MPEG Bitstream for Video Classification with Deep Learning

Alexander-Andre Stefani
Chair for Data Processing, Technical University of Munich
ga63maq@mytum.de

***Abstract*—Today's neuronal networks for action recognition often use a pixel-based representation of videos as input for classification. The disadvantage for this type of representation is the high redundancy between frames and therefore the resulting complexity of networks. A possible approach for reducing the complexity of the networks is to perform classification in the compressed video domain as video encoders reduce redundancies in videos to optimize them for storage and transmission. Motivated by this, in this paper I intend to find an appropriate approach to use components of compressed videos for action recognition, while reducing the overall complexity of the network, and investigate the possibility to describe temporal relations with motion vectors. Experimental results show, that the developed network is not performing as well as expected since the part of the network using motion vectors as input is not able to learn successful temporal features.**

Keywords*—*Deep Learning, Neuronal Networks, Compressed Videos, MPEG, Action Recognition.

I. Introduction

The amount of videos in the internet traffic increased enormously in the last few years[1] [1]. With the growing influence of videos to our daily life, the relevance of computer vision tasks is also growing. Thereby, many state-of-the-art deep learning approaches use an intrinsic way to feed the neuronal networks with the videos as sequence of RGB frames [2]. However, this type of representation of videos is mostly inefficient, as it is memory and computationally intensive. The reason for this is the structure of videos as there are heaps of redundant information. This means that it may happen that within several frames only few new information occur. To extract thereby the relevant information, it is necessary to apply complex neuronal networks, that have high computational and storage requirements [1][3]. As today's encoders try to minimize the redundant information to reduce the storage size of videos, one could profit from this aspect. That would imply that, theoretically, neuronal networks fed with encoded videos can achieve the same results as fed with decoded videos, but with fewer parameters, because the networks can learn on the relevant information rather than on the repetition of almost similar signals [3][4]. To get one step closer to this long-term goal, this work investigates the approach to feed neuronal networks with partly decoded videos to reduce redundancies and such as the number of parameters. Therefore the following research questions are examined:

- Is it possible to reduce the complexity of neuronal networks with parts of compressed videos as input compared to neuronal networks using a pixel-based representation of videos as input?
- Are motion vectors capable to describe temporal relations?
- Is it possible to replace the commonly used optical flow with the motion vectors?

The main focus for video classification is on action recognition as it is one of the most important tasks in the field of video understanding [5]. First, existing projects with compressed and uncompressed videos are presented, that are relevant in the context of this work. Second, a possible deep learning architecture is investigated in the context of experiments to observe, if it produces a significant reduction of parameters, with comparable performance, compared to approaches that use a pixel-based representation of videos. In addition it will be inspected, if the optical flow, which is regularly used for action recognition, can be replaced by a component of the encoded video, the motion vectors. The paper concludes with a discussion of the results.

II. State-of-the-art

A. Action Recognition

Action recognition in videos is the task of classification actions from a sequence of observations. The attention for action recognition started in the 1980s as it is useful for a wide diversity of applications. Since then, a large variety of promising approaches were developed. The achieved performances have a broad range and will be presented and compared with the in this work achieved accuracy. The modeling of long range temporal information and high computational costs showed to be problems for the development of action recognition systems [5][6].

1) Pixel-based Networks for Action Recognition: Pixel-based approaches, where the RGB frames are used as inputs, are the most natural way to interpret and classify videos. As Convolutional Neuronal Networks (CNN) provide good performance on images, these are mostly used to extract spatiotemporal features from videos. Karpathy et al. [7] proposed an architecture with two spatial streams. Hereby one stream is fed with low resolution frames and the other one with high resolution frames. Both streams use the same network architecture, are concatenated and fed in two fully connected layers. This network is originally trained on the Sports-1M

[1] https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

dataset, but with fine tuning of the top three layers it is transferred to perform on UCF101 [7].
Another pixel-based approach is proposed by Tran et al. [8], who introduced a 3D CNN with 8 convolutional layers. As last layer a support vector machine (SVM) is used instead of a Softmax layer. They determined that a kernel size of $3\times3\times3$ shows the best performance for the convolutional layers. As input 16 video frames are stacked [8].
Carreira et al. [9] took advantage of the good performance of 2D CNNs on images and tried to convert the on ImageNet trained weights of the convolutional layers to 3D kernels by repeating the 2D kernel N times such that a $N \times N$ kernel becomes a $N \times N \times N$ kernel. The network contains 4 convolutional layers combined with 9 so called Inception modules [9].

2) Two-Stream Networks: Beneath concepts like Temporal Segment Networks (TSN), 3D Convolutional Networks and Recurrent Neuronal Networks there is the concept of Two-stream networks for action recognition [2]. Initially Simonyan et al. [6] took advantage of the fact, that single frame models already display strong performances, but with the addition of motion information the whole performance can be lifted to a new level. The reason for this is the information in the pattern of movement over multiple frames, that can not be extracted from single frames. Therefore the most effective way to describe the motion in videos is the so called optical flow. The only downside is the expensive calculation of the optical flow and therefore the infeasibility for real-time applications. As mentioned the originally proposed Two-stream network by Simonyan et al. [6] consists of two different branches. One branch extracts the spatial information of a single video frame. The other branch extracts the temporal information of stacked optical flow frames along the third dimension. Both streams use 2D convolutional layers to extract the features. There are several methods for merging both streams like averaging or SVM. Nevertheless, the classical approach of the two-stream model has some disadvantages. Firstly, it is not able to gather long term motion with the optical flow. Secondly, the spatial stream has the drawback to depend on randomly selected single video frames and thus is prone to clutter and viewpoint variations [2]. Based on the originally idea more variations of two-stream networks with improved performance established [5] [10].

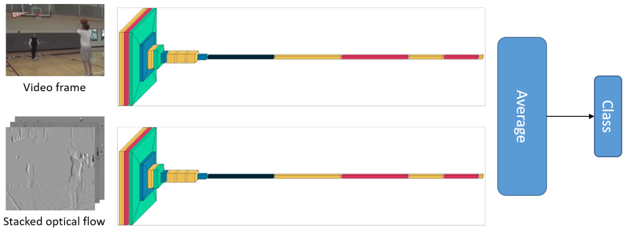

Fig. 1. Two-stream network redrawn from [6].

B. Video Compression

In the 1990s first approaches started to develop codecs to store and transmit videos efficiently. Therefore, they took advantage of the property of videos, that consecutive frames can be partly very similar. Nowadays, most of the codecs work relatively comparable. They group a defined number of frames together, called group of pictures (GOP). A GOP consists of three different types of frames. At the beginning of every GOP there is an intra-coded frame (I-frame), which is a regularly compressed image. The remaining frames are either predictive frames (P-frames) or bi-directional frames (B-frames). In Fig. 2 a possible structure of a GOP is shown. P-frames are predicatively coded images. Therefore the codec includes the previous I- or P-frame. For this the codec estimates and saves so called motion-vectors from the previous frame to the current one. The residuals compensate possible errors between the predicted and original frames after the motion estimation. They belong to P-frames as well. The proceeding for B-frames is related to the proceeding of P-frames, but hereby the codec includes a future I-frame. Thereby most of the present codecs nowadays work block-based. This means that coded frames are composed of sequences of macro-blocks. Through the prediction of image data the codec can reduce the needed bit rates for transmissions. B-frames show the best compression ratios, but they are the biggest expense to estimate [3][4][11][12].

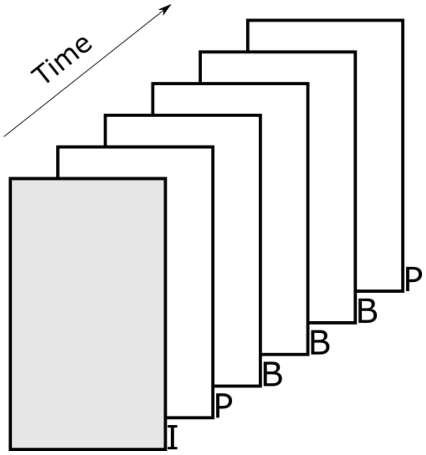

Fig. 2. Example of a GOP and a possible order of the different types of frames [11].

C. Compressed Video Action Recognition

For video classification tasks already some work was put in human action classification and mostly the classification is only performed with the motion vectors [4]. Chadha et al. [1] proposed a 3D CNN with a sequence of motion vectors as input to classify human actions on the HMDB51 and UCF101 database. The network achieved an overall accuracy similar to single stream networks using only optical flow as input, while reducing the complexity of the neuronal network compared to these networks.

Shou et al. [13] introduced a generator network to estimate an optical flow similar tensor with the help of motion vectors to classify the human action with it afterwards. Wu et al. [3] used thereby three different components of the compressed videos: I-frames, motion vectors and residuals. Each component is classified with its own CNN and the predictions are fused afterwards for one frame. This process is repeated with several frames and the results are averaged along the chosen frames [3][13]. The described methods can so far register a reduced complexity of their models, but accuracies on the same level as state-of-the-art networks for video classification can only be achieved with components of non-compressed videos [1][3][13]. Besides action recognition, compressed videos have been used in other topics, such as segmentation and object tracking. Some approaches can already keep up with state-of-the-art methods. Mostly the motion vectors are used as input of the neuronal networks [4] [12].

III. Action Recognition

A. Approach

The way videos are structured, when they are compressed, offers to feed them into neural networks in a similar way as it is done in the Two-stream network [1]. This means, that the already available I-frames and motion vectors provide a simple and fast way to feed the components into a Two-stream network without the need for a lot of pre-processing, as for example needed for the computation of the optical flow. So the approach is to take the original idea of the in subsubsection II-A2 mentioned Two-stream network for action recognition and replace the inputs of both streams with the components of the bitstream. For the spatial stream a sequence of I-frames is used to extract the information. The reason for replacing a single frame with a sequence is the idea to improve the spatial estimation as one randomly picked frame can potentially contain wrong information about the shown action due to noise, camera movement and so on [2]. With the use of multiple frames the influence of disruptive elements to the prediction could be minimized. Pre-trained networks from image classification showed to achieve the best results for the spatial streams [9]. Accordingly, it also makes sense to apply them to the extracted I-frames. For the temporal stream stacked motion vectors will be used, as they need no expensive calculation compared to the optical flow. In addition they should roughly describe the same pattern as the optical flow. Because of the sparse shape an increased number of motion vectors is needed, compared to the optical flow [1]. This fact could be helpful to gather long term motion. Therefore, the 3D CNN proposed by Chadha et al. [1] suits the best for the temporal stream as it achieved satisfying results with reduced complexity. Thus the two components are a sequence of I-frames and the stacked motion vectors as both, spatial and temporal information, are represented. The residual could be added as well, but Wu et. al [3] showed, that the influence of this component is small compared to the two other ones.

B. Network Input

For the task of action recognition the UCF101 dataset is used. It consists of 101 natural human actions with videos collected from YouTube. It represents a great variety of realistic videos like surfing, riding and many other. In total there are over 13k clips with a mean length of 7.21 seconds. The resolution of the videos is 320×240 and they are provided as AVI format [10] [14]. Before use the videos are converted to the H264 format. Due to a macroblock size of 8×8 the motion vectors have a dimension of $40 \times 30 \times 2$. The last dimension describes the movement of the pixels in x- and y-direction. The temporal dimension is set to $T = 160$ as this is the mean number of P-frames in the dataset. Chadha et.al [1] showed for this number of P-frames, that the best results can be achieved. Furthermore, the small spatial dimension can be balanced with the large temporal dimension. For the spatial branch a sequence of I-frames is chosen. A single I-frame has the same resolution as a frame of the video: $320 \times 240 \times 3$. The number of I-frames for spatial feature extraction is set to 4 as this number showed the best results between performance and input size. Both inputs, the motion vectors as well the as the I-frames, are cropped in the spatial dimensions to be independent from the input video size. Thus the final dimensions are $M \in \mathbb{R}^{160\times24\times24\times2}$ for the motion vectors and $I \in \mathbb{R}^{4\times224\times224\times3}$ for the I-frame tensors.

C. Architecture

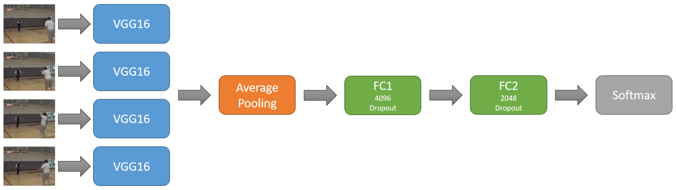

Fig. 3. Architecture of the spatial stream. FC1 and FC2 denote the 2 fully connected layers.

In the originally proposed Two-stream network architecture a single video frame is used to extract spatial features. The frame is fed into a 2D convolutional neuronal network [6]. In this work a modified approach is used to improve the performance of the spatial branch. Thus not only one frame is processed, but n frames are utilized to perform action recognition in the spatial branch. The Features of n I-frames are extracted with the VGGNet-16 Network. The VGGNet-16 architecture takes 224×224 RGB images as input and is originally trained for large-scale image classification. It consists of 13 convolutional layers with filter sizes of 3×3 and 1×1. 5 max pooling layers are placed between the convolutional layers [15]. The last 3 fully-connected layers of the VGG16 Network are dropped, such that n feature maps with the dimension of $7 \times 7 \times 512$ are returned. The weights are pre-trained on ImageNet and only the weights of the last 3 convolutional layers are trained. Afterwards a pooling operation is applied to get the average of all n feature maps. In the following 2 fully connected layers are added. For the fully connected layer rectified linear units (Relu) are used as activation and a dropout of 0.85 is added after the activation to reduce overfitting. A Softmax layer for classification is placed at the top of the network. As mentioned before is the 3D CNN proposed by Chadha et al. [1] used for the temporal stream. It consists of 5 convolutional layers. The first, the second and the fifth convolutional layers are followed by max pooling layers with a pooling size of $2 \times 2 \times 2$ and strides of $2 \times 2 \times 2$. After the spatiotemporal feature extraction with the convolutional layers 2 fully connected layers and a Softmax layer are added. All convolutional and fully connected layers use the parametric Relu as activation [1]. The best method for merging both streams is evaluated in the following sections.

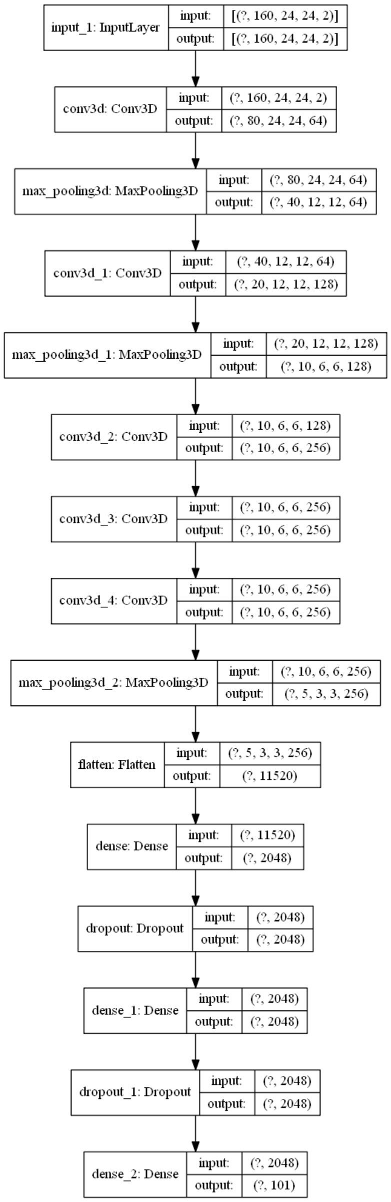

Fig. 4. Architecture of the temporal stream. Implemented after [1].

D. Implementation Details

Both streams are trained separately. Afterwards the streams are fused and fine-tuned such that the best results are achieved. Both networks use the stochastic gradient descent method with momentum 0.9 as optimizer to learn the network parameters. For the temporal stream the learning rate is set to 0.01 and it is reduced after 115 epochs to 0.001. The weights of the temporal stream are initialized with the Glorot normal initializer. A batch size of 32 is used and the network is trained for 230 epochs [1][16]. The spatial stream is trained with a batch size of 16 and a learning rate of 0.001. The training is stopped after 12 epochs without improvement on the validation accuracy. For both inputs data augmentation methods are applied to avoid overfitting. For the temporal stream the proposed methods from Chadha et. al [1] are used. At first the motion vectors are randomly cropped and resized to the expected input size of $24{\times}24{\times}2$. Additionally, they are randomly horizontally flipped and normalized afterwards. The stacked motion vectors start at a random frame and if there are not enough, they are repeated until there are 160 motion vectors. For the training process of the spatial stream all RGB frames of the videos can be picked to have more training data and such that reduce overfitting, while for testing only I-frames are used. Thus randomly 4 frames are picked from the video. The I-frames are cropped and resized to the input size of $224 \times 224 \times 3$ for preprocessing. Additionally the mean RGB value is subtracted [15]. For training of the fusion layers and fine tuning of both streams the same optimizer and learning rate as for the spatial stream is used. The loss function for all models is the categorical cross-entropy. All models are implemented and trained with Keras [17].

E. Results

Both streams are trained separately and fused afterwards. At first the performance of the single streams is evaluated and afterwards the overall model performance in combination with the number of parameters. All models are evaluated on the first test split provided by the UCF101 dataset [14]. After 234 epochs of training the temporal stream achieved an accuracy of 40.09%. Compared to the performance of the proposed implementation by Chadha et al. [1] (77.5%) drops this one almost by half less of, although no differences are recognisable. It would be interesting to clarify the cause of this in the course of further investigations, but this is beyond the scope of this work. After training the spatial stream for 29 epochs, it achieved an accuracy of 76.4%. This result is already better than some pixel-based approaches [9].

For the fusion of both streams only late fusion methods are investigated as this fits the best with the architecture of both streams. Therefore the streams are connected at different fully connected layers and an additional layer is added on top. The complete model is trained again on the first training split of the UCF101 dataset, while only the added fully connected layer and the layer before are allowed to train their weights. All methods, the resulting performances and the overall number of parameters is listed in III-E.

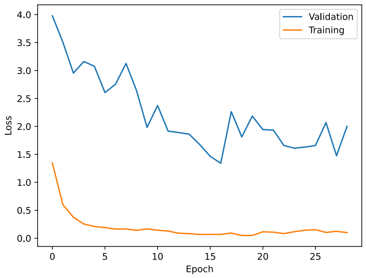

Fig. 5. Evolution of the loss of the spatial stream.

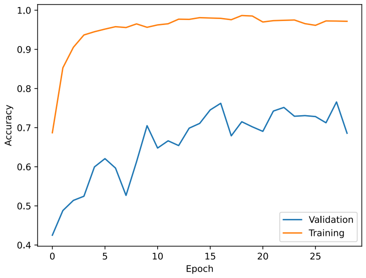

Fig. 6. Evolution of the accuracy of the spatial stream.

Method	Accuracy (%)
Averaging (No Fine-tuning)	74.70
Softmax layer on top + Fine-tune top layers	74.87
Softmax layer after last FC layers + Fine-tune top layers	77.44
Softmax layer after first FC layer + Fine-tune top layer	76.05

TABLE I. COMPARISON OF THE TEMPORAL STREAM ACCURACY

IV. DISCUSSION

A new method for using components of compressed videos for action recognition is presented. For comparison to the presented methods using pixel-based representation for videos the proposed method shows an acceptable result for the combination of accuracy and complexity. It has been shown that with a smaller input size, respectable results still can be achieved. Thus, only I3D [9] has an advantage in the comparison of the accuracy and the parameters. Comparing the proposed approach with other approaches using compressed videos, it can be seen that they either manage to achieve the same accuracy with fewer parameters or show a much better performance [1][3]. All over it is to say that the proposed model is not significant less complex than pixel-based approaches. The main reason for this is the spatial stream as it is really close to standard pixel-based architectures. However, therefore the advantage is that only the I-frames need to be decoded and not the complete video. Furthermore, they are needed to generally provide information about the scene, the objects and so on. In addition, the temporal stream influences the performance in a bad manner as it performs more worse than expected. Actually, the motion vectors should be capable to achieve much better results and even more should it be possible to model long term relations in videos as their size per frame is limited and so more frames can be stacked without excessive increase of complexity. Thus they are not able to compete with the optical flow, although they are 'on the fly' available.

Method	#Parameter	Input Size	Accuracy (%)
3D-ConvNet [8]	79M	16 RGB	82.3
Deep Networks [7]	230M[2]	50 RGB	65.4
Two-Stream [6]	35M[2]	1 RGB, 10 flow	87.0 (split1)
I3D [9]	25M	250 RGB	84.5
Coviar [3][18]	83.6	25 Frames (I-or P-frame)	90.8
MVCNN [1]	29.4M	160 MV[3]	77.5
DMC-Net [13][19]	83.6	25 MV[4]	90.9
Proposed	60.2M	4 RGB 160 MV[3]	77.44 (split1)

TABLE II. COMPARISON OF THE TEMPORAL STREAM ACCURACY

V. CONCLUSION

In this paper, I have investigated the possibility to deploy components of compressed videos to neuronal networks to reduce the overall complexity. Therefore an architecture, originally proposed for decompressed videos, is transferred to be used with compressed videos and the performance is evaluated on the action recognition dataset UCF101. The proposed model does not perform as presumed as the part for temporal modeling did not achieve expectable results. The reason for this could not determined.

REFERENCES

[1] A. Chadha, A. Abbas, and Y. Andreopoulos, "Compressed-domain video classification with deep neural networks: there's way too much information to decode the matrix," *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 1832–1836, 2017.

[2] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: recent updates, datasets, challenges, and applications," *Artificial Intelligence Review*, pp. 1–64, 2020.

[3] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl, "Compressed video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6026–6035.

[4] R. Babu, M. Tom, and P. Wadekar, "A survey on compressed domain video analysis techniques," *Multimedia Tools and Applications*, vol. 75, pp. 1043–1078, 01 2016.

[5] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha, and M. Li, "A comprehensive study of deep video action recognition," *arXiv preprint arXiv:2012.06567*, 2020.

[6] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *arXiv preprint arXiv:1406.2199*, 2014.

[2] Reproduced

[3] Reduced input size compared to RGB frames

[4] Upsampled to input size of 224 × 224 × 2

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[8] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[9] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[10] I. Rodrguez-Moreno, J. M. Martinez-Otzeta, B. Sierra, I. Rodriguez Rodriguez, and E. Jauregi Iztueta, “Video activity recognition: State-of-the-art,” *Sensors*, vol. 19, p. 3160, 2019.

[11] T. Strutz, *Bilddatenkompression*. Wiesbaden, Germany: Vieweg+Teubner Verlag, 2009.

[12] S. Wang, H. Lu, P. A. Dmitriev, and Z. Deng, “Fast object detection in compressed video,” *CoRR*, 2018.

[13] Z. Shou, X. Lin, Y. Kalantidis, L. Sevilla-Lara, M. Rohrbach, S. Chang, and Z. Yan, “Dmc-net: Generating discriminative motion cues for fast compressed video action recognition,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1268–1277.

[14] K. Soomro, A. Zamir, and M. Shah, “Ucf101: A dataset of 101 human actions classes from videos in the wild,” *CoRR*, 2012.

[15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015.

[16] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9, 2010, pp. 249–256.

[17] F. Chollet *et al.*, “Keras,” https://keras.io, 2015.

[18] B. Battash, H. Barad, H. Tang, and A. Bleiweiss, “Mimic the raw domain: Accelerating action recognition in the compressed domain,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 684–685.

[19] Y. Huo, X. Xu, Y. Lu, Y. Niu, M. Ding, Z. Lu, T. Xiang, and J.-r. Wen, “Lightweight action recognition in compressed videos,” in *European Conference on Computer Vision*. Springer, 2020, pp. 337–352.

[20] J. Deng, L. Wang, S. Pu, and C. Zhuo, “Spatio-temporal deformable convolution for compressed video quality enhancement,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 10 696–10 703, 2020.

[21] R. Yang, M. Xu, Z. Wang, and T. Li, “Multi-frame quality enhancement for compressed video,” *CoRR*, vol. abs/1803.04680, 2018.

[22] V. Kantorov and I. Laptev, “Efficient feature extraction, encoding and classification for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2593–2600.

[23] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, “Real-time action recognition with enhanced motion vector cnns,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2718–2726.

[24] Z. Shou, X. Lin, Y. Kalantidis, L. Sevilla-Lara, M. Rohrbach, S.-F. Chang, and Z. Yan, “Dmc-net: Generating discriminative motion cues for fast compressed video action recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1268–1277.