Human Action Recogntion using Compressed Videos in H.264 Format

Khalil Smaoui
Chair for Data Processing, Technical University of Munich
khalil.smaoui@tum.de

***Abstract*—Human Action Recognition (HAR) is gaining more and more attention in the field of Computer Vision. As it seeks to comprehend the human behaviour, analyze it and label it to an action, HAR is used in various domains such as video surveillance systems, smart homes, and hospital environments. Commonly, the existing deep learning approaches consist of Convolutional Neural Networks that are capable to learn robust representations of image data by processing RGB pixels. However, some recent works proposed the usage of compressed video data in their networks as an alternative to reduce the complexity of the networks caused by the high redundancy between frames. In this paper, a deep neural network is implemented that is capable to learn from the Discrete Cosine Transform (DCT) coefficients of I-frames from compressed H.264 data. The results show that the developed network is not performing as well as expected and still need to be reviewed.**

Keywords*—Human Action Recognition, Compressed Data, H.264, Convolutional Neural Network, Deep Learning

I. Introduction

With the increasing availability of video content and rapidly developing computational power, different computer vision tasks on videos have also become available to the research community. Among these computer vision tasks, Human Action Recognition has gained a lot of attention in the last decades. Numerous deep learning methods for human action recognition have been proposed. Historically, deep learning methods such as convolutional neural networks and recurrent neural networks have shown remarkable performance and even achieve state-of-the-art results by automatically learning features from the raw pixel-based images [1–3]. However, such methods have limitations since they process video information as RGB sequences which is usually redundant [4]. Moreover, most of the transmitted or stored video data available nowadays are represented in compressed format. Therefore, each video must first be decoded into RGB image before being fed to the network. To alleviate all these limitations, recent research has shown that the HAR task can also be performed using compressed video data [5–8]. Therefore the following research questions are examined during this work:

- Is it possible to achieve similar results as the state-of-the-art networks in the task of HAR using only the frequency-domain information of the intra-coded frames (I-frames) encoded in H.264 format?
- How should the H.264 DCT coefficients of I-frames be processed?

To answer these questions, entropy decoded and parsed compressed video data is straightly used for the sake of performing the Human Action Recognition task. The features that are extracted from the compressed data are restricted to the Discrete Cosine Transform (DCT) coefficients of the I-frames (also known as keyframes). The experiments in this work were realized on the UCF-50 dataset [9]. The performance and complexity of the proposed method were observed, discussed, and compared to the state-of-the-art methods of video recognition in the compressed domain.

II. State Of The Art

A. Video Compression

The main goal of video compression is to reduce the spatio-temporal redundancies by applying various image transforms and motion compensation [4].

1) Video Coding Format: Most of the modern compression standards divide video data into three major picture types: intra-coded (I-frames), predicted (P-frames), and bi-directionally predicted (B-frames). As shown in Figure 3, a Group of Pictures (GOP) starts with an I-frame followed by P-frames and/or B-frames [4]. An I-frame or a keyframe or an intra-frame consists only of macroblocks that use intra-prediction. As shown in Figure 1, every macroblock in an I-frame is allowed to refer to other macroblocks only within the same frame. A P-frame allows macroblocks to be compressed using temporal prediction in addition to spatial prediction. For motion estimation, P-frames use frames that have been previously encoded. A B-frame may be viewed as a special P-frame. It is a bi-directional frame that can refer to frames that occur both before and after it. B-frames are used in codecs that use macroblock-based compression such as H.264/AVC [10] and HEVC [11]. An example that illustrates the inter-prediction process is shown in Figure 2. Thus, images from B-frames and P-frames are stored in a compressed format and are reconstructed using the encoded offsets, namely motion vectors (MV) and residuals (R). The syntax elements of the compression standard such as frame number, frame type (I, P, or B), the positions and motion vectors of inter-coded macroblocks, the DCT coefficients of intra-coded frames and residuals can be obtained by parsing and entropy (Huffman) decoding video bitstreams. These operations take less than 20% of the computational load in the full video decoding process [12].

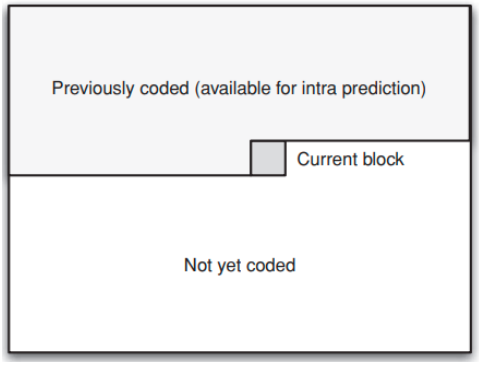

Fig. 1: INTRA-PREDICTION [4]

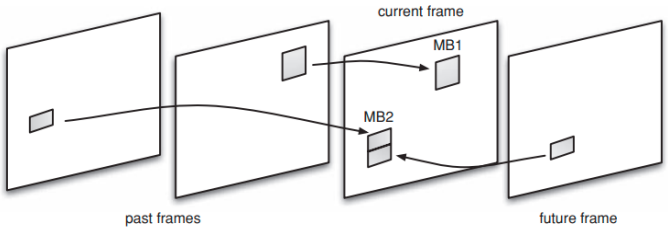

Fig. 2: INTER-PREDICTION: MB1 IN THE CURRENT FRAME IS PREDICTED FROM A REGION OF A PREVIOUS FRAME. MB2 IS PREDICTED FROM TWO PREVIOUSLY CODED FRAMES: A PAST FRAME AND A FUTURE FRAME. [4]

2) DCT Transform: The DCT transform [13] is applied to non-overlapping blocks, generally of size 8×8 and commonly called macroblocks. Each block is projected onto a basis of 64 patterns representing various horizontal, vertical, and composite frequencies. Any block can be fully recovered from the knowledge of its coefficients since the basis is orthogonal [14]. The DCT transformation processes each of the three input channels (Luminance channel Y and two Chrominance channels Cb and Cr) separately.

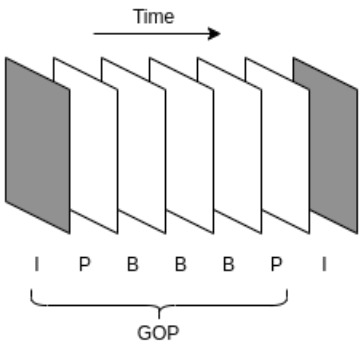

Fig. 3: EXAMPLE OF A GOP.

B. Human Action Recognition

Human activity recognition, or HAR, is a challenging time series classification task. The task of HAR has the potential to aid in different applications such as video surveillance systems [15] and understanding of visual information [16]. Recently, deep learning has been successfully used to learn powerful and interpretable features for recognizing human actions in pixel-based videos and compressed videos.

1) RGB Human Action Recognition:
The majority of the existing action recognition algorithms are implemented using large CNNs [1] [2]. K. Simonyan et al. suggested a two-stream network that uses two CNNs to simulate raw video frames and optical flow, respectively [1]. The Temporal Segment Network (TSN) is one of many enhanced versions of [1] created to capture the long-range temporal structure, although it still depends on the optical flow stream, which is too expensive to compute [3].

2) Compressed Human Action Recognition:
The most modern compression standards use the motion compensation technique that reduces the video data size based on motion estimation from neighboring frames. Recent works profited from this aspect and showed different approaches to acquiring useful information videos in compressed format. The CoViAR method [6] (**Co**mpressed **Vi**deo **A**ction **R**ecognition), which uses MPEG-4 compressed streams, is a multi-stream network composed of three independent CNNs. Each CNN is for one of these three features available in the compressed data:

1) RGB images encoded in I-frames
2) motion vectors
3) residuals encoded in P-frames

In fact, the CoViAR still operates on the pixel-based domain since the frequency domain representation used to encode the pictures in I-frames and the residuals in P-frames needs to be decoded to the spatial domain (RGB pixel values) before being fed to the network. Ultimately, the final prediction is computed via a weighted averaging of the video scores from all three streams. Gueguen et al. [14] approached the problem from another perspective by proposing various architectural modifications to the ResNet-50 network in order to operate directly in the frequency domain. The DCT coefficients are obtained by partial decoding, thus saving the high computational load and memory usage in fully decoding the JPEG images. Recently, Santos et al. [7] proposed the Fast-CoViAR, an extended version of CoViAR, that also operates directly on the frequency domain. The network is a two-stream network that is comprised of two different CNNs:

1) I-frames network using a modified ResNet-50 proposed by Santos et al. [8].
2) Motion vectors from P-frames network using a ResNet-18.

According to Santos et al. [7], the residual network was excluded since it only results in a minor increase in network performance at the cost of a significant increase in computational complexity. The Fast-CoViAR method utilizes a technique called FBS (Frequency Band Selection). The idea is to reduce the network complexity by selecting the DCT coefficients of

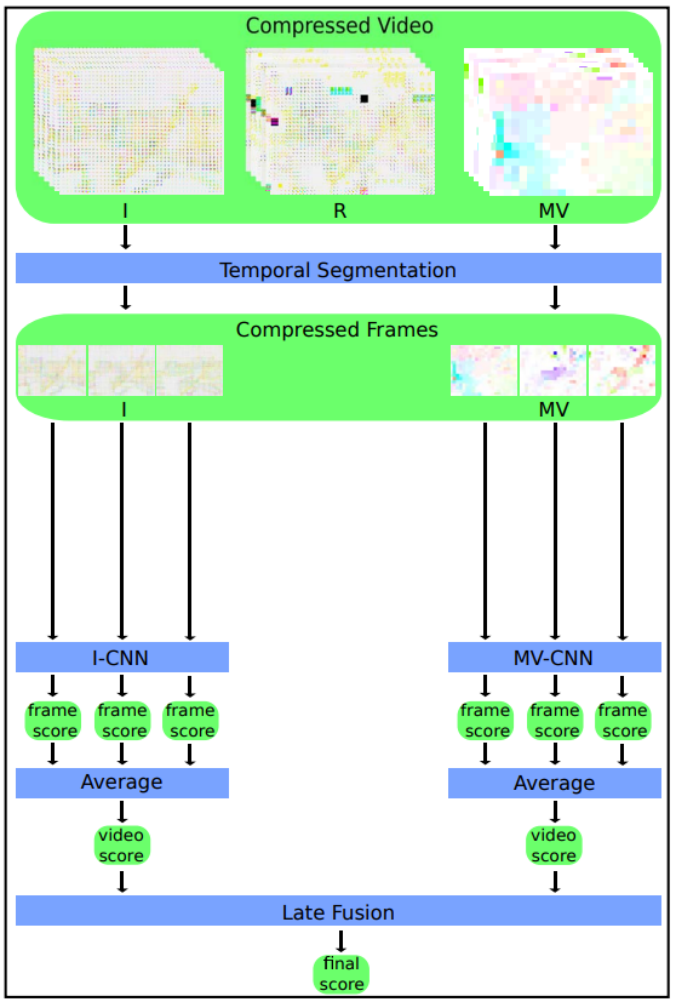

Fig. 4: OVERVIEW OF THE FAST-COVIAR METHOD [7].

the lowest frequencies which are more relevant in terms of visual effect. Similar to CoViAR, the final prediction of the Fast-CoViAR is computed via a weighted average of the video scores from both streams. An illustration of the Fast-CoViAR method can be observed in Figure 4. The Fast-CoViAR also utilizes videos in MPEG-4 format.

III. METHODOLOGY

A. Dataset and Feature Extraction

The UCF-50 dataset is used in this work. The UCF-50 consists of 6676 realistic videos taken from YouTube[1] and contains 50 action classes. It has a more popular dataset called the UCF-101 [17].

To re-encode the UCF-50 dataset according to the H.264/MPEG-4 AVC standard, FFMPEG[2] command-line tools are used. Since the default GOP size is set to 250 frames in the FFMPEG and many videos in the dataset are not long enough, the GOP size is then forcefully set to 50 frames in order to reach a decent number of I-frames for short videos. In Table I, the distribution of the videos in both training and test sets after forcing the GOP to 50 can be observed.

Following the compression, the macroblock information is extracted using the JM 16.1[3] which then creates XML-based trace files that contain information of the encoded bitstream. As mentioned before, only DCT coefficients of I-frames are used in this work since the I-frames record the main information of video data. Therefore, a customized parser is implemented that returns only the DCT coefficients of the I-frames for each video. The parser goes through all macroblocks of each I-frame for all three planes (Luminance Y and Chrominance Cb/Cr). Considering that the two chrominance planes, Cb and Cr are down-sampled horizontally and vertically by a factor of 2 during the JM 16.1 extraction, due to the fact that human vision is more sensitive to brightness details than to color detail, both planes are zero-padded in the frequency domain to match the dimension of the luminance plane. Namely, each frame that has a size of (352, 256, 3) will end up with (352, 256, 3) shaped DCT coefficients.

	I-frames= 1	I-frames= 2	I-frames= 3	I-frames> 3
Train Set(∼70%)	9	657	938	3028
Test Set(∼30%)	2	267	389	1179
Total	11	924	1327	4207

TABLE I: DISTRIBUTION OF VIDEOS BY NUMBER OF I-FRAMES AFTER CHANGING THE GOP SIZE TO 50.

B. Network Input

Inspired by [7] and [3], exactly 3 I-frames of each video is fed into the network. Furthermore, the DCT coefficients are rearranged in a way so that for each color channel, each macroblock is then projected into the 64 orthonormal DCT basis vectors which gives a frame of size (44, 32, 64) given a macroblock size of (8×8) and frame size of (356, 252, 3). In other words, the DCT coefficients of the macroblocks that correspond to the same DCT basis function are arranged together on the same depth dimension while conserving the spatial order. As a result, low-frequency DCT coefficients will end up in the shallow channels along the depth dimension, while high-frequency coefficients will end up in the deep channels [14]. Lastly, DCT coefficients for all three Y, Cb, and Cr planes are concatenated channel-wise. Thus the final dimensions are $I \in \mathbb{R}^{3\times192\times44\times32}$

C. Network Architecture

As a whole, the network is inspired by the Fast-CoViAR network proposed by Santos et al. [7]. Unlike the Fast-CoViAR, the proposed network in this work is a spatial single-stream network that operates only on the DCT coefficients from the

[1] https://www.youtube.com/
[2] https://github.com/FFmpeg/FFmpeg
[3] https://vqeg.github.io/software-tools/encoding/modified-avc-codec/

I-frames. The network corresponds to the improved version of ResNet-50 network [8] that extends the ResNet-50 network proposed by [14]. Additionally, the network uses a Frequency Band Selection (FBS) technique to select the most significant DCT coefficients before feeding them to the network [8]. Since higher frequency information has a less visual effect, only the lowest frequency coefficients are retained. An illustration of the network can be observed in Figure 5.

IV. Experiments and Results

A. Experiments

As mentioned in the previous section, exactly 3 frames are selected for each video. Therefore, 4 different cases were investigated:

1) **Case 1:** Number of I-frames is equal to 1
2) **Case 2:** Number of I-frames is equal to 2
3) **Case 3:** Number of I-frames is equal to 3
4) **Case 4:** Number of I-frames is greater than 3

At this stage, different approaches were tested to select the I-frames for each video.

1) Case 1:

- **Pick the existing I-frame and fill the 2 remaining frames with zeros**
- Pick the existing I-frame and assign the 2 remaining frames with the same values as the existing I-frame

2) Case 2:

- **Pick the 2 existing I-frames and fill the remaining I-frame with zeros**
- Pick the 2 existing I-frames and assign the third frame the values of either frame 1 or 2
- Pick the 2 existing I-frames and assign the third frame the averaged values of frames 1 and 2

3) Case 3:

- **Pick all three existing I-frames**

4) Case 4:

- Randomly pick 3 I-frames
- **Pick the first 3 I-frames**
- Pick the first, the middle, and the last I-frames

However, the best accuracy results are obtained with the points mentioned in bold text. Understanding the cause of this is beyond the scope of this work.

To recapitulate, the learning procedure of the proposed network can be divided into these 4 steps:

1) Parse compressed video and obtain DCT coefficients of the I-frames
2) Pick exactly 3 I-frames.
3) Feed the network with rearranged and concatenated DCT coefficients. (Optionally: apply the FBS technique with $n = 32$ or $n = 16$ for each color channel)
4) Assign a score for each frame and average all scores to give a final score to the whole video.

The experiments were performed on a computer provided by the Chair of Data Processing at the Technical University of Munich equipped with an Intel Core i7-7000K processor and a GeForce RTX 2060 SUPER GPU. The ResNet-50 network is pre-trained on the ImageNet dataset [18]. Adam is used as an optimizer to fine-tune the model with a batch size of 16. Step-decay is implemented that divides the initial learning rate by a factor of 10 after a certain number of epochs. Other hyper-parameters used in the experiments can be observed in Table II.

Hyperparameter	Value
Initial Learning Rate	0.01
Total Number of epochs	255
The step-decay scheduler setting	75, 135, 195

TABLE II: The hyperparameters used in the experiments.

As mentioned in the previous chapter, the UCF-50 dataset is used in the experiments. There is no official train/test split for UCF-50. Since UCF-101 is the extension of UCF50, split 1 from UCF-101 was used after removing the additional 51 categories.

Similar to [7], two data augmentation techniques were applied during the training phase namely, a horizontal flipping with a probability of 50% and random cropping with scale jittering. Afterward, the input size is resized to 28×28. In the end, the size of the input that is fed into the network is (3, 192, 28, 28).

Three different ResNet-50 networks were tested: the standard network that operates on all 64 frequency channels for all three color channels (Y, Cb, and Cr); and the network with the FBS technique that only considers the lowest 32 and 16 frequency channels for each color channel. For FBS with $n = 32$, the input size would be (3, 96, 28, 28), and (3, 48, 28, 28) in case of FBS with $n = 16$.

B. Results

All three models are evaluated on the first test split of the UCF-50 dataset as discussed in the previous section.

Table III shows the classification accuracy and the number of learnable parameters of the tested networks. The best results in terms of performance are obtained from the network that utilizes the FBS technique with $n = 32$ with an accuracy of 57.16 $\%$. In terms of network complexity, the network that utilizes the FBS technique with $n = 16$ is obviously the best network among other networks with a number of parameters equal to 25.6M.

Table IV compares the accuracy of the best-performing version of the proposed network (with FBS (32)) and the state-of-the-art human action recognition networks. Although, since the UCF-50 dataset is used in this work while the UCF-101 dataset is used in nearly all the state-of-the-art models, the comparison is more or less inequitable.

Figure 6 illustrates the confusion matrix of the best-performing version of the proposed network (with FBS (32)) over the 50 classes of the UCF-50 dataset. Almost all of the action classes in the UCF-50 dataset have inner-class variation. The network performed well in the following action

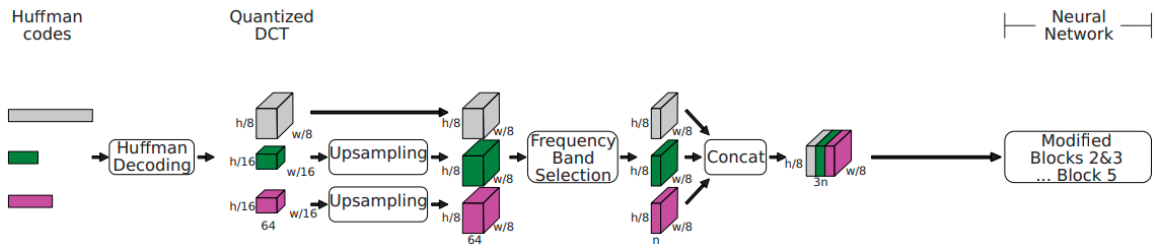

Fig. 5: ILLUSTRATION OF THE NETWORK [8].

classes: *Billiards, and BreastStroke*. By investigating the UCF-50 dataset videos, it is noticed that the videos for each of these classes show similar background scenes. However, the network performance is extremely poor in the following action classes: *Mixing, PullUps, and YoYo.* where the videos for each class contain different background scenes. The lack of motion information should also be a potential reason for the low performance.

Method	Accuracy	Complexity (# of Parameters (10^6))
Standard Network	56.18%	28.4
w/ FBS n = 32	57.16%	26.2
w/ FBS n = 16	52.53%	25.6

TABLE III: THE CLASSIFICATION ACCURACY AND THE COMPLEXITY OF THE THREE DIFFERENT PROPOSED VERSIONS OF THE RESNET-50 NETWORKS. ALL THREE VERSIONS PROCESS DCT COEFFICIENTS OF I-FRAMES FROM THE UCF-50 DATASET.

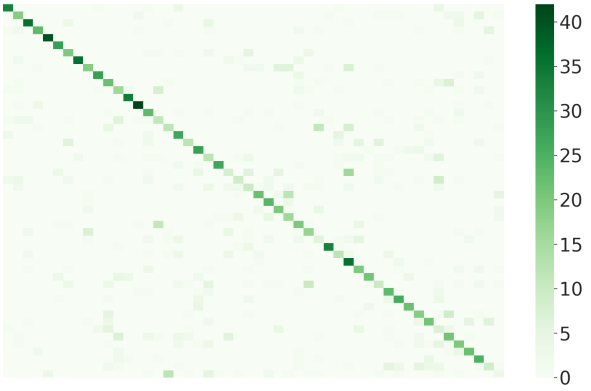

Fig. 6: CONFUSION CHART FOR UCF-50 DATASET.

Method	Accuracy	Complexity (# of Parameters (10^6))
DCT [7]	78.8% (split1 UCF-101)	28.4
DCT w/ FBS (32) [7]	80.9% (split1 UCF-101)	26.2
CoViAR [6]	90.8%	83.6
DMC-Net [19]	90.9%	83.6

TABLE IV: PERFORMANCE AND COMPLEXITY OF SOME STATE-OF-THE-ART NETWORKS THAT ARE USED FOR COMPARISON PURPOSES AGAINST THE PROPOSED NETWORKS.

V. DISCUSSION

Inspired by the Fast-CoViAR method, the model proposed in this work consists of the frequency stream network proposed by Santos et al. [7] that operates only on the DCT coefficients of the I-frames. Therefore, only I-frames are extracted and parsed from the compressed videos. They provide general information about the scene and the objects within the scene. Unlike Fast-CoViAR, the implemented network supports H.264 compressed data instead of MPEG-4 compressed videos. Furthermore, the model in this work is trained on the UCF-50 dataset instead of the UCF-101 dataset. The obtained results in terms of accuracy are mediocre compared to the Fast-CoViAR model and other state-of-the-art models. The complexity of the proposed network which matches the same complexity of the Fast-CoViAR network is significantly less than other state-of-the-art models.

VI. CONCLUSION

In the scope of this paper, a deep neural network is proposed that straightly digests frequency domain information in contrast to the conventional networks that operate on pixel-based information.

The network consists of a single-stream network that learns from the DCT coefficients of the I-frames that are extracted and parsed from compressed videos. As a compression standard, H.264 is used throughout the experiments. The training of the network is performed with the UCF-50 dataset; a sub-dataset of the famous UCF-101 dataset. Three different versions of the network were tested during the experiments. The results show that the best classification accuracy over 50 action classes is ordinary (57.16%) and it is achieved by the network that utilizes the Frequency Band Selection technique with $n = 32$.

In future work, UCF-101 can be used in the experiments for a more fair comparison with the state-of-the-art models.

Furthermore, the techniques that are used to select the I-frames that are fed into the network can be reviewed and investigated. Additionally, a temporal stream that digests MV from predicted frames (P and/ or B frames) can be implemented and investigated.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[2] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[3] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36.

[4] I. E. Richardson, *The H. 264 advanced video compression standard*, 2nd ed. John Wiley & Sons. Hoboken (New Jersey), 2011.

[5] 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"," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 1832–1836.

[6] 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.

[7] S. F. dos Santos and J. Almeida, "Faster and accurate compressed video action recognition straight from the frequency domain," in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2020, pp. 62–68.

[8] S. F. dos Santos, N. Sebe, and J. Almeida, "The good, the bad, and the ugly: Neural networks straight from JPEG," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1896–1900.

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

[10] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[11] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[12] V. Bhaskaran and K. Konstantinides, *Image and video compression standards: algorithms and architectures*, 2nd ed. Springer Science and Business Media, 1997.

[13] S. Liu and A. C. Bovik, "Efficient DCT-domain blind measurement and reduction of blocking artifacts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1139–1149, 2002.

[14] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, "Faster neural networks straight from JPEG," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[15] O. Mishra, P. S. Kavimandan, M. Tripathi, R. Kapoor, and K. Yadav, "Human action recognition using a new hybrid descriptor," in *Advances in VLSI, Communication, and Signal Processing*. Springer, 2021, pp. 527–536.

[16] M. A. Khan, K. Javed, S. A. Khan, T. Saba, U. Habib, J. A. Khan, and A. A. Abbasi, "Human action recognition using fusion of multiview and deep features: an application to video surveillance," *Multimedia tools and applications*, pp. 1–27, 2020.

[17] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[19] 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.