial dimensions. When applied to video analysis problems, it is desirable to capture the motion information encoded in multiple contiguous frames. To this end, we propose to perform 3D convolutions in the convolution stages of the CNN to compute features from both spatial and temporal dimensions. The 3D convolution is achieved by convolving a 3D kernel to the cube formed by stacking multiple adjacent frames together. By this construction, the feature maps in the convolution layer are connected to multiple neighboring frames in the previous layer, thereby capturing motion information. Figures 2.(a) and (b) illustrate the generic representation of the 2D and 3D convolutions, respectively. As explained in [29], the 2D convolution extracts the independent features from a sequence of images with several 2D convolution kernels sliding along the M and N axis as shown in Fig.2(a). In contrast, the 3D convolution captures both spatial and temporal information using a 3D convolution kernel sliding along the M, N and S axis as shown in Fig.2(b). Formally, the value of an unit at a position (m, n, s) on the j^{th} feature map in the i^{th} layer is calculated according to Eq.1 given by [30].

$$V_{ij}^{mns} = tanh(b_{ij} + \sum_{k} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijk}^{pqr} v_{(i-1)k}^{(m+p)(n+q)(s+r)}) \quad (1)$$

where $tanh()$ is the hyperbolic tangent function, b_{ij} is the bias for the j^{th} feature map, k indexes over the set of feature maps in the $(i-1)^{th}$ layer connected to the current feature map, R_i is the size of 3D kernel along the temporal dimension, P_i and Q_i are the height and width of the kernel respectively and

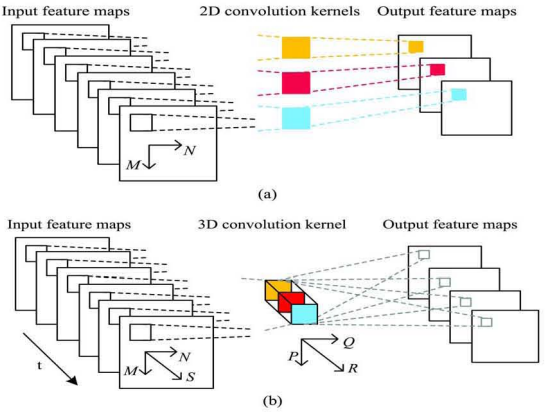

Fig. 2. Comparison between 2D and 3D convolution layers, (a) 2D convolution kernels, (b) 3D convolution kernels [29]

w_{ijk}^{pqr} is the $(p, q, r)^{th}$ value of the kernel connected to the k^{th} feature map in the previous layer.

Generally, the 2D CNN are designed as that the number of feature maps should be increased in late layers by generating multiple types of features from the same set of lower-level feature maps. As for 3D CNN, this process can be achieved by applying multiple 3D convolutions with distinct kernels to the same location in the previous layer. Note that a 3D convolutional kernel can only extract one type of features from the frame cube, since the kernel weights are replicated across the entire cube.
In this work, we propose a HAR approach based on a 3D CNN architecture. Indeed, the 3D CNN, using three-dimensional data as input, have a better adaptability to the data with continuous temporal and spatial domain characteristics of the video. The architecture of the proposed model, shown in Fig.3, is composed of two *Conv3D* layers with a kernel size 5×5 and a depth size 32, followed by their *ReLu* layers. To perform a down-sampling operation along the spatial dimensions, *MaxPooL* layer is used. It operates independently on every depth slice of the input and resizes it spatially using the MAX operation with a kernel size 3×3 and a depth size 3. Thus, this layer allows us to reduce representation size and speed up the computation. To prevent overfitting during training process, *Dropout* layers are used with a rate of 0.5. In order to perform classification on the features extracted by the *Conv3D* layers and down-sampled by the *MaxPooL* layers, we used two *Dense* (or fully connected) layers. The last *Dense*

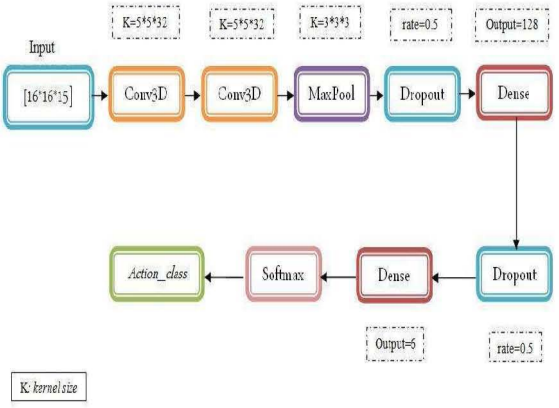

Fig. 3. Overview of the proposed 3D CNN model

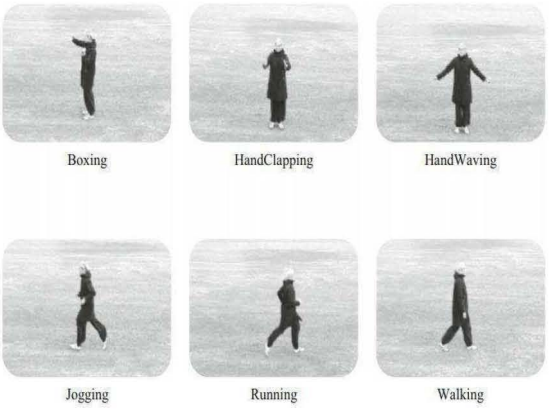

Fig. 4. Examples from the KTH dataset

layer has an output size fixed to 6 which represents the number of classes to recognize. As an activation function, we use *softmax* layer.

III. Experiments and Results

In this section, the implementation of the proposed 3D CNN model and achieved results upon used datasets are described.

A. Datasets

In order to evaluate the proposed model performance, we used the well known datasets: KTH [31] and J-HMDB [32].

KTH database: contains six types of human actions: *walking, jogging, running, boxing, hand waving and hand clapping*, performed by 25 subjects in four different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4). Currently the database contains 2391 sequences. All sequences were taken over homogeneous backgrounds with a static camera with 25fps frame rate. The sequences were down-sampled to the spatial resolution of 160×120 pixels, and they have a length of four seconds on average. Therefore, there are $25 \times 6 \times 4 = 600$ video files for each combination of 25 subjects, 6 actions, and 4 scenarios. We followed the original experimental setup of [33]. Some action samples of different classes are shown in Fig.4.

J-HMDB dataset: represents a fully annotated dataset for human actions and poses. It contains 928 videos collected from movies and Youtube with 15 fps (i.e 31838 frames with 240×320 resolution). The

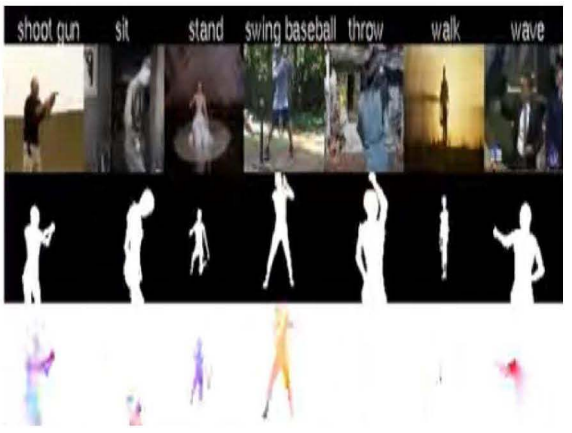

Fig. 5. Examples from the J-HMDB dataset

21 action classes are conducted with one main actor. Some sample frames are shown in Fig.5. In order to make the experimental results better, we continuous tried new network parameters, modified the size of the input images and increased/decreased the batch size and/or the number of epochs. The accuracy metric was used to evaluate the performance of the proposed approach [34].

B. Implementation Details

In this section, we describe the implementation details of our approach. 3D CNN training was performed on a single NVIDIA GTX Titan GPU. To implement the proposed model, we used the two Python deep learning libraries: Keras (with Theano backend) and Scikit-learn [35], [36], which represent model-level libraries providing high-level building blocks for developing deep learning mod-

els. We conducted several experiments in order to evaluate the network parameters initialization effect, such as the number of training iterations on the predicted results. The different implementation details are presented in Tab.I.

TABLE I
IMPLEMENTATION DETAILS OF THE PROPOSED MODEL

Batch size	2
Number of epochs	50
Loss function	categorical_crossentropy
Optimizer	RMS
Evaluation metric	Accuracy

As shown in Tab.I, we take use of the *categorical cross-entropy* as a loss function. Indeed, cross-entropy loss measures the performance of a classification model whose output is a probability value. It increases as the predicted probability diverges from the actual label. In multi-class classification, cross-entropy loss is calculated according to Eq.2.

$$loss = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \qquad (2)$$

where M is the number of classes to predict, y is the binary indicator (0 or 1) if class label c is the correct classification for observation o, and p represents the predicted probability observation o of a class c.

C. Results

In order to evaluate the network parameters initialization effect, we conducted several experiments by varying some of them such as the number of epochs and training iterations. The network was trained from scratch in order to perform weights initialization. We began with a number of epochs equal to 1 and increased it in order to view its effect on the convergence of the loss to 0. To present maximally varying input data to the network and avoid over-fitting problem, each frame is randomly augmented and shuffled prior to training. The validation set is used for hyper-parameters estimation. In our work, we trained and evaluated multi-class classifier and reported the accuracy average over all classes. The accuracy of HAR is evaluated on KTH an J-HMDB datasets. We achieved an average accuracy rate of 78% and 90% on KTH and J-HMDB respectively, compared with state of the art methods as shown in Tab.II and Tab.III.

TABLE II
COMPARISON WITH STATE OF THE ART RESULTS ON KTH DATASET

References	**Accuracy (%)**
Schuldt et al. [31]	77
Dollar et al [37]	81.2
Nieble et al. [38]	81.5
Ji et al. [30]	90.2
Ours	78

TABLE III
COMPARISON WITH STATE OF THE ART RESULTS ON J-HMDB DATASET

References	**Accuracy (%)**
Nie et al. [39]	61.2
Cheron et al. [40]	66.8
Mavroudi et al. [41]	70
Ours	90

In Fig.6 and Fig.7, the accuracy curve during training and validation process is reported according to the number of epochs for both KTH and J-HMDB datasets respectively.

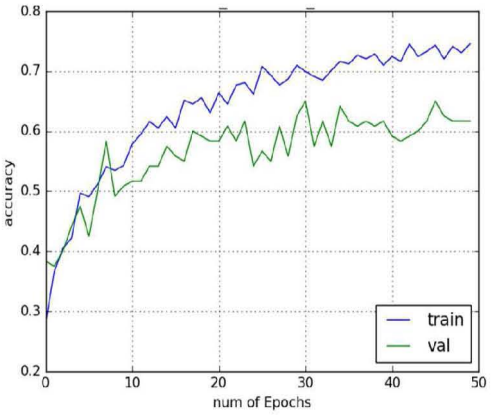

Fig. 6. Train vs. validation accuracy on KTH dataset according to the number of epochs

In Fig.8, we present the loss curve during training and validation process according to the number of epochs on KTH dataset.

IV. CONCLUSION

This paper addresses the importance of automatic understanding and characterization of human action. We proposed a 3D CNN model to recognize human actions. This model constructs features from both spatial and temporal dimensions by performing 3D

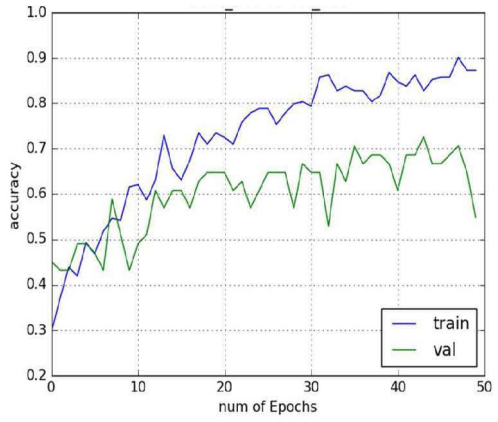

Fig. 7. Train vs. validation accuracy on J-HMDB dataset according to the number of epochs

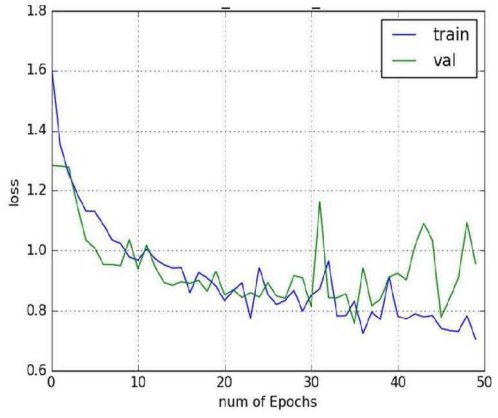

Fig. 8. Train vs. validation loss on KTH dataset according to the number of epochs

convolutions. We have validated our approach on the KTH and J-HMDB datasets. We have shown that 3D CNN architecture can be a very useful tool for recognizing human action without the need for hand-tuned foreground segmentation or any pre-processing steps. As future work, we will improve the proposed 3D CNN model through exploring more data augmentation techniques and enhancing the training parameters.

REFERENCES

[1] A. G. Ivakhnenko, "Polynomial theory of complex systems," *IEEE transactions on Systems, Man, and Cybernetics*, no. 4, pp. 364–378, 1971.

[2] V. Vapnik, S. E. Golowich, and A. J. Smola, "Support vector method for function approximation, regression estimation and signal processing," in *Advances in neural information processing systems*, 1997, pp. 281–287.

[3] N. Yu, P. Jiao, and Y. Zheng, "Handwritten digits recognition base on improved lenet5," in *Control and Decision Conference (CCDC), 2015 27th Chinese*. IEEE, 2015, pp. 4871–4875.

[4] E. Raschman, R. Záluskỳ, and D. Ďuračková, "New digital architecture of cnn for pattern recognition," *Journal of Electrical Engineering*, vol. 61, no. 4, pp. 222–228, 2010.

[5] S.-J. Lee, T. Chen, L. Yu, and C.-H. Lai, "Image classification based on the boost convolutional neural network," *IEEE Access*, vol. 6, pp. 12 755–12 768, 2018.

[6] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[7] S. Zuffi, J. Romero, C. Schmid, and M. J. Black, "Estimating human pose with flowing puppets," in *proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3312–3319.

[8] S. P. Singh, L. Wang, S. Gupta, B. Gulyás, and P. Padmanabhan, "Shallow 3d cnn for detecting acute brain hemorrhage from medical imaging sensors," *IEEE Sensors Journal*, pp. 1–1, 2020.

[9] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1904–1912.

[10] A. Mhalla, H. Maamatou, T. Chateau, S. Gazzah, and N. EssoukriBenAmara, "Faster r-cnn scene specialization with a sequential monte-carlo framework," in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Nov 2016, pp. 1–7.

[11] F. Gomez-Donoso, E. Cruz, M. Cazorla, S. Worrall, and E. Nebot, "Using a 3d cnn for rejecting false positives on pedestrian detection," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–6.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.

[13] A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt, "Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 3810–3818.

[14] X. Fan, K. Zheng, Y. Lin, and S. Wang, "Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation," *arXiv preprint arXiv:1504.07159*, 2015.

[15] S. Neili, S. Gazzah, M. A. El Yacoubi, and N. Essoukri Ben Amara, "Human posture recognition approach based on convnets and svm classifier," in *Advanced Technologies for Signal and Image Processing (ATSIP), 2017 International Conference on*. IEEE, 2017, pp. 1–6.

[16] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Workshop at the European conference on computer vision*. Springer, 2014, pp. 474–490.

[17] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, and M. A. Mekhtiche, "Hand gesture recognition for sign language using 3dcnn," *IEEE Access*, vol. 8, pp. 79 491–79 509, 2020.

[18] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, "R-cnns for pose estimation and action detection," *arXiv preprint arXiv:1406.5212*, 2014.

[19] J. Li, X. Liu, W. Zhang, M. Zhang, J. Song, and N. Sebe, "Spatio-temporal attention networks for action recognition and detection," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2990–3001, 2020.

[20] 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.

[21] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015.

[22] E. P. Ijjina and K. M. Chalavadi, "Human action recognition using genetic algorithms and convolutional neural networks," *Pattern recognition*, vol. 59, pp. 199–212, 2016.

[23] B. Seddik, S. Gazzah, and N. Essoukri Ben Amara, "Human-action recognition using a multi-layered fusion scheme of kinect modalities," *IET Computer Vision*, vol. 11, no. 7, pp. 530–540, 2017.

[24] K. Wang, X. Wang, L. Lin, M. Wang, and W. Zuo, "3d human activity recognition with reconfigurable convolutional neural networks," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 97–106.

[25] J. Shao, K. Kang, C. Change Loy, and X. Wang, "Deeply learned attributes for crowded scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4657–4666.

[26] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4489–4497.

[27] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[28] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 1417–1426.

[29] H. Wang, M. Shao, Y. Liu, and W. Zhao, "Enhanced efficiency 3d convolution based on optimal fpga accelerator," *IEEE Access*, vol. 5, pp. 6909–6916, 2017.

[30] 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.

[31] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3. IEEE, 2004, pp. 32–36.

[32] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *International Conf. on Computer Vision (ICCV)*, Dec. 2013, pp. 3192–3199.

[33] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013.

[34] J. Liu, L. Chen, J. Tian, and D. Zhu, "Learning-based leaf occlusion detection in surveillance video," in *Industrial Electronics and Applications (ICIEA), 2016 IEEE 11th Conference on*. IEEE, 2016, pp. 1000–1004.

[35] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[37] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*. IEEE, 2005, pp. 65–72.

[38] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International journal of computer vision*, vol. 79, no. 3, pp. 299–318, 2008.

[39] B. X. Nie, C. Xiong, and S.-C. Zhu, "Joint action recognition and pose estimation from video," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 1293–1301.

[40] 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.

[41] E. Mavroudi, L. Tao, and R. Vidal, "Deep moving poselets for video based action recognition," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017, pp. 111–120.