Grenze International Journal of Engineering and Technology, Jan Issue

Fast Real-Time Video Analytics for Human Action Recognition in Compressed Domain

Praveenkumar S M[1], Prakashgoud Patil[2] and P. S. Hiremath[3]
[1-3]Department of Master of Computer Applications, KLE Technological University, Hubballi-580031, Karnataka, India
Email: praveenkumar_sm@kletech.ac.in, {prakashpatil, pshiremath} @ kletech.ac.in

***Abstract*—Herein, a novel methodology is proposed for real-time human activity detection and recognition in compressed domain of videos by using motion vectors and long-term recurrent convolutional networks (MVLRCN). The videos in MPEG-4 and H.264 compression formats are considered for the present study. Any video source without any prior setup could be considered by adapting the proposed method to various video codecs and camera settings. Existing algorithms for human action recognition in a compressed domain video have some limitations in this regard such as (i) requirement of keyframes at a fixed interval, (ii) usage of only P frames, and (iii) normally support only a single codec. These limitations are overcome in the proposed method by using arbitrary keyframe intervals, using both P and B frames, and supporting both MPEG-4 and H.264 codecs. The experimentation is carried out using the benchmark datasets, namely, UCF101, HMDB51 and THUMOS14, and the recognition accuracy in compressed domain is found to be comparable to that observed in raw video data by using other recent methods. The proposed MVLRCN method has outperformed other methods in the literature significantly by improving the video action recognition inference in the compressed videos.**

***Index Terms*— Convolution neural network, Video processing, Data association, Motion vectors, Long-term recurrent convolution networks.**

I. INTRODUCTION

In computer vision, the task of a human activity detection and recognition is the fundamental step in the design of intelligent surveillance systems that automatically recognize various human actions in a real scene video. Recent research efforts in this direction have yielded significant research progress. Of late, as digital video capturing of some social events has become a more common widespread phenomenon, many researchers' efforts are devoted to improving human activity detection for such videos, especially in the context of security surveillance, information retrieval, content analysis, and HCI (human-computer interaction) involving video data. Generally, a video in compressed formats contains object motion information. The video codecs H.264 and MPEG-4 Part2, that are often utilized in surveillance video cameras of surveillance systems, deal with only a few frames with complete RGB images, which are termed as key frames. Each intermediate frame (non-key frame) is constituted by a residual frame with accompanying motion vectors, that encode variations between the current frame and its subsequent frame. The video data size is thus significantly reduced in terms of storage and transmission time requirements. In the literature, several methods have been reported for human activity detection and recognition in the compressed domain videos [1]. The limitation of these methods arises due to the

Grenze ID: 01.GIJET.9.1.504_2

usage of a keyframe interval of fixed length and only the P frames, which consequently reduce the video processing rate and visual quality in the compression domain [2]. Moreover, setting a specific codec, length of keyframe interval, or type of frame might not be allowed in some video sources. With a motivation to overcome the observed limitations of the existing methods, it is proposed to develop a faster human action recognition in a compressed domain video data streaming by employing deep learning techniques. The videos in MPEG-4 and H.264 compression format are considered for the present study. It follows the spatial-temporal action recognition using motion vectors and Long-term Recurrent Convolutional Network (MVLRCN). It has a CNN as encoder for encoding deep spatial state vectors, which is spatially deep, and an LSTM as decoder for capturing temporal features represented in the form of motion vectors, which is temporally deep.

Major improvement in the proposed MVLRCN over previous works is the flexibility provided in the support for compressed video analytics with the usage of variable length intervals of key frame and usage of both P and B frames. This technique allows for varying video compression formats and thus making it more amenable for application to a wider range of real-world scenarios. The validation of the experimental results of MVLRCN is done by performing action recognition in the videos taken from the benchmark video datasets, namely, UCF101, HMDB51and THUMOS14, and analysis of its performance is done for different codecs, namely, MPEG-4 and H.264, and frame types, namely, P and B frames.

The main novel contributions in the present work are summarized as following:

- A fast real-time action recognition algorithm, called MVLRCN, in compression domain (MPEG-4 or H.264) using motion vectors and LRCN, which supports keyframes at intervals of variable length, and both P and B frames.
- Evaluation of MVLRCN on UCF101, HMDB51 and THUMOS14 benchmark video datasets.

The organization of paper comprises five sections. The Section II contains a brief account of the literature related to the present problem. The proposed action recognition algorithm MVLRCN is described in the Section III. The results analysis of experiments is discussed in the Section IV. Finally, in the Section V, the conclusions are given.

II. Related Work

The problem of human action recognition in videos is well-studied using various datasets prepared for specific purposes [12, 13, 14]. The two-stream approach [3], which is the most common approach to this problem, employs two convolutional neural networks, one to process RGB frames and the other to process optical flow separately. Although this method has improved results significantly, it incurs higher computational cost in the form of two CNNs and optical flow computations. The CNNs have used 3D convolution [4] to exploit the temporal domain features of optical that yield better results. Alternative approaches for generating temporal domain features are based on RNNs [5] or other aggregation techniques. While these methods have shown enhanced performance significantly in various action recognition applications, the high parameter counts lead to higher cost of computation.

A. Video Compression

Digital video compression achieves reduction in the number of bits per frame by removing spatial as well as temporal redundancies and thus transforming to a form suitable for efficient storage and transmission of videos. Video codecs split a video into a Group of Pictures (GOP), which. Consists of an Intra frame (I-frame) followed by a sequence of inter frames (P-frames or B-frames). I-frame is a self-contained RGB frame with full visual representation, while P-frames or B-frames only represent the changes that occur with respect to the preceding frame. The data representation in P-frames are motion vectors and residuals. Motion vectors denote the shift of a block of pixels from a one frame to the next frame. The residual denotes the error of the motion prediction (B-frames).

B. Action Recognition in Compressed Domain

Wu et al. [1] have designed a deep learning based CoViAR model for action recognition in the compressed domain itself. Herein, three different 2D CNNs are used, one each for training with each of the three types of compressed domain data: the I-frame, and the P-frame with the residuals and motion vectors. In order to enhance performance, optical flow is also accounted, which essentially decodes the video for the purpose of extracting the desired optic flow, and thus defeats the main advantage of operating directly in the compressed domain itself. Further, it enhances the computational cost in terms of memory space and time, since each optical flow frame computation time required is in the range 20-80ms. In [6, 7], the knowledge learnt in optical flow is transferred

by correlating the optical flow and motion vectors. In [8], the DMC-Net is proposed as an improvement over CoViAR by employing a lightweight algorithm for motion vector noise reduction and capturing finer details of the motion, wherein optical flow computations are avoided and thus significantly reduce the computation time. As in CoViAR, the DMC-Net also uses multiple 2D CNN models, one each for each of the different modalities. Recently, following CoViAR [1] and DMC-Net[8], Huo et al.[9] used 2D CNNs in combination with MobileNet [10], one each for every modality. Further, the three modalities (I-frame, Motion Vectors, Residuals) are combined in a new pooling layer, namely, Temporal Trilinear Pooling, and thereby enhance the recognition performance.

III. PROPOSED METHODOLOGY

A. Overview

The proposed action recognition approach MVLRCN is based on LRCN (Long-term Recurrent Convolutional Networks) [11] and motion vectors. It has two main stages: action recognition and data association, as shown in the Fig.1. During the action recognition stage, new action A^tat time t is identified based on the previous action A^{t-1}at time t-1 and motion vectors M^tat time t. In the update step, that is executed on each frame in intervals of fixed length K, resets the accumulated recognition error. During update step, action recognizer LRCN recognizes the action D^t. The Hungarian method [21] for assignment of recognized action to predicted action through motion vectors is implemented. The LRCN is trained on compressed videos, namely, MPEG-4 and H.264. It uses a combo of CNNs and LSTMs, wherein CNN performs encoding deep spatial state vectors and LSTM performs as decoder for capturing temporal features.

CNN: It undergoes training parallelly along with other CNNs and action of each is independent of others.

LSTM: It models learning from sequential data of varying lengths. In action recognition step, the algorithm takes sequential input< F1, F2, F3,...,Ft > (sequence of frames at time 1, 2, 3, …, t) and predicts recognized video action class at each corresponding time step < Y1, Y2, Y3,…,Yt>. In the LRCN model, these outputs are averaged to predict the output class label Y and then compare with the action predicted through the motion vectors to predict the recognition error. Thus, the computation in update step is more expensive than that in recognition step.

B. Motion Vectors in compressed domain

To reduce the memory requirement for storage or transmission of a video in compressed domains, the encoded video contains only a few key frames in full form. Each of the non-key frames, that are between any pair of consecutive key frames, is segmented into macroblocks and then motion vectors are computed which point to macroblocks having similar appearance in the subsequent frames (depicted as red arrows in Fig. 2). Thus, reuse of information from adjacent frames leads to an efficient encoding of the content in a video. In MPEG-4, each macroblock is of size 16 ×16 pixels, while in H.264, macroblock is of size varying between4 ×4 and 16 ×16 pixels. A non-key frame is referred to as a P frame when motion vectors have reference only to preceding frames, while it is a B frame when motion vectors have references to both preceding and succeeding frames. The non-key frames in MPEG-4 are only P frames, whereas the H.264 utilizes P as well as B frames. The motion vectors, which imply the object motions in a video, assist in visual human action recognition.

The motion vectors are computed in compressed domain videos, say MPEG-4 or H.264 [2]. In the set $M^t \subset Z^{10}$ of motion vectors computed in the current frame F^t at time t, the number of $|M^t| = j^t$ may vary with change in frame. For the key frames, $M^t = \emptyset$. Each element $m_j^t \in M^t$ represents a motion vector given by the ten-tuple

$$m_j^t = (s_j^t, v_{j,w}^t, v_{j,h}^t, x_{j,src}^t, y_{j,src}^t, x_{j,dst}^t, y_{j,dst}^t, v_{j,x}^t, v_{j,y}^t, v_{j,s}^t) \quad (1)$$

where the vector's reference frame $F^{t+s_j^t}$ has the offset s_j^t with respect to the current frame F^t.In case of a P vector, s_j^t is negative, while it is positive for a B vector. The $v_{j,w}^t$ and $v_{j,h}^t$denote the width and height, respectively, of the vector's macroblock. As illustrated in the Fig.3, each motion vector in the current frame starts at the center of its corresponding macroblock $(x_{j,dst}^t, y_{j,dst}^t)$ and points to the center of the macroblock $(x_{j,src}^t, y_{j,src}^t)$ of similar appearance in the reference frame. The $v_{j,x}^t$ and $v_{j,y}^t$ denote x- and y-components of motion vector, after scaling by $v_{j,s}^t$ which satisfy

$$x_{j,src}^t = \lfloor x_{j,dst}^t + v_{j,x}^t / v_{j,s}^t \rfloor, \quad y_{j,src}^t = \lfloor y_{j,dst}^t + v_{j,y}^t / v_{j,s}^t \rfloor \quad (2)$$

giving values for the components.

C. Handling of Key Frames

In earlier methods for action recognition, the update step is performed on key frames only, while action recognition step is on non-keyframes only, since motion vector computation, that is needed for action recognition, is not done in key frames. Thus, it requires key frames at intervals of fixed length. However, the proposed method MVLRCN adopts a novel strategy for action recognition and updation steps, which are scheduled in regular intervals irrespective of the frame type. When a key frame is encountered in an action recognition step, the motion vectors of its preceding frame are considered, i.e., set $M^t = M^{t-1}$, and then action recognition is performed on it. The underlying assumption is that there is not much change in the action performed by the object in between two consecutive frames, and thus, $M^{t-1} \cong M^t$ and it could be used for action recognition in case of a key frame.

D. Action Recognition

In the action recognition stage, MVLRCN predicts action A^t at time t based on action A^{t-1} at instant t-1 and motion vectors M^t at time t. MVLRCN computations, which are designed to handle P frame only or both P and B frames, comprises the following steps:

a) Retrieve the subset $\widetilde{M}^t$ of the of the set M^t of motion vectors having non-zero magnitude:

$$\widetilde{M}^t = \{m_j^t \in M^t | ((v_{j,x}^t)^2 + (v_{j,y}^t)^2)^{1/2} > 0\} \tag{3}$$

b) Do scaling of the x- and y-components of each motion vector m_j^t in $\widetilde{M}^t$ by the temporal distance s_j^t :

$$\overline{v}_{j,x}^t = v_{j,x}^t / s_j^t,\ \overline{v}_{j,y}^t = v_{j,y}^t / s_j^t. \tag{4}$$

c) Apply motion scale $v_{j,s}^t$ to obtain the floating-point resolution:

$$\tilde{v}_{j,x}^t = \overline{v}_{j,x}^t / v_{j,s}^t, \tilde{v}_{j,y}^t = \overline{v}_{j,y}^t / v_{j,s}^t. \tag{5}$$

d) Compute the new reference points $(\tilde{x}_{j,src}^t, \tilde{y}_{j,src}^t)$ of the vectors with respect to rescaled x- and y-components:

$$\tilde{x}_{j,src}^t = x_{j,dst}^t + s_j^t \tilde{v}_{j,x}^t, \tilde{y}_{j,src}^t = y_{j,dst}^t + s_j^t \tilde{v}_{j,y}^t \tag{6}$$

e) For each action $a_i^{t-1} = (x_i^{t-1}, y_i^{t-1}, w_i^{t-1}, h_i^{t-1})$ in A^{t-1} obtain the subset $\widehat{M}_i^t$ of motion vectors in $\widetilde{M}^t$ corresponding to new reference points $(\tilde{x}_{j,src}^t, \tilde{y}_{j,src}^t)$

$$\widehat{M}_i^t = \{m_j^t \in \widetilde{M}^t | x_i^{t-1} - \frac{w_i^{t-1}}{2} \le \tilde{x}_{j,src}^t \le x_i^t - 1 + \frac{w_i^{t-1}}{2}$$
$$\wedge\ y_i^{t-1} - h_i^{t-1} \le \tilde{y}_{j,src}^t \le y_j^{t-1} + \frac{h_i^{t-1}}{2}\} \tag{7}$$

f) For each $\widehat{M}_i^t$, evaluate aggregates (mean or median) $\mu_{i,x}^t$ and $\mu_{i,y}^t$ of the components $\tilde{v}_{i,x}^t$ and $\tilde{v}_{i,y}^t$ of all vectors $m_j^t \in \widehat{M}_i^t$.

Handling of motion vectors in both P and B frames is done by the rescaling of motion components in step (b). The vectors in P are rotated by 180° so as to point them in the same direction as that of the vectors in B. Thus, the vectors in P and B frames are treated in the same manner during further computations in the subsequent steps.

IV. Results and Discussion

A. Dataset

The proposed algorithm MVLRCN for action recognition is experimented using the publicly available benchmark datasets, namely, UCF 101dataset [13], THUMOS14 [14] and HMDB-51[12], which have the

characteristics as given in the Table I. UCF101 has 13,320 trimmed videos with 101 classes of human actions. THUMOS14 was announced for action recognition challenge 2014 and it has 15,994 videos. Unlike UCF101, THUMOS14 has untrimmed videos with 24 classes of action. HMDB-51 has 6,766 trimmed videos with 51 classes of action. The performance of the proposed MVLRCN is evaluated using the metric mAP (Mean Average Precision). In case of THUMOS14, the MVLRCN testing at every 20 frames is conducted, since videos are untrimmed and have longer frame sequences. During these experiments, the evaluation of recognition speed is done in terms of number of frames per seconds (fps). The dataset is divided in to two set in ratio 80:20, one for training and other for testing.

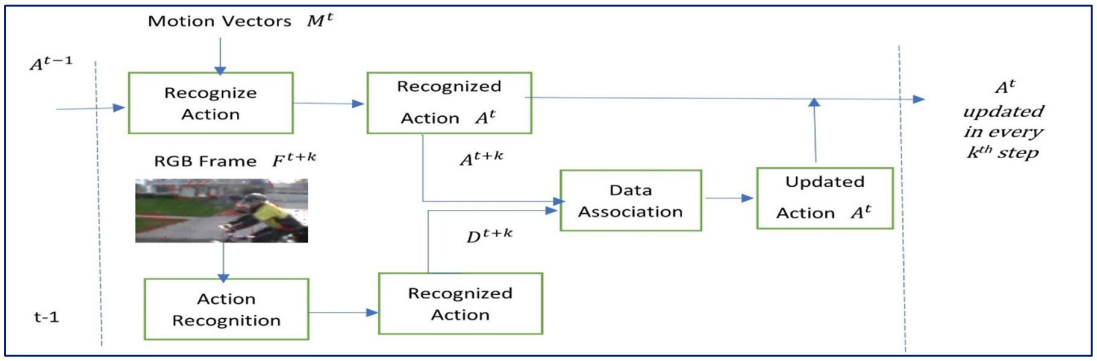
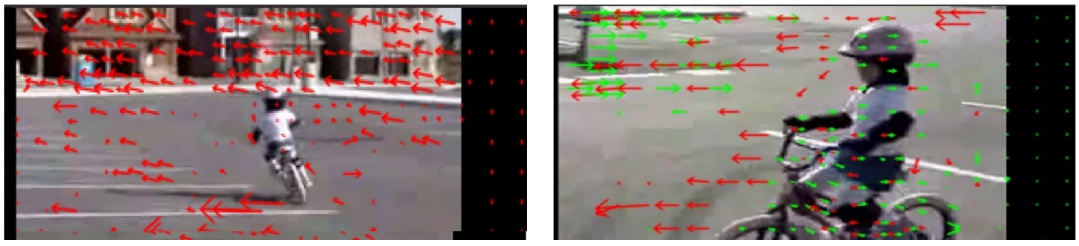


Figure 1. The Framework of the proposed MVLRCN. During Action Recognition, Action A^t is recognized using the previous Action A^{t-1} and motion vectors M^t of current frame F^t. During update step done on every k^{th} frame, Action recognizer runs on RGB frame F^{t+k} . Data association is done using the Hungarian method which matches recognized Action D^{t+k} from the RGB frame with predicted Action A^{t+k} from Motion vectors



a. b.

Figure 2. Motion vectors in (a) MPEG-4 and (b) H.264 video frame. Macroblock borders are white, while motion vectors in P and B frames are red and green, respectively. Macroblocks with no motion vectors are intra-coded

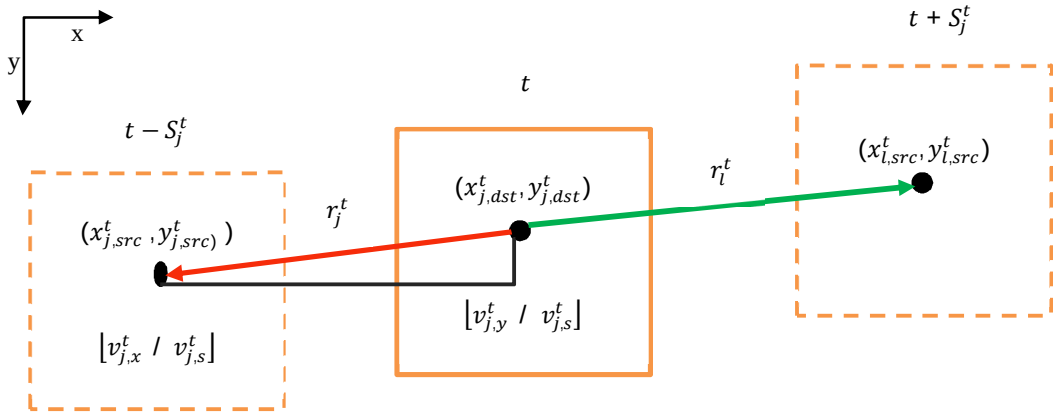


Figure 3. Each inter-coded macroblock (with solid boundary) in a P frame (MPEG-4/H.264) at time t has one motion vector v_j^t, which points to an image patch with a similar appearance (with dashed boundary) in the previous frame at time $t - s_j^t$. In case of H.264, each macroblock (with solid boundary) is a P frame or B frame at time t, which has a motion vector v_j^t (red) or v_l^t (green), respectively, or both. Motion vector in B frame points to an image patch with a similar appearance in the next frame at $t + s_l^t$

B. Comparison of MVLRCN variants

The performance of two variants of MVLRCN on UCF 101, THUMOS14 and HMDB-51 in MPEG-4 compressed domain, wherein the first variant uses the arithmetic mean and the second one uses the median of the vector components, is compared in the Table II. It is observed that both the variants perform closely.

Table I: Characteristics of Datasets UCF 101[13], THUMOS14[14] and HMDB-51[12].

Video specs	**UCF 101**	**THUMOS14**	**HMDB-51**
Number of Videos	13,320	15904	6,766
Video groups	25	-	-
Trimmed/Untrimmed videos	Trimmed	Untrimmed	Trimmed
Action classes	101	24	51

Table II: Results for variants of MVLRCN on UCF 101, THUMOS 14 and HMDB-51.

Dataset	**Method**	**Recognition Accuracy**	**Speed (FPS)**
UCF 101	MVLRCN (Mean, MPEG-4, P)	**95.85%**	**422**
	MVLRCN (Median, MPEG-4, P)	95%	276.5
	MVLRCN (Mean, H.264, P)	90.25%	179.9
	MVLRCN (Median, H.264, P and B)	91%	162.4
THUMOS14	MVLRCN (Mean, MPEG-4, P)	**86.25%**	**319**
	MVLRCN (Median, MPEG-4, P)	85.85%	233
	MVLRCN (Mean, H.264, P)	83.25%	243
	MVLRCN (Median, H.264, P and B)	84.3%	224
HMDB-51	MVLRCN (Mean, MPEG-4, P)	**73.25 %**	**408**
	MVLRCN (Median, MPEG-4, P)	72.25%	318
	MVLRCN (Mean, H.264, P)	71.45	145
	MVLRCN (Median, H.264, P and B)	72.21%	135

Table III: Comparison of recognition accuracy of MVLRCN with state-of-the art on datasets UCF101 and HMDB51.

Method	Accuracy	
	UCF 101	HMDB51
CoViaR [1]	90.4%	59.1%
DMC-net (Resnet18) [8]	90.9%	62.8%
TTP [9]	87.2%	58.2%
MFCD-Net [15]	93.2%	66.9%
MVLRCN (Proposed)	**95.85%**	**73.25%**

C. Comparison of MVLRCN variants in different compression domains

The performance of action recognition of MVLRCN (Mean and Median) on UCF 101, THUMOS14 and HMDB-51, for MPEG-4 with P frames, H.264 with P frames, and H.264 with both P and B frames, is evaluated using the metrics, namely, recognition accuracy (%) and speed (FPS). The results presented in Table II show that MVLRCN performs better with higher recognition accuracy and speed in MPEG-4 compressed domain in comparison with that in H.264 compressed domain. Further, it is observed that, in case of H.264, action recognition accuracy is marginaly improved when both P and B frames are handled rather than only the P frame, but with a loss in speed. The action recognition on MPEG-4 videos is done with higher speed than that on H.264 videos, sinice the macroblocks in each frame of H.264 have smaller size than that of MPEG-4 which results in computation of a higher number of motion vectors in each frame.

D. Performance Analysis

The performance comparison of the proposed MVLRCN with other compressed domain action recognition algorithms in the literature, namely, CoViaR [1], DMC-net (Resnet18) [8], TTP [9], MFCD-Net [15] is done in the Table III. The mean variant of MVLRCN and MPEG-4 frame sequences are used for the purpose of performance comparison on all the three datasets considered for the experiments. The robustness of the proposed MVLRCN action recognizer in terms of accuracy and speed is demonstared by its performance comparison with other methods, that are also trained using the same datasets, namely, UCF 101 and THUMOS14 datasets, is shown in the Tables IV and V,respectively. The sample results of action recognition using proposed MVLRCN

in compressed domains MPEG-4 and H.264 using UCF101 dataset, which contains different action scenarios, are shown in the Figs.4 and 5. In the Fig.4, the 'Biking' action is identified with motion vectors also being extracted. In the Fig.5, 'Diving' action is identified with motion vectors also being extracted. From the Tables III, IV and V, it is observed that the proposed method MVLRCN for action recognition has outperformed other methods in terms of accuracy as well as speed.

Table IV: Performance comparison of proposed MVLRCN with other methods using UCF101 dataset .

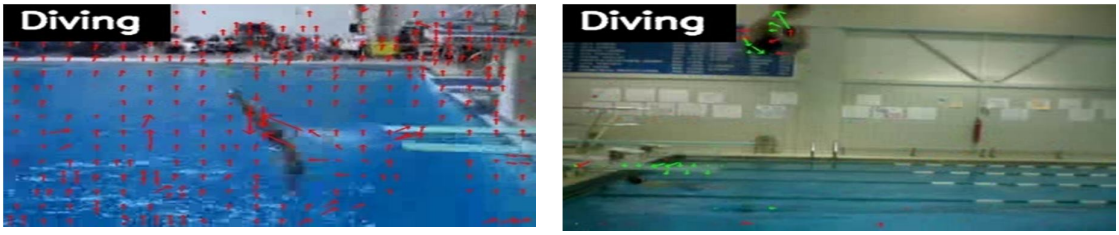
Method	Recognition Accuracy	Speed (FPS)
V. Kantorov et al. [16]	78.5%	132.8
D. Tran et al. [17] (1 net)	82.3%	313.9
D. Tran et al. [17] (3 net)	85.2%	-
H. Wang et al. [18]	85.9%	2.1
K. Simonyan et al. [3]	88.0%	14.3
B. Zhang et al. [7]	86.4%	390.7
MVLRCN (Proposed)	**95.85%**	**422**

Table V: Performance comparison of proposed MVLRCN with other methods using THUMOS14 dataset.

Method	Recognition Accuracy	Speed (FPS)
M. Jain et al. [19](Objects (GPU))	44.7%	-
L. Wang et al. [20]	62.0%	< 2.38
M. Jain et al. [19] (Motion (CPU))	63.1%	2.38
M. Jain et al. [19] (Objects+Motion (CPU+GPU))	71.6%	< 2.38
B. Zhang et al. [7] (EMV+RGB-CNN)	61.5%	403.2
MVLRCN (Proposed)	**86.25 %**	**422**

a b

Figure 4. Sample results for UCF101 dataset with Biking action and corresponding motion vectors in (a) MPEG-4 domain and (b) H.264 domain



a b

Figure 5. Sample results for UCF101 dataset with Diving action and corresponding motion vectors in (a) MPEG-4 domain and (b) H.264 domain

V. CONCLUSION

In this paper, a novel method MVLRCN is developed for real-time action recognition in the compressed domains, namely, MPEG-4 and H.264, which has flexibility to deal with videos having variable length of key frame intervals and to consider both P and B frames. The experimentation of the proposed algorithm is carried out on three datasets, namely, THUMOS14, UCF-101, HMDB-51. The analysis of results indicate that the proposed algorithm performs with higher recognition accuracy and speed, which makes it more effective in real-time action recognition in compressed domain as compared to other methods. Limitation of MVLRCN is the poor action recognition performance for H.264 domain in comparison with MPEG-4, this issue would be addressed in future work.

REFERENCES

[1] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, “Compressed Video Action Recognition,” in 2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 6026–6035, IEEE, jun 2018.

[2] L. Bommes, X. Lin and J. Zhou, "MVmed: Fast Multi-Object Tracking in the Compressed Domain," 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2020, pp. 1419-1424, doi: 10.1109/ICIEA48937.2020.9248145.

[3] K. Simonyan and A. Zisserman, “Two-Stream Convolutional Networks for Action Recognition in Videos,” in Neurips, pp. 568–576, 2014.

[4] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4724-4733, doi: 10.1109/CVPR.2017.502.

[5] J. Donahue et al., "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 677-691, 1 April 2017, doi: 10.1109/TPAMI.2016.2599174.

[6] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, “Real-Time Action Recognition With Deeply Transferred Motion Vector CNNs,” IEEE Trans. Image Process., vol. 27, pp. 2326–2339, may 2018.

[7] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, “Real-time Action Recognition with Enhanced Motion Vector CNNs,” in CVPR, 2016.

[8] Z. Shou, Z. Yan, Y. Kalantidis, L. Sevilla-Lara, M. Rohrbach, X. Lin, and S.-F. Chang, “DMC-Net: Generating Discriminative Motion Cues for Fast Compressed Video Action Recognition,” tech. rep., Columbia Univ. & Facebook, 2019.

[9] Y. Huo, X. Xu, Y. Lu, Y. Niu, Z. Lu, and J.-R. Wen, “Mobile Video Action Recognition,” tech. rep., 2019.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 4510–4520, IEEE Computer Society, dec 2018.

[11] M. Ranjit and G. Ganapathy, "A Study on the use of State-of-the-Art CNNs with Fine Tuning for Spatial Stream Generation for Activity Recognition," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019, pp. 1-5, doi: 10.1109/ICECCT.2019.8869360.

[12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: A large video database for human motion recognition,” in 2011 Int. Conf. Comput. Vis., pp. 2556–2563, IEEE, nov 2011.

[13] Khurram Soomro, A. R. Zamir, and M. Shah, “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild,” tech. rep., University of Central Florida, 2012.

[14] "Jiang, Y.-G. and Liu, J. and Roshan Zamir, A. and Toderici, G. and Laptev, I. and Shah, M. and Sukthankar, R.", "THUMOS Challenge: Action Recognition with a Large Number of Classes", http://crcv.ucf.edu/THUMOS14/, 2014.

[15] B. Battash, H. Barad, H. Tang and A. Bleiweiss, "Mimic The Raw Domain: Accelerating Action Recognition in the Compressed Domain," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2926-2934, doi: 10.1109/CVPRW50498.2020.00350.

[16] V. Kantorov and I. Laptev. Efficient feature extraction, encoding, and classification for action recognition. In CVPR’14, pages 2593–2600, 2014.

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

[18] H. Wang and C. Schmid. Action recognition with improved trajectories. In ICCV’13, pages 3551–3558, 2013.

[19] M. Jain, J. C. van Gemert, and C. G. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In CVPR’15, pages 46–55, 2015.

[20] L. Wang, Y. Qiao, and X. Tang. Action recognition and detection by combining motion and appearance features. In THUMOS Action Recognition challenge, 2014.

[21] H. W. Kuhn, “The Hungarian method for the assignment problem,” Naval Research Logistics Quarterly, vol. 2, pp. 83–97, 1955.