CV-C3D: Action Recognition on Compressed Videos with Convolutional 3D Networks

Samuel Felipe dos Santos[1], Nicu Sebe[2], and Jurandy Almeida[1]

[1]Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo – UNIFESP
12247-014, São José dos Campos, SP – Brazil
Email: {felipe.samuel, jurandy.almeida}@unifesp.br

[2]Dept. of Information Engineering and Computer Science
University of Trento – UniTn
38123, Trento, TN – Italy
Email: niculae.sebe@unitn.it

***Abstract*—Action recognition in videos has gained substantial attention from the computer vision community due to the wide range of possible applications. Recent works have addressed this problem with deep learning methods. The main limitation of existing approaches is their difficulty to learn temporal dynamics due to the high computational load demanded for processing huge amounts of data required to train a model. To overcome this problem, we propose a Compressed Video Convolutional 3D network (CV-C3D). It exploits information from the compressed representation of a video in order to avoid the high computational cost for fully decoding the video stream. The speed up of the computation enables our network to use 3D convolutions for capturing the temporal context efficiently. Our network has the lowest computational complexity among all the compared approaches. Results of our approach in the task of action recognition on two public benchmarks, UCF-101 and HMDB-51, were comparable to the baselines, with the advantage of running at faster inference speed.**

I. Introduction

Over the past decade, the problem of recognizing actions in a video has received considerable attention from the computer vision research community. One of the reasons of such growing interest is due to the comprehensive range of applications, from surveillance, medical, and industrial environments to smart homes [1].

One of the main issues concerning the action recognition problem refers to the extraction of proper representations capable of encoding valuable information from video content [2]. Many prior approaches rely on hand-crafted features, where low-level appearance or motion cues, such as color, texture or optical flow, are computed based on space-time interest points detected in a video, which are then utilized to train classifiers, like support vector machines (SVM) [3]–[6]. These features are designed by hand and usually require high expertise for domain-expert knowledge [7].

Recently, data-driven features have emerged as an alternative to overcome those shortcomings. They are learned directly from the data without the necessity of incorporating any domain knowledge, as in deep learning methods, which aim at learning feature hierarchies, where features from lower levels are composed to form higher level features. Feature learning at different levels of abstraction enables to model complex functions mapping the input data directly to the outputs [8].

A variety of deep learning methods for action recognition can be referred in the literature [2], [7], [9]–[12]. In most of them, a video is parsed frame by frame with convolutional neural networks (CNNs) designed for images [13], [14]. Other methods process videos as image sequences using 2D CNNs, 3D CNNs, or recurrent neural networks (RNNs) [15], [16].

In spite of all the advances, the temporal structure of videos poses some challenges for training deep learning models [17]. First, the computational costs are expensive, as a huge amount of videos is required for training and they need to be decoded during this process. In addition, the number of learnable parameters is high, increasing the complexity of the model and, consequently, the chances of overfitting. These aspects are crucial for the performance of deep learning methods [10].

To address the aforementioned issues, we propose a Compressed Video Convolutional 3D network (CV-C3D). Our approach exploits relevant information pertaining to visual content available in the compressed representation used for video storage and transmission. This enables to save high computational load in full decoding the video stream and therefore greatly speed up the processing time.

We evaluated our approach on two action recognition benchmarks: UCF-101 and HMDB-51. Results point that our network is efficient for it has the lowest computational complexity. For action recognition, our approach performed similar to the other methods on the UCF-101 dataset and achieved the second best performance on the HMDB-51 dataset. Despite CoViAR performs better than CV-C3D in terms of classification accuracy, CV-C3D is one order of magnitude faster than CoViAR for inference.

The remainder of this paper is organized as follows. Section II introduces some basic concepts, like action recognition and video compression. Section III discusses related work.

Section IV describes our CV-C3D network. Section V presents the experimental protocol and the results from the comparison of CV-C3D with other methods. Finally, we offer our conclusions and directions for future work in Section VI.

II. Background

This section presents a brief overview about video action recognition and video compression.

A. Action Recognition

A comprehensive review of methods for action recognition is presented in [2], [7], [9]–[12]. In general, existing solutions are based on a two-step approach: (*i*) extraction and encoding of features, and (*ii*) classification of features into classes [9].

Prior approaches are generally based on hand-crafted features, which are normally built on the pixel-level and carefully designed to deal with challenging issues, such as occlusions and viewpoint changes. They can be grouped into to four categories: (1) spatial-temporal volume-based approaches, (2) skeleton-based approaches, (3) trajectory-based approaches, and (4) global approaches [11]. Even though these approaches may achieve high performance, they are problem-dependent, thus restricting their applicability in the real-world [7].

Over the last few years, data-driven features have become a promising alternative in recent approaches thanks to significant advances introduced by deep learning. These approaches are capable of building a high-level representation of the raw inputs automatically by learning features from multiple layers hierarchically [7]. They can be grouped into to five categories: (1) learning from video frames, (2) learning from frame transformations, (3) learning from hand-crafted features, (4) three-dimensional convolutional networks, and (5) hybrid models [11]. The main limitation of such approaches is their capacity in dealing with the temporal dimension [18].

The temporal structure of videos poses some challenges for training deep learning models [17]. First, the computational costs are expensive, as a huge amount of videos is required for training and they need to be decoded during this process. In addition, the number of learnable parameters is high, increasing the complexity of the model and, consequently, the chances of overfitting. These aspects are crucial for the performance of deep learning methods [10].

B. Video Compression

Compression of video data aims to minimize the spatio-temporal redundancies by exploiting image transforms and motion compensation [19]. Therefore, a lot of superfluous information can be discarded by processing compressed videos.

In most video compression algorithms, a video is splitted into three main types of pictures: intra-coded (I-frames), predicted (P-frames), and bidirectionally predicted (B-frames). Those pictures are organized into sequences of groups of pictures (GOPs) in video streams.

A GOP must start with an I-frame and can be followed by any number of I and P-frames, which are usually known as anchor frames. Between each pair of consecutive anchor frames can appear several B-frames.

Each video frame is divided into a sequence of non-overlapping macroblocks. For a video coded in 4:2:0 format, each macroblock consists of six 8x8 pixel blocks: four luminance (Y) blocks and two chrominance (CbCr) blocks. Each macroblock is then either intra- or inter-coded.

An I-frame is completely intra-coded: every 8x8 pixel block in the macroblock is transformed to the frequency domain using the discrete cosine transformation (DCT). The 64 DCT coefficients are then quantized (lossy) and entropy (run length and Huffman, lossless) encoded to achieve compression.

Each P-frame is predictively encoded with reference to its previous anchor frame (the previous I or P-frame). For each macroblock in the P-frame, a local region in the anchor frame is searched for a good match in terms of the difference in intensity. If a good match is found, the macroblock is represented by a motion vector to the position of the match together with the DCT encoding of the difference (or residue) between the macroblock and its match. The DCT coefficients of the residue are quantized and encoded while the motion vector is differentiated and entropy coded (Huffman) with respect to its neighboring motion vector. This is usually known as encoding with forward motion compensation. Macroblocks encoded by such a process are called as inter-coded macroblocks.

In order to achieve further compression, B-frames are bidirectionally predictively encoded using forward and/or backward motion compensation with reference to its nearest past and/or future I and/or P-frames.

The frame number, frame encoding type (I, P or B), the positions and motion vectors of inter-coded macroblocks, the number of intra-coded blocks, and the DC coefficients of each DCT encoded pixel block can be obtained by parsing and entropy (Huffman) decoding video streams. Those operations take less than 20% of the computational load in the full video decoding process [20].

III. Related Work

Unlike pixel-level information, the transform coefficients and the motion vectors from a compressed video provide useful information about its visual content, like appearance changes and motion information. These information can be easily extracted by partial decoding the video stream and used for recognizing actions. In this way, it is possible to improve not only effectiveness by taking advantage of richer information, but also efficiency by avoiding the full decoding of the video stream [18].

A few methods have explored the compressed domain as an alternative to speed up the computational performance [21]–[23]. Most of them are based on hand-crafted features and therefore their application is limited to specific domains. Focusing on a particular domain helps to reduce levels of ambiguity when analysing the visual content by applying prior knowledge of the domain during the analysis process [19].

The use of compressed domain information by deep learning methods is quite recent and has been exploit only by very few

works. The pioneering work of Zhang et al. [24], [25] extended the two-stream architecture of Simonyan and Zisserman [26] to use motion vectors instead of optical flow maps in the temporal stream network. However, videos still need to be decoded, since the spatial stream network is fed with RGB images.

The recent work of Wu et al. [18] has introduced CoViAR: a deep learning model fully trained on compressed videos. The key idea exploited by their work is to use RGB images obtained by decoding I-frames and motion features computed from P-frames as input to a multi-stream CNN, with one stream for each input, which are trained separately and then combined by a simple weighted average of their output scores. Although this approach is efficient, its capacity to learn the temporal structure is rather limited, since video frames are processed independently.

IV. Approach

In this section, we present our approach for action recognition: the Compressed Video Convolutional 3D network (CV-C3D). Section IV-A presents C3D: a tri-dimensional convolutional network. Section IV-B presents CoViAR: a method that uses information from compressed videos for action recognition. Finally, Section IV-C shows how the advantages of both C3D and CoViAR are exploited by our approach.

A. Convolutional 3D Network (C3D)

The Convolutional 3D (C3D) network [27] is capable of learning spatio-temporal patterns of video data directly from pixels. Basically, it extends the convolution along the temporal dimension, thus maintaining a certain temporal structure. For this, 3D filters instead of 2D ones are used in the convolutional layer. In this way, the feature maps of a convolution layer are connected to several continuous frames in the input layer, enabling to learn discriminative features along both spatial and temporal dimensions, like motion information. However, the number of parameters and the computational complexity of the model are inevitably increased, making them harder to train.

The C3D network is composed of 8 convolutional, 5 pooling, 3 fully connected layers (2 with ReLu and 1 with softmax activation). All the convolutional layers have a kernel of size $3\times3\times3$ (temporal$\times$spatial$\times$spatial dimensions) with strides of $1\times1\times1$. All the pooling layers are max-pooling with $2\times2\times2$ kernels and strides of $2\times2\times2$ (except for the first one that has a kernel of size $1\times2\times2$ with strides of $1\times2\times2$). The name of layers, number of filters for each of the convolutional layers, and number of neurons for each of the fully connected layers is presented in Figure 1. The input size of the C3D network is fixed to 16 frames with spatial resolution of 112×112 pixels.

Fig. 1. The architecture of the C3D network [27].

B. Compressed Video Action Recognition (CoViAR)

The Compressed Video Action Recognition (CoViAR) method [18] is a deep neural network capable to learn directly from compressed videos. Basically, it extends the Temporal Segment Networks (TSN) [28] to exploit three information readily available in MPEG-4 compressed streams: (1) RGB images encoded in I-frames (**I**), (2) motion vectors (**MV**) and (3) residuals (**R**) encoded in P-frames.

Following TSN [28], CoViAR learns temporal dynamics from multiple segments of a video. For this, uniform sampling is used to take a set of frames. Then, frame scores are obtained by feeding the network with one frame at a time. Finally, a video score is obtained by averaging the frame scores.

In terms of architecture, CoViAR is a multi-stream network containing three independent CNNs, one for each of the three information (i.e., **I**, **MV**, and **R**) extracted from compressed videos. To combine the individual CNNs, late fusion is performed by the weighted average of their prediction scores.

C. Compressed Video Convolutional 3D Network (CV-C3D)

On one hand, C3D is more suitable than CoViAR for modeling the temporal structure of videos, but the computational complexity makes it often impractical. On the other hand, CoViAR is much faster than C3D, but its capacity to capture temporal dynamics is limited.

Motivated by the aforementioned observations, we propose a Compressed Video Convolutional 3D network (CV-C3D). It combines the advantages of both C3D and CoViAR, yielding significantly improved performance. Basically, CV-C3D extends CoViAR by replacing CNNs with C3Ds. The similarities and differences of C3D, CoViAR, and CV-C3D can be observed in Figure 2.

Similar to CoViAR, the architecture of CV-C3D is a multi-stream network composed of three independent C3Ds, instead of CNNs, that are fed with the **I**, **MV**, and **R** information, respectively, obtained from MPEG-4 compressed streams. In this way, CV-C3D saves high computational load and memory usage in full decoding the video stream and also takes advantage of 3D convolutions to model temporal dynamics.

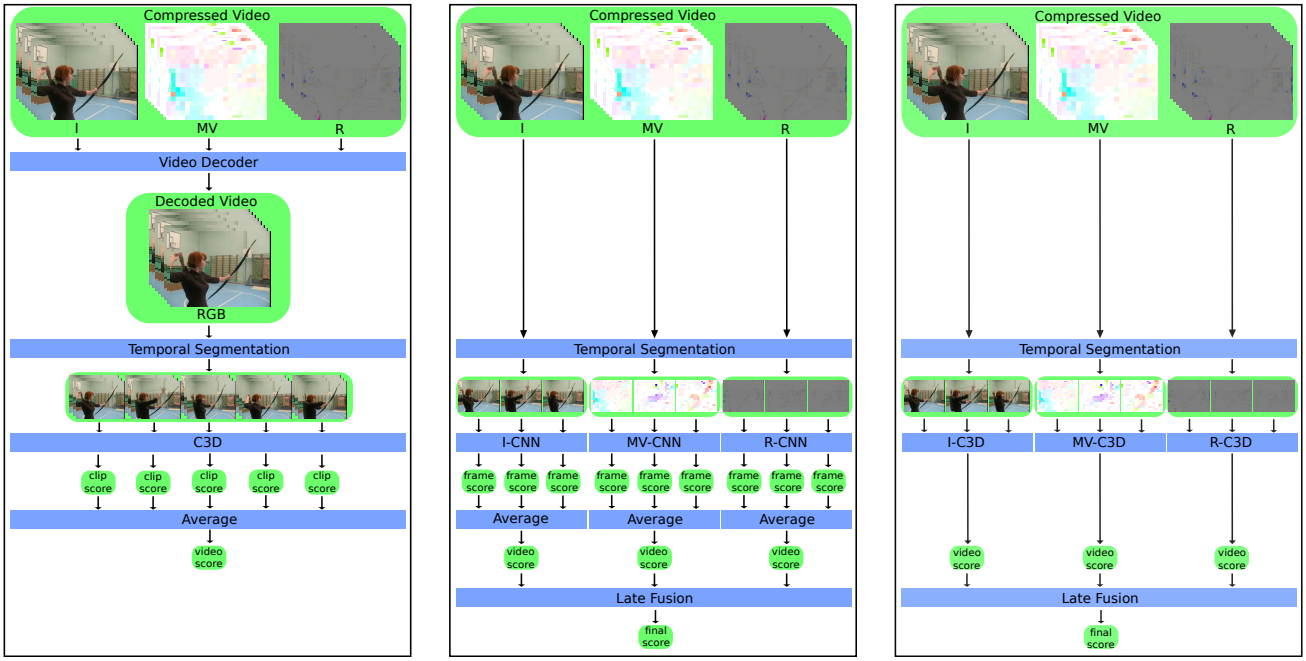
Unlike C3D, in CV-C3D, our C3Ds are fed with 16 frames obtained by uniform sampling, like in CoViAR, therefore they are not continuous. Different from CoViAR, these frames are passed all at once through the network, enabling CV-C3D to capture the temporal context efficiently. Finally, only 16 frames of a video are processed by the C3Ds of our CV-C3D, thus its computational complexity is acceptable.

V. Experiments and Results

For benchmarking purposes, experiments were conducted on two public datasets composed by a large and varied repertoire of different actions [29]: UCF-101 and HMDB-51.

The UCF-101 dataset[1] [30] is composed of $13,320$ videos (27 hours) collected from YouTube. All videos are in MPEG-4 format (at 25 frames per second and 320×240 resolution), in

[1] http://crcv.ucf.edu/data/UCF101.php (As of June 2019)



(a) C3D (b) CoViAR (c) CV-C3D

Fig. 2. Illustrations of (a) the C3D network [27], (b) the recent CoViAR [18] method, and (c) our proposed CV-C3D. Unlike CoViAR, where a video score is computed by averaging frame scores obtained by feeding a CNN with one frame at a time, our CV-C3D takes advantage of 3D convolutions used by C3D to compute a video score by feeding the network with all frames at once, enabling us to capture the temporal context efficiently.

color and with sound. They have large variations in camera motion, object appearance and pose, illumination conditions, etc. Those videos are distributed among 101 action classes and their duration varies from 1.06 to 71.04 seconds. Each of the action classes is divided into 25 groups containing 4-7 videos with common features, like actors and background.

The HMDB-51 dataset[2] [31] contains 6,766 videos (6 hours) collected from various sources, such as movies and internet sites like YouTube and Google. All videos are in MPEG-4 format (at 30 frames per second and with a fixed height of 240 pixels and width ranging from 176 to 592 pixels), in color and no sound. Such videos were annotated with information about camera motion, camera viewpoint, video quality, number of actors, visible body parts, etc. They are categorized into 51 action classes containing at least 102 videos in each and their duration varies from 0.64 to 35.44 seconds.

For evaluation, three training and testing splits are provided with the UCF-101 and HMDB-51 datasets. In our experiments, we follow the official evaluation protocol, which consists in evaluating the default training and testing splits separately and reporting the average accuracy over these three splits.

Following C3D [27], all videos were resized to 128×171 resolution. Then, we uniformly sample 16 frames from each video to feed the CV-C3D network. During testing phase, the action category is predicted by passing only a single center crop with size 112×112 through the network.

During training phase, we followed CoViAR [18] and applied three strategies for data augmentation: (1) color jittering, (2) horizontal flipping with 50% probability, and (3) random cropping with scale jittering, where the width and height of the cropped region are randomly selected on different scales (4 scales for **I**: 1, 0.875, and 0.75; and 3 scales for **MV** and **R**: 1, 0.875, and 0.75) and then resized to 112×112 resolution. The CV-C3D models were pre-trained on the Sports-1M dataset [32] and fine-tuned using Adam [33] with a batch size of 20. Step-decay was used to reduce the initial learning rate by a factor of 10 after a number of epochs. Table I presents the initial learning rates, the total number of epochs, and the step-decay scheduler setting used in our experiments.

The experiments were performed on a machine equipped with a processor Intel Core i7 6850K 3.6 GHz, 64 GBytes of DDR4-memory, and 4 NVIDIA Titan Xp GPUs. The machine runs Ubuntu 16.04 LTS (kernel 4.15.0) and the ext4 file system. Our approach was implemented in PyTorch (version 1.1.0) upon the CoViAR implementation[3].

Table II presents the classification accuracy achieved by CV-C3D in each of the three splits of the UFC-101 and HMDB-51 datasets. We compare the results obtained by feeding the network with different inputs and any combination of them. Using only one type of input, **I** and **R** obtained similar satisfactory results, while **MV** obtained inferior results, although when combined with the other inputs it was able to increase the performance, showing that **MV** offers complementary

[2]http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/ (As of June 2019)

[3]https://github.com/chaoyuaw/pytorch-coviar (As of July 2019)

TABLE I
THE HYPERPARAMETERS USED FOR TRAINING THE CV-C3D NETWORK.

Hyperparameter	**UCF-101**			**HMDB-51**		
	I	MV	R	I	MV	R
Initial learning rate	0.000075	0.0025	0.00125	0.00015	0.00125	0.00025
Total number of epochs	510			220	360	300
The step-decay scheduler setting	150, 270, 390			55, 110, 165	120, 200, 280	120, 180, 240

information to **I** and **R**. The best results were achieved by combining all three inputs, obtaining gains up to 12.9%, reaching classification accuracies of 83.9% on the UCF-101 dataset and 55.7% on the HMDB-51 dataset. These results indicate that the use of the information contained on the compressed video is promising.

TABLE II
CLASSIFICATION ACCURACY (%) ACHIEVED BY CV-C3D IN THE THREE SPLITS OF THE UFC-101 AND HMDB-51 DATASETS. THE NETWORK WAS FED WITH DIFFERENT INPUTS: (I) I-FRAMES, (M) MOTION VECTORS, AND (R) RESIDUALS. WE COMPARE THE PERFORMANCE OF EACH MODEL IN ISOLATION AND ALSO THEIR LATE FUSION (+) BY A WEIGHTED AVERAGE OF THEIR OUTPUT SCORES. THE BEST AND THE SECOND BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINING, RESPECTIVELY.

	I	MV	R	I+MV	I+R	I+MV+R (gain)
UCF-101						
Split 1	74.5	49.2	74.6	79.3	<u>81.3</u>	**83.1** (+8.5)
Split 2	75.0	50.2	76.9	80.3	<u>83.0</u>	**84.7** (+7.8)
Split 3	75.9	48.2	76.4	81.1	<u>82.4</u>	**83.9** (+7.8)
Average	75.1	49.2	75.9	80.2	<u>82.3</u>	**83.9** (+8.0)
HMDB-51						
Split 1	44.3	29.7	45.2	52.9	<u>53.0</u>	**57.7** (+12.5)
Split 2	40.9	30.3	39.5	<u>50.4</u>	47.7	**54.6** (+13.7)
Split 3	40.6	29.6	43.7	<u>50.8</u>	48.6	**54.8** (+11.1)
Average	41.9	29.9	42.8	<u>51.4</u>	49.8	**55.7** (+12.9)

Table III compares the computational complexity and classification accuracy of different networks. In terms of classification accuracy, CV-C3D achieved the third best performance on the UCF-101 dataset and the second best performance on the HMDB-51 dataset. Notice that CV-C3D performed better than C3D on both the datasets, indicating that the use of motion vectors and residuals has benefited our approach. On the other hand, the highest classification accuracies were achieved by CoViAR. We believe that it is because, in the testing phase, CoViAR is fed with 25 video frames chosen by uniform sampling, from which are extracted 5 crops with flips. By taking more frames and using data augmentation, CoViAR benefits from much more information than CV-C3D in terms of both appearance and temporal dynamics. In addition, the CV-C3D architecture is based on C3D, which is a VGG [34] alike structure. CoViAR is built on top of a ResNet [35] architecture, which takes advantage of residual connections, making the learning process easier. However, CV-C3D has the lowest computational complexity among all the networks, requiring only 12% GFLOPs used by CoViAR.

Table IV compares the classification accuracy of CV-C3D and the state-of-the-art compressed video methods. Again, CV-C3D achieved the second best performance on the HMDB-51 dataset, showing that it retains high accuracy while greatly reducing computational cost. However, the results for CV-C3D were slightly worse than EMV-CNN and DTMV-CNN on the UCF-101 dataset. Unlike CV-C3D, in addition to motion vectors, they also use optical flow during the training phase. This feature can also be used by CV-C3D, but its computation is significantly slower as video decoding is required.

TABLE III
COMPARISON OF THE COMPUTATION COMPLEXITY (GFLOPS) AND CLASSIFICATION ACCURACY (%) OF DIFFERENT NETWORKS. THE BEST AND THE SECOND BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINING, RESPECTIVELY.

		Accuracy (%)	
	GFLOPs	UCF-101	HMDB-51
ResNet-50 [36]	<u>3.8</u>	82.3	48.9
ResNet-152 [36]	11.3	83.4	46.7
C3D [27]	38.5	82.3	51.6
Res3D [37]	19.3	<u>85.8</u>	54.9
CoViAR [18][5]	4.2	**90.4**	**59.1**
CV-C3D	**0.5**	83.9	<u>55.7</u>

[5]For a fair comparison, we considered the results reported by CoViAR [18] using only information from compressed domain. To improve its accuracy, CoViAR use optical flow besides motion vectors.

TABLE IV
COMPARISON OF THE CLASSIFICATION ACCURACY (%) ON THE UCF-101 AND HMDB-51 DATASETS FOR STATE-OF-THE-ART COMPRESSED VIDEO BASED METHODS. THE BEST AND THE SECOND BEST RESULTS ARE HIGHLIGHTED IN BOLD AND UNDERLINING, RESPECTIVELY.

	UCF-101	HMDB-51
EMV-CNN [24]	86.4	51.2[6]
DTMV-CNN [25]	<u>87.5</u>	55.3
CoViAR [18]	**90.4**	**59.1**
CV-C3D	83.9	<u>55.7</u>

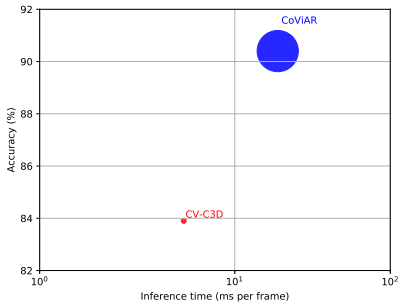
[6]This result was reported in [25] and refers to the classification accuracy obtained only on Split 1 of the HMDB-51 dataset. We included here just for reference.

The key advantage of our approach is its computational efficiency. To evaluate its efficiency, we measured the average inference time per-frame, which refers to the time spent to prepare data and pass through the network. For this, we sum up the total time taken to feed the multi-streams sequentially. To obtain a fair comparison, the forwarding time of CoViAR was measured using the authors' implementation[4], upon which we implemented CV-C3D using the same code optimization.

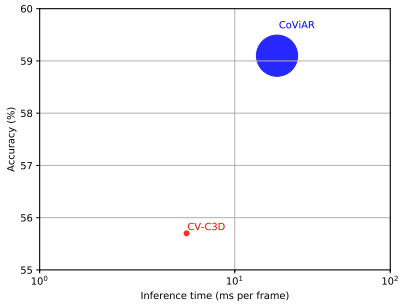
Figure 3 compares the classification accuracy, the network computation complexity, and the inference time for CV-C3D

[4]https://github.com/chaoyuaw/pytorch-coviar (As of July 2019)

and CoViAR on the UCF-101 and HMDB-51 datasets. Overall, CV-C3D is one order of magnitude faster than CoViAR.



(a) UCF-101



(b) HMDB-51

Fig. 3. Comparison of the classification accuracy (%) and the inference time (ms per frame) for CV-C3D and CoViAR on the UCF-101 and HMDB-51 datasets. Node size denotes the network computation complexity (GFLOPs).

VI. Conclusion

In this paper, we presented a Compressed Video Convolutional 3D network (CV-C3D). Our approach combines the advantages of both C3D and CoViAR. Following CoViAR, our method is capable to learn directly from compressed videos, speeding up the processing time. Similar to C3D, 3D convolutions are used in our network to model temporal dynamics. Architecturally, CV-C3D is a multi-stream network composed of three independent C3Ds, whose predictions are combined by late fusion.

Our network has the lowest computational complexity among all the compared approaches. In terms of classification accuracy, our approach performed similar to the other methods on the UCF-101 dataset and achieved the second best performance on the HMDB-51 dataset. For both the UCF-101 and HMDB-51 datasets, the best results were achieved by CoViAR. In contrast, CV-C3D is much faster than CoViAR for performing inferences.

As future work, we plan to evaluate the use of other 3D CNNs in our approach, like Res3D [37] or I3D [38]. In addition, we intend to evaluate different strategies for modelling the temporal structure of videos (e.g., using recurrent neural networks), as well as smarter fusion strategies for combining the outputs from CNNs related to different streams. Also, we want to perform an extensive analysis of the parameter-space of our approach. The evaluation of CV-C3D in large-scale datasets, like Kinetics [39], and in other applications besides action recognition is also a possible future work.

Acknowledgment

This research was supported by the São Paulo Research Foundation - FAPESP (grant #2018/21837-0), the FAPESP-Microsoft Research Virtual Institute (grant #2017/25908-6), and the Brazilian National Council for Scientific and Technological Development - CNPq (grants #423228/2016-1 and #313122/2017-2) for funding. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] Y. Yan, C. Xu, D. Cai, and J. J. Corso, "Weakly supervised actor-action segmentation via robust multi-task ranking," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017, pp. 1022–1031.

[2] M. Koohzadi and N. M. Charkari, "Survey on deep learning methods in human action recognition," *IET Computer Vision*, vol. 11, no. 8, pp. 623–632, 2017.

[3] J. Almeida, A. Rocha, R. S. Torres, and S. Goldenstein, "Making colors worth more than a thousand words," in *ACM International Symposium on Applied Computing (ACM-SAC'08)*, 2008, pp. 1180–1186.

[4] F. S. P. Andrade, J. Almeida, H. Pedrini, and R. S. Torres, "Fusion of local and global descriptors for content-based image and video retrieval," in *Iberoamerican Congress on Pattern Recognition (CIARP'12)*, 2012, pp. 845–853.

[5] O. A. B. Penatti, L. T. Li, J. Almeida, and R. S. Torres, "A visual approach for video geocoding using bag-of-scenes," in *ACM International Conference on Multimedia Retrieval (ICMR'12)*, 2012, pp. 1–8.

[6] I. C. Duta, J. R. R. Uijlings, B. Ionescu, K. Aizawa, A. G. Hauptmann, and N. Sebe, "Efficient human action recognition using histograms of motion gradients and VLAD with descriptor shape information," *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 22 445–22 472, 2017.

[7] D. Wu, N. Sharma, and M. Blumenstein, "Recent advances in video-based human action recognition using deep learning: A review," in *International Joint Conference on Neural Networks (IJCNN'17)*, 2017, pp. 2865–2872.

[8] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[9] S.-M. Kang and R. P. Wildes, "Review of action recognition and detection methods," *CoRR*, vol. abs/1610.06906, 2016.

[10] M. Asadi-Aghbolaghi, A. Clapés, M. Bellantonio, H. J. Escalante, V. Ponce-López, X. Baró, I. Guyon, S. Kasaei, and S. Escalera, "A survey on deep learning based approaches for action and gesture recognition in image sequences," in *IEEE International Conference on Automatic Face & Gesture Recognition (FG'17)*, 2017, pp. 476–483.

[11] F. Zhu, L. Shao, J. Xie, and Y. Fang, "From handcrafted to learned representations for human action recognition: A survey," *Image and Vision Computing*, vol. 55, pp. 42–52, 2016.

[12] S. Herath, M. T. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and Vision Computing*, vol. 60, pp. 4–21, 2017.

[13] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-temporal VLAD encoding for human action recognition in videos," in *International Conference on MultiMedia Modeling (MMM'17)*, 2017, pp. 365–378.

[14] L. A. Duarte, O. A. B. Penatti, and J. Almeida, "Bag of attributes for video event retrieval," in *SIBGRAPI - Conference on Graphics, Patterns and Images (SIBGRAPI'18)*, 2018, pp. 447–454.

[15] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-temporal vector of locally max pooled features for action recognition in videos," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017, pp. 3205–3214.

[16] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Simple, efficient and effective encodings of local deep features for video action recognition," in *ACM International Conference on Multimedia Retrieval (ICMR'17)*, 2017, pp. 218–225.

[17] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. C. Niebles, "What makes a video a video: Analyzing temporal information in video understanding models and datasets," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018, pp. 7366–7375.

[18] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl, "Compressed video action recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018, pp. 6026–6035.

[19] R. V. Babu, M. Tom, and P. Wadekar, "A survey on compressed domain video analysis techniques," *Multimedia Tools and Applications*, vol. 75, no. 2, pp. 1043–1078, 2016.

[20] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*, 2nd ed. Kluwer Academic Publishers, 1997.

[21] J. Almeida, N. J. Leite, and R. S. Torres, "Comparison of video sequences with histograms of motion patterns," in *IEEE International Conference on Image Processing (ICIP'11)*, 2011, pp. 3673–3676.

[22] V. Srinivasan, S. Lapuschkin, C. Hellge, K.-R. Müller, and W. Samek, "Interpretable human action recognition in compressed domain," in *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP'17)*, 2017, pp. 1692–1696.

[23] M. Tom, R. V. Babu, and R. G. Praveen, "Compressed domain human action recognition in H.264/AVC video streams," *Multimedia Tools and Applications*, vol. 74, no. 21, pp. 9323–9338, 2015.

[24] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector cnns," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 2016, pp. 2718–2726.

[25] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with deeply transferred motion vector cnns," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2326–2339, 2018.

[26] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Annual Conference on Neural Information Processing Systems (NIPS'14)*, 2014, pp. 568–576.

[27] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *IEEE International Conference on Computer Vision (ICCV'15)*, 2015, pp. 4489–4497.

[28] 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 (ECCV'16)*, 2016, pp. 20–36.

[29] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 633–659, 2013.

[30] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.

[31] H. Kuehne, H. Jhuang, E. Garrote, T. A. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *IEEE International Conference on Computer Vision (ICCV'11)*, 2011, pp. 2556–2563.

[32] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F.-F. Li, "Large-scale video classification with convolutional neural networks," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'14)*, 2014, pp. 1725–1732.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 2016, pp. 770–778.

[36] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017, pp. 7445–7454.

[37] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "Convnet architecture search for spatiotemporal feature learning," *CoRR*, vol. abs/1708.05038, 2017.

[38] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'17*, 2017, pp. 4724–4733.

[39] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *CoRR*, vol. abs/1705.06950, 2017.