ON THE ROBUSTNESS OF ACTION RECOGNITION METHODS IN COMPRESSED AND PIXEL DOMAIN

Vignesh Srinivasan[1,2], *Serhan Gül*[1], *Sebastian Bosse*[1], *Jan Timo Meyer*[1], *Thomas Schierl*[1], *Cornelius Hellge*[1], and *Wojciech Samek*[1,2]

[1] Dept. of Video Coding & Analytics, Fraunhofer Heinrich Hertz Institute, Berlin, Germany
[2] Berlin Big Data Center, Berlin, Germany

ABSTRACT

This paper investigates the robustness of two state-of-the-art action recognition algorithms: a pixel domain approach based on 3D convolutional neural networks (*C3D*) and a compressed domain approach requiring only partial decoding of the video, based on feature description using motion vectors and Fisher vector encoding (*MV-FV*). We study the robustness of the two algorithms against: (i) quality variations, (ii) changes in video encoding scheme, (iii) changes in resolutions. Experiments are performed on the HMDB51 dataset. Our main findings are that *C3D* is robust to variations of these parameters while the *MV-FV* is very sensitive. Hence, we consider *C3D* as a baseline method for our analysis. We also analyze the reasons behind these different behaviors and discuss their practical implications.

Index Terms— Compressed domain analysis, human action recognition, convolutional neural networks, fisher vector encoding, robust classification

1. INTRODUCTION

Videos have become an integral part of our day to day lives. Today videos already comprise the majority of consumer internet traffic. By 2019 it is estimated to make up to 80% of all uploads and downloads which corresponds to 1.6 zettabytes[1] per year [1]. The transmission and storage of such huge amount of video data was only made possible through the use of steadily improving video compression algorithms such as specified by the currently most used H.264 coding standard [2] or its successor H.265 [3]. Video coding standards only specify the decoding process to ensure interoperability. Therefore, video sequences on the internet differ in a multitude of parameters, mainly influenced by the choice of the codec, the encoder implementation, the target bitrates, noise level, and so on.

In the field of machine learning and computer vision, videos are often processed and analyzed to extract useful and relevant information. Possible applications include face detection and identification, object recognition, video retrieval, and video copy detection. Also action recognition, the topic of this paper, is a very popular task which involves analyzing the videos to identify the performed actions. It has several applications including activity recognition, human-computer interaction, and video surveillance.

Traditionally, action recognition algorithms perform a *pixel domain* analysis, i.e., take a set of video frames as input and annotate the video with predefined action categories (e.g. walking, kissing, swimming etc.). Popular pixel domain approaches are bag-of-words models [4] and deep learning methods [5, 6, 7, 8]. A practical disadvantage of these approaches is that they require a full decoding of the video before starting the analysis which increases the computation time and memory requirements. Recently, action recognition in *compressed domain* have proven to be a faster alternative to pixel domain approaches [9, 10]. This is due to the fact that the compressed domain features can be extracted by only the partial decoding of the video. In terms of performance, the state-of-the-art compressed domain algorithms presented in [9, 10] have slightly worse recognition accuracy compared to the state-of-the-art pixel domain approaches [7, 8, 11].

Typically, action recognition algorithms are studied in very controlled settings where training and testing videos come from the same source and have similar quality. However, the quality of online videos varies due to differences in compression ratio and encoding scheme. From a practical point of view the best performing action recognition method is useless if it is unable to cope with this variability. In this paper, we study the robustness against variations of quality, resolution and encoding scheme, of state-of-the-art action recognition algorithm in the compressed domain approach. A pixel domain method is considered as a baseline performer since it is robust to variations of these parameters.

This paper is organized as follows. Section 2 introduces

*This work was supported by the German Ministry for Education and Research as Berlin Big Data Center BBDC (01IS14013A) and by the German Ministry for Economic Affairs and Energy as Celtic-Plus Project VIRTUOSE. This publication only reflects the authors' views. Funding agencies are not liable for any use that may be made of the information contained herein.

[1] 1 zettabyte $\widehat{=}$ 10^{21} bytes

the two action recognition algorithms used in the paper. Section 3 investigates the robustness of both methods on the HMDB51 dataset. We conclude this work with a discussion and an outlook in Section 4.

2. ACTION RECOGNITION

Compressed Domain *MV-FV*: Hand-crafted features like histograms of oriented flow (HOF), motion boundary histograms (MBH) in combination with histogram of oriented gradients have shown state-of-the-art results for action recognition using optical flow [11]. The compressed domain action recognition method described in [10] requires only a partial decoding of the bit stream to extract the motion vectors as shown in Fig. 1. It then uses these coarse motion vectors to construct HOF and MBH features [10].

To compute the local hand-crafted features, histograms of motion vectors from a $32 \times 32 \times 5$ spatio-temporal cube are considered. The histograms of both HOF and MBH consist of eight motion bins for different orientations and one no-motion bin. The local descriptors for a video are then obtained by stacking the histograms over all the cubes over all the time slices. Fig. 2a displays sample frames from videos encoded using x265 with quantization parameters (QP) 0, 30, and 50, respectively. Fig. 2b visualizes the motion vector flows of the respective frames by color encoding the motion vectors according to their orientation and magnitude. Fig. 2c illustrates the computation of the HOF features using the motion vector flow.

Fisher vectors have been widely used as a robust global descriptor [12]. To obtain Fisher vector representations from local descriptors, Gaussian Mixture Models (GMM) are learned. Consider a *D*-dimensional feature $I = (x_1, x_2, \ldots, x_D)$ extracted from a video and the parameters of the GMM to be $\theta = (\mu_k, \Sigma_k, \pi_k; k = 1, 2, \ldots, K)$. The GMM then associates each local descriptor to a mode *k* in the mixture with a strength given by a posterior probability. The Fisher vectors are then computed by stacking the mean and covariance vectors of all the modes in the mixtures [12]. In *MV-FV* method, Fisher vectors are computed for the HOG and MBH features respectively. The computed Fisher vectors are then aggregated to obtain a fixed length representation for each video. A linear SVM is used to infer the action.

Pixel Domain *C3D*: The 3D Convolutional Neural Network based method in [8] is a pixel domain method (we refer it to as *C3D*). Since the application of 2D convolution on videos has been shown to result in the loss of temporal information [5, 7], *C3D* implements 3D convolution and 3D pooling operations to mitigate this issue [8]. *C3D* operates on 16 consecutive frames from a video at a time. Using deconvolution method described in [13], it was found that in the initial frames, *C3D* learns the appearance of the objects while the motion of the learned object is tracked in the following frames. Thus, *C3D* takes into account both the appearance and the motion of an object [8].

3. EVALUATION

3.1. Dataset And Setup

Evaluation was performed on the HMDB51 dataset, which contains 6766 videos. HMDB51 contains 51 different action categories each having around 100 videos [14]. The sequence length of each video is about 5 seconds. Video sequences focus on the particular actions only and do not contain multiple actions. The dataset was split into training (3570 videos), validation (1666 videos), and test (1530 videos) set.

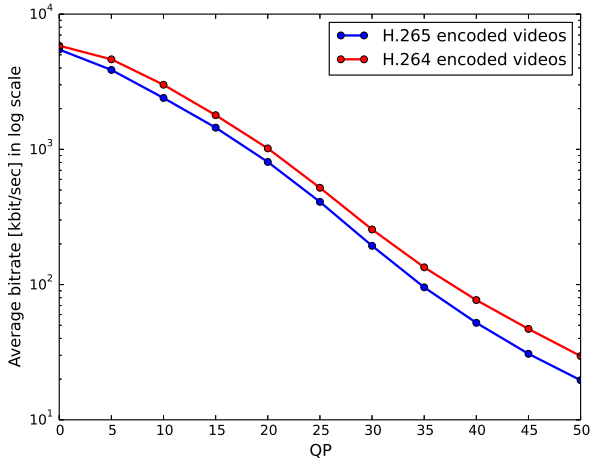


Fig. 3: Average of bitrates for the videos from HMDB51 dataset transcoded to H.264 and H.265 for QPs 0-50.

Videos in HMDB51 were collected from various sources and originally encoded with MPEG-4 Part 2 / DivX. For our analysis, videos were transcoded to H.264 [2] and H.265 [3] using the x264 and x265 implementations in FFmpeg. For both H.264 and H.265, an IPPP coding GOP structure was adopted with a single I frame and no B frames. The operational point of the encoder defining the quality was set by the Quantization Parameter (QP). Fig. 3 shows the bitrates of the videos encoded with x264 and x265 for QPs from 0 to 50, averaged over the whole dataset. On average, the videos used in our experiments have similar range of bitrates for x264 and x265.

MV-FV Setup: Unlike [10], which first samples the motion vector flow with a coarse 16×16 pixel spatial resolution and then uses bilinear interpolation to increase the resolution of the flow field, we created an 8×8 motion vector field by considering the codec-specific inter-prediction features. For H.264, we sampled the motion vectors of 16×8, 8×16, and 8×8 macroblock partitions. For H.265, we sampled the motion vectors from each prediction unit. In order to simplify

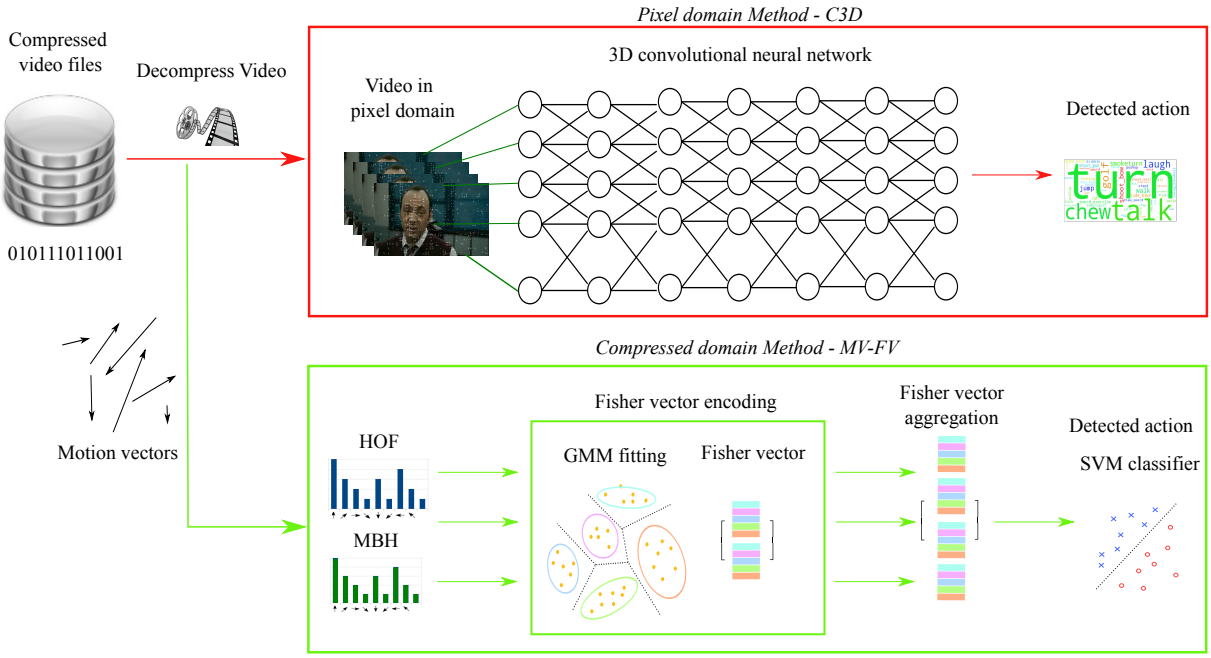
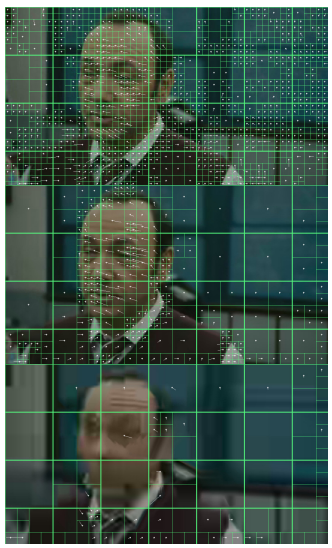
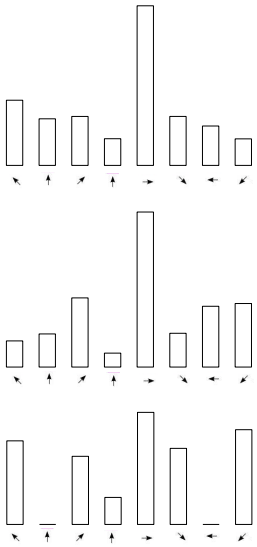


Fig. 1: Overview of pixel and compressed domain methods. Since the compressed domain method requires only the partial decoding of the video stream, it is much faster compared to the pixel domain method. In terms of accuracy, state-of-the-art pixel domain methods perform better.



(a) Visualization of prediction units and their respective motion vectors for QPs 0, 30, and 50. At lower QP levels, the encoder tends to split the frame into smaller prediction units, hence the motion vector flow is more fine-granular. Circles mark the beginning point of the motion vectors.

(b) Visualization of the motion vector flow using a color wheel. Orientation and magnitude of the motion vectors are coded by different hue and saturation levels, respectively. Darker tones of a color imply motion vectors with greater magnitude.



(c) The motion vectors from a $32 \times 32 \times 5$ cube are discretized into histogram bins based on their direction and magnitude.

Fig. 2: Visualization of the motion vectors and coding block structure of a single frame in one sample video from HMDB51 dataset transcoded to H.265 at QPs $0, 30$, and 50.

data processing, motion vectors from both the codecs were arranged in an uniform 8×8 grid. For the block sizes larger than 8×8, motion vectors were duplicated to provide uniformity. For example, in the case of H.264, motion vectors in 16×16, 8×16, and 16×8 blocks were duplicated to assign a motion vector to each 8×8 block.

C3D Setup: To train a *C3D* model, the training set of videos with the best quality (QP=0) was chosen for H.264 as well as H.265. Models were computed by finetuning the *C3D* network over the pretrained model of the *Sports-1M* dataset [7]. In the test phase, the network is given an input of 16 frames stacked together. The predictions from the softmax layer over the various inputs for a video were averaged to obtain one inferred action for the video.

3.2. Experiments

This work investigates the robustness of the classification accuracy of *MV-FV* for videos encoded with H.264. For the analysis, experiments were performed under different scenarios. In [15], it is argued that the accuracy of compressed domain algorithms suffer at the extremes of the QP range (0 or 50). At the higher extreme of the QP range, accuracy of the motion vectors decrease due to degradations in video quality and the obtained features are not very accurate. At the lower extreme, the decrease in accuracy might be caused by the increasing proportion of intra-coded blocks in P frames for which no motion vectors are available. Hence, for the compressed domain algorithm MV-FV, we avoided using training videos encoded at QP 0 and instead used training videos encoded at QP 5.

Same Encoder "SE" Test: Compression of videos lead to a reduced size of the video at the cost of quality. In this experiment, we analyze the effect of decrease in quality on the performance of the action recognition methods using the HMDB51 dataset transcoded to H.265.

a **Train**: H.265 encoded videos at QP = 5 for *MV-FV*.
Test: H.265 encoded videos at QP = $0, 5, ..., 50$.

b **Train**: H.265 encoded videos at QP = $0, 5, ..., 50$.
Test: H.265 encoded videos at QP = $0, 5, ..., 50$ resp.

Cross Encoder "CE" Test: The main aim of this test is to evaluate the robustness of *MV-FV* when trained using videos encoded with one codec and tested with videos encoded with another codec. This might be particularly useful in a large-scale video analysis scenario in which training and test videos might not necessarily be encoded with the same codec.

Train: H.265 encoded videos at QP = 5.
Test: H.264 encoded videos at QP = $0, 5, .., 50$.

Resolution "RE" Test: In an actual scenario, test videos can have different resolutions in addition to variations in quality and encoding schemes. This test is performed to test the robustness of a trained model to given dataset with respect to variations in its resolutions. Towards this goal, the resolution of the videos are changed to 360p, 720p and 1080p at QP 25.

Train: H.264 encoded videos at resolution = 360p.

Test: H.264 encoded videos at resolution = 360p, 720p, 1080p.

The mean average precision is computed by averaging over the recognition accuracy over all the test videos.

3.3. Analysis

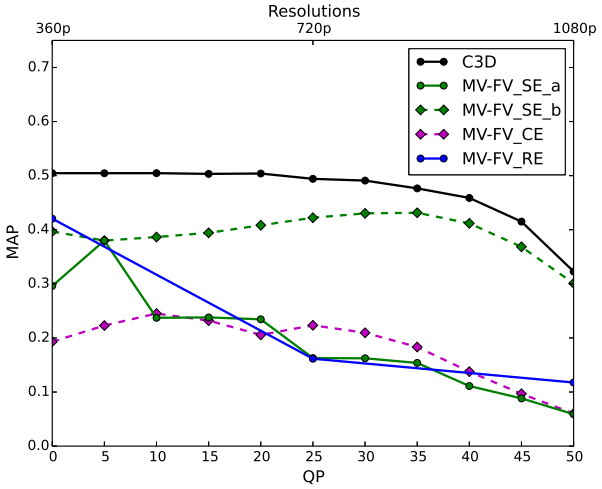
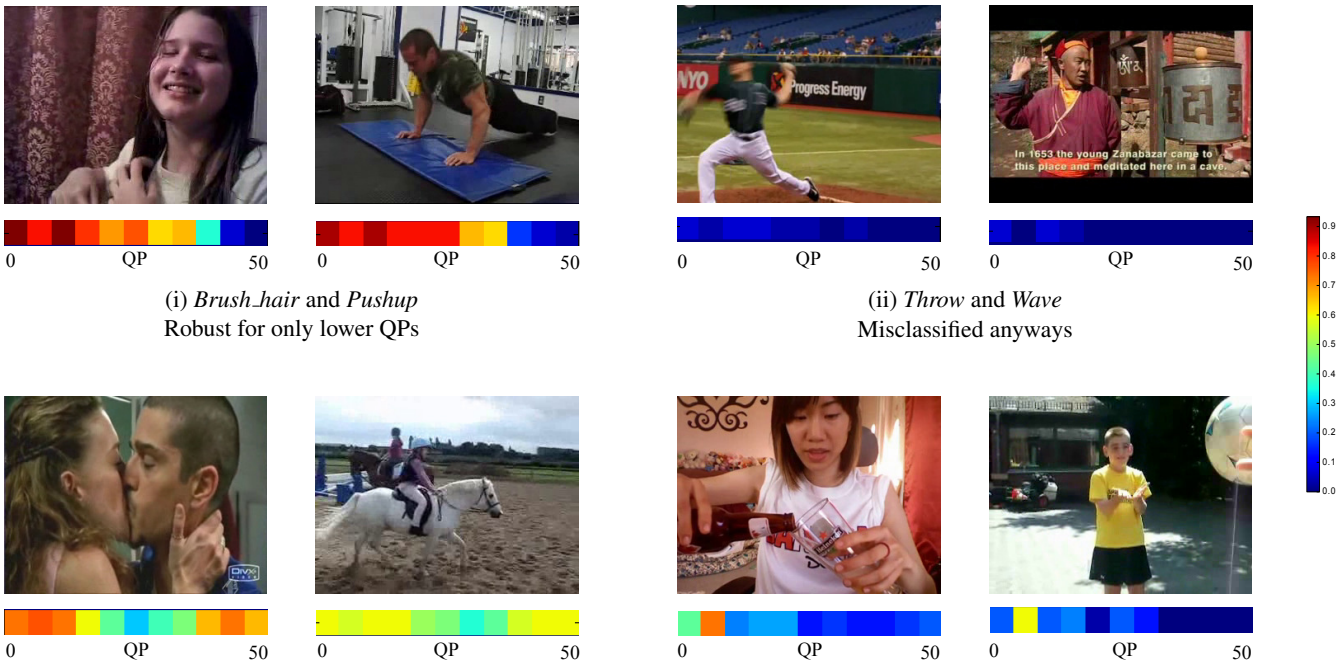


Fig. 4: Comparison of mean average precision (MAP) obtained for the compressed domain (*MV-FV*) for the SE and CE tests as described in Section 3.2. The MAP for the pixel domain (*C3D*) is given as a baseline.

C3D was found to perform similar in all the tests described below. Hence, the results of test SE (a) are displayed as a baseline for comparison.

SE Test: Varying QP has a strong effect on the motion vectors, as illustrated in Fig. 2c. QP affects the rate-distortion optimization performed in the encoder and thus changes the decision of encoder in the motion estimation process. Hence, motion vectors have a different distribution for each QP. This varying distribution of the motion vectors affects the accuracy of the system, if training videos and test videos are encoded with different QPs, making the compressed domain *MV-FV* method much less robust to changes in the compression ratio than the *C3D* method.

We observe in Fig. 4 that a classifier, which was trained with motion vectors from videos encoded with a particular QP, experiences a drop in accuracy when tested with videos encoded with a different QP. But, when they are trained and tested on the same QP, as in SE tests (b), they tend to have



(i) *Brush_hair* and *Pushup*
Robust for only lower QPs

(ii) *Throw* and *Wave*
Misclassified anyways

(iii) *Kiss* and *Ride_horse*
Robust even at higher QPs

(iv) *Pour* and *Catch*
Overfitting

Fig. 5: Average precision (AP) computed for classes encoded with H.265 for the SE test (a) over QPs. Given are sample frames from each of the classes. AP score of 0% is represented by the color *blue* while a score of 100% is given by color *red* as displayed by the colorbar on the right.

higher accuracy. This is a classic case of a classifier overfitting to a particular distribution of the motion vectors. However, in practice, it is not feasible to train a model multiple times over a range of QPs. Hence, robustness of *MV-FV* is quite sensitive to the particular QP it is trained on.

Results of SE tests (a) shown in Fig. 4 (denoted as *MV_FV_SE_a*) display the lack of robustness of *MV-FV* against quality variations. A significant decrease in MAP is observed due to the more complex motion estimation structure of H.265, which makes the obtained local histogram features overfit for the quality level on which they were trained.

CE Test: *MV-FV* performs significantly worse when training and test videos are not encoded with the same codec, as observed in Fig. 4. Since *MV-FV* operates on motion vectors, a model which was trained using the motion vectors produced by the H.265 encoder, performs worse when tested using H.264 encoded videos. Enhanced motion estimation and motion vector prediction techniques in H.265 lead to generation of a significantly different set of motion vectors (compared to H.264) on the same video content at a similar quality. This observation can also be explained via the overfitting phenomenon, as mentioned above for the SE test. Hence, the reduction in accuracy.

RE Test: Motion vectors are also very sensitive to the resolution of videos. The distribution of the motion vectors of a video in a different resolution can be quite different from the distribution of motion vectors of the video in its original resolution. The main take-away from this test is that the *MV-FV* trained on a particular resolution also tends to overfit to motion vectors from that resolution. Here, the model trained on videos at 360p resolution were found to perform well when tested with videos at the same resolution. When this model was tested with videos at 720p and 1080p resolution, then it performed significantly worse.

Fig. 5 displays the average precision (AP) of various classes and the behavior they exhibit under SE test. AP gives the percentage of videos correctly classified for a given class. Some classes like *Throw* and *Wave* have very low average precision even for the best quality of the video. In the Fig. 5i, classes like *Brush_hair* and *Pushup* are quite robust. They start to lose their accuracy significantly after QP 40.

Kiss and *Ride_horse* are a set of classes which are observed to be have considerably high AP at all QP in Fig. 5iii. Although the videos in these classes are accurately classified, their constant high AP can also be attributed to higher recall values. When videos are compressed with a high QP, the features belonging to classes like *Kiss*, *Talk* and *Turn*, which focus on the face of the person in the video, could be similar. And hence, at higher QPs, *Talk* and *Turn* are misclassified

as *Kiss*, while *Cartwheel* and *Ride_bike* are misclassified as *Ride_horse*.

Overfitting behavior is observed in classes *Catch* and *Pour* in the Fig. 5iv. They show a high AP for the videos with same QP, while performing poorly against videos with all other QPs. We also find that classes that are observed to overfit tend to have localized action in general like catching and pouring − movement of hands alone.

4. DISCUSSION AND FUTURE WORK

In this paper, we evaluated the robustness of action recognition algorithms in compressed domain against compression, changes in encoding scheme and changes in resolution. The action recognition performance of the compressed domain algorithm (*MV-FV*) fluctuates significantly with compression ratio variations. Different encoding schemes for training and testing leads to a significant decrease in accuracy. The same is true for variations in resolutions. As a benchmark, we compared our results to a state-of-the-art pixel domain action recognition algorithm *C3D*.

Supervised learning algorithms for human action recognition require immense amounts of data for satisfactory recognition performance. As the popularity of applications such as adaptive video streaming has been increasing, it is becoming more and more important to ensure the robustness of the action recognition algorithms against compression artifacts of different compression levels and resolutions. In some use cases, test videos might have a significantly degraded quality compared to the original videos on which the action recognition algorithm was trained. For this reason, it is very important to understand the effects of video compression and transmission errors on existing video analysis technologies.

In future work, we will investigate the advantages and limitations of compressed and pixel domain methods by analyzing *what* exactly classifiers rely on in order to distinguish between human actions [16] and by comparing different models [17]. Furthermore, we plan to incorporate additional features (such as DCT coefficients and macroblock types) into compressed domain action recognition algorithms and increase the robustness of the algorithms against compression and noise. There is scope for plenty of future work to make the features more robust and transferable to ensure the robustness of video analysis applications, such as human action recognition.

References

[1] I Cisco, "Cisco visual networking index: Forecast and methodology, 2011–2016," *CISCO White paper*, pp. 2011–2016, 2012.

[2] T. Wiegand, G. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[3] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[4] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. CVPR*, 2008, pp. 1–8.

[5] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Adv. in NIPS*, 2014, pp. 568–576.

[6] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE TPAMI*, vol. 35, no. 1, pp. 221–231, 2013.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. CVPR*, 2014, pp. 1725–1732.

[8] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: generic features for video analysis," *CoRR*, vol. abs/1412.0767, 2014.

[9] R. V. Babu, M. Tom, and P. Wadekar, "A survey on compressed domain video analysis techniques," *Multimed. Tools Appl.*, pp. 1–36, 2014.

[10] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding, and classification for action recognition," in *Proc. CVPR*, 2014, pp. 2593–2600.

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

[12] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Computer Vision–ECCV 2010*, pp. 143–156. Springer, 2010.

[13] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer vision–ECCV 2014*, pp. 818–833. Springer, 2014.

[14] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proc. ICCV*, 2011.

[15] S. H. Khatoonabadi, I. V. Bajić, and Y. Shan, "Compressed-domain correlates of fixations in video," in *Proc. Int. Workshop on Perception Inspired Video Processing*, 2014, pp. 3–8.

[16] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, vol. 10, no. 7, pp. e0130140, 2015.

[17] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek, "Analyzing classifiers: Fisher vectors and deep neural networks," in *Proc. CVPR*, 2016, pp. 2912–2920.