Journal of Theoretical and Applied Information Technology
15th May 2024. Vol.102. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFICIENT VIDEO COMPRESSION USING DEEP JOINT OPTIMIZATION METHOD WITH MOTION ESTIMATION AND INTER-FRAME PREDICTION

[1]DR. M. CHANDRA SEKHAR [2]SAMPURNIMA PATTEM , [3]B.SHRAVAN KUMAR [4]DR. SUMAGNA PATNAIK

[1]Professor, CSE,Presidency University Bengaluru,
[2] Sr.Asst.Professor ,CVR College of Engineering,College
[3]Research Scholar JNTUH Hyderabad
[4]Principal Bhaskar Engineering College Moinabad Mandal, R.R.District
Telangana (India)

[1]mchandrasekhar@presidencyuniversity.in [2]psampurnima@gmail.com [3]shravanbk6@gmail.com
[4]sumagnapa72@gmail.com[4]

ABSTRACT

In the contemporary era, there is unprecedented increase in multimedia content, especially videos, leading to consumption of more bandwidth when transmitted. Video compression is the technique that leverages performance of video transmission as it reduces original size of the video. Though the conventional video compression methods have classical architecture to encode motion and residual information efficiently, it lacks the ability to have non-linear representation of data. In this paper, we proposed a framework named Artificial Intelligence (AI) enabled Video Compression Framework (AIVCF) which exploits the traditional classical architecture and combines it with a deep learning model for non-linear data representation. This framework has ability to have joint optimization of underlying components. Convolutional Neural Network (CNN) is used to reconstruct current frames by getting motion information through a process known as optical flow estimation. The information of given video is compressed using deep learning models in auto-encoder fashion. The framework strikes balance between quality and compression ability. An algorithm named Deep Joint Optimization for Video Compression (DJO-VC) is proposed to realize the AIVCF. The proposed framework is evaluated with empirical study. The experimental results, in terms of PSNR and SSIM revealed that the proposed framework outperforms existing models such as H.264.

Keywords – *Video Compression, Deep Learning, Convolutional Neural Network, Artificial Intelligence Enabled Video Compression Framework*

1. INTRODUCTION

Deep learning based approaches have paved way for solving many real world problems. They are widely used in computer vision applications due to their inspiration with learned solutions video/image processing problems such as super resolution, action recognition and compression to mention few. Thus deep learning became an indispensable approach for nonlinear signal processing. Moreover, it is found from recent works that learned models have achieved significant performance improvements in perceptual quality measures when compared with state of the art [1]. From the literature, there are many deep learning models found for video compression. Ma *et al*. [2] opined that CNN has potential to solve problems associated with signal processing. Yang *et al*. [3] proposed a compression technique known as Recurrent Learned Video Compression (RLVC). Liu *et al*. [6] explored many CNN based models for solving video compression problems. Pessoa *et al*. [10] proposed a deep learning based framework for video compression with end to end learning by exploiting spatio-temporal auto-encoders. Zhang *et al*. [13] proposed a CNN based methodology for post processing towards video compression. They explored Generative Adversarial Network (GAN) architecture comprising generator (G) and discriminator (D) for efficiency in video compression. Nagaraj *et al.* [17] used deep learning technique like LSTM to improve feature extraction and apply it for data compression. As found in the related works, it is observed that the existing compression methods use only few reference frames to compress a video frame which jeopardises the ability to extract temporal correlation among different video frames. To address the aforementioned problem, we proposed a deep learning based framework. Figure 1 shows the outline of our approach for video frame prediction.

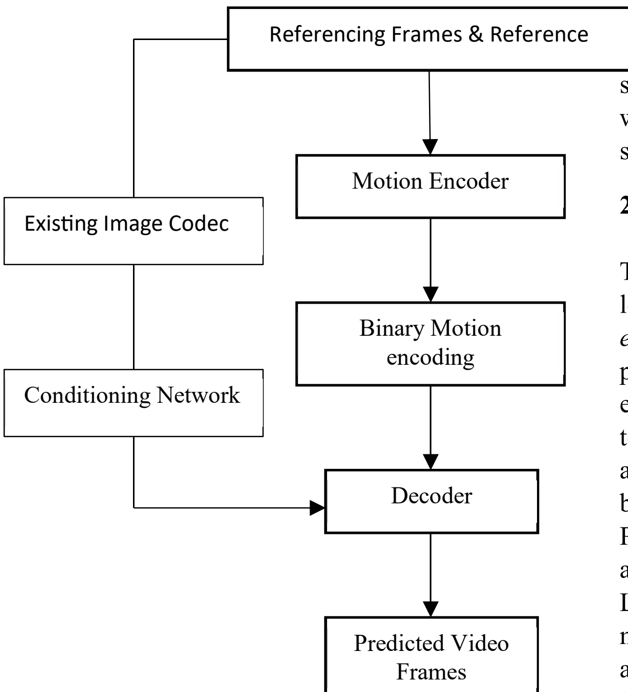

Figure 1: Overview Of Video Frame Prediction Process

As presented in Figure 1, our approach in this paper for video prediction process is illustrated. It makes use of reference frames and reference. They are subjected to motion encoder, binary motion encoding and decoder towards prediction of video frames. The process involves usage of existing image codec and conditioning network. More details of the proposed approach are provided in Section 3. Our contributions in this paper are as follows.

1. A framework named Artificial Intelligence (AI) enabled Video Compression Framework (AIVCF) is proposed. It exploits the traditional classical architecture and combines it with a deep learning model for non-linear data representation.
2. An algorithm named Deep Joint Optimization for Video Compression (DJO-VC) is proposed to realize the AIVCF.
3. A prototype application is developed to evaluate the proposed framework and underlying algorithm.

The remainder of the paper is structured as follows. Section 2 reviews latest related works on deep learning based video compression echniques. Section 3 presents our framework and lgorithm. Section 4 gives details of experimental setup. Section 5 presents experimental results while section 6 concludes our work besides specifying future scope.

2. RELATED WORK

This section reviews latest related works on deep learning based video compression techniques. Ma *et al.* [2] opined that CNN has potential to solve problems associated with signal processing. They emphasized that cutting edge video compression techniques are possible with deep learning models as they can exploit parallel computing supported by Graphical Processing Unit (GPU) and Tensor Processing Unit (TPU). Yang *et al.* [3] proposed a compression technique known as Recurrent Learned Video Compression (RLVC). RLVC makes use of Recurrent Probability Model (RPM) and Recurrent Auto-Encoder (RAE). It is a learned video compression technique which could extract temporal correlations mong frames. However, it still suffers from rate-distortion performance and complexity. Lu *et al.* [4] proposed an end-to-end framework for video compression using deep learning. It makes use of pixel wise motion information and auto-encoder with joint optimization considering rate-distortion trade-off. It exploits non-linear representation capability of deep neural networks (DNNs). Chen *et al.* [5] proposed a methodology for video compression using deep feature coding and lossy compression technique. It enables cloud based visual analysis by reducing overhead with novel data transmission strategy.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Liu *et al*. [6] explored many CNN based models for solving video compression problems. They suggested to deepen learning processes with variants of CNN for further improvement in compression performance. Xu *et al*. [7] made a comparative study of traditional methods and deep learning based approaches for compressing videos. They found that end to end learning and usage of different learning based entropy methods could improve compression performance. Westland *et al*. [8] exploited decision trees in order to reducing complexity in the process of video compression. Friedland *et al*. [9] investigated on the influence of perceptual compression on deep learning models. Their empirical study has found that deep learning models have the capability to exploit perceptual compression. They advocate the importance of using novel metrics rather than tuning hyper parameters. Pessoa *et al*. [10] proposed a deep learning based framework for video compression with end to end learning by exploiting spatio-temporal auto-encoders. It has provision for rate-distortion optimization to reduce inconsistencies among video frames. They achieved latent space representation through by obtaining spatio-temporal dependencies. Poyser *et al*. [11] explored CNN architectures and investigated the impact of lossy video compression methods on them. They found that lossy compression has potential to impact performance of deep learning models. Valenzise *et al*. [12] focused on deep learning based approaches for image compression. They have made subjective evaluation of two deep CNN models for image compression and found that both do have performance improvement over traditional methods.

Zhang *et al*. [13] proposed a CNN based methodology for post processing towards video compression. They explored Generative Adversarial Network (GAN) architecture comprising generator (G) and discriminator (D) for efficiency in video compression. Chen *et al*. [14] proposed a compression model to compress deep learning models for ease of transmission over Internet. Liu *et al*. [15] proposed a deep learning model for distortion prediction in image compression use cases. Birman *et al*. [16] investigated on various deep learning models including CNN, auto encoder and GAN for video compression. Nagaraj *et al*. [17] used deep learning technique like LSTM to improve feature extraction and apply it for data compression. Krishnaraj *et al*. [18] considered an IoT use case known as Internet of Underwater Things (IoUT). In such environment, they implemented real-time image compression using DWT-CNN model. Das *et al*. [19] explored JPEG compression and deep learning models to incorporate security to images. Chen *et al*. [20] proposed a methodology for knowledge as a service for automatic compression of images using deep learning.

Table 1: Shows Summary Of Most Relevant Deep Learning Models For Video Compression

Reference	**Approach**	**Algorithm /Technique**	**Dataset**	**Limitations**
Ravi *et al*., [3]	Deep auto-encoder	Recurrent Auto-Encoder (RAE) and Recurrent Probability Model (RPM)	Vimeo-90k [37]	More complexity
Dong *et al*., [7]	Deep neural networks	CNN based model	-	Only baseline models are explored.
Zhang *et al*., [13]	CNN based post processing	CNN	JVET [38] and N0254 [39]	Improvement in training and reduction in computational complexity are still desired.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Nagaraj *et al.*, [17]	Deep learning and feature extraction	LSTM	MNIST	Error rate is more.
Krishnaraj *et al.*, [18]	Deep learning based on DWT	DWT-CNN	UWSN	It has issues with noisy environment.
Wiedemann *et al.*, [22]	DNN based Universal Compression	Context-based Adaptive Binary Arithmetic Coder (CABAC)	ImageNet, CIFAR10, MNIST	Achievable compression limits are to be investigated.
Duan *et al.*, [24]	Deep learning with collaborative compression	Video Coding for Machines	PKU-MMD	It has overfitting problem.
Sinha *et al*, [29]	CNN based approach	Temporal 3-D CNN based encoder and Y-style CNN based decoder	UCF 101, Kinetic-5K and UVG	Lower visual quality and loss of motion information.

Prakash *et al.* [21] proposed a novel CNN architecture to achieve semantic perceptual image compression. In the process, they exploited multi-structure Region of Interest (ROI). Wiedemann *et al.* [22] proposed a common compression technique using deep learning and named it as DeepCABAC. It has provision to reduce rate-distortion and also a novel quantization scheme. Vega *et al.* [23] proposed deep learning method for examining quality of live video streaming. Duan *et al.* [24] investigated on the notion of collaborative compression with video coding approaches. Chen *et al.* [25] proposed a deep feature compression technique for intelligent sensing. Kuanar *et al.* [26] focused on HEVC in-loop filtering using deep learning for improving quality of decoder. Li *et al.* [27] proposed a deep learning model based on Trellis Coded Quantization for image compression. Other contributions found in the literature include HEVC intra-frame coding with deep learning [28] and Temporal 3-D CNN based method for video compression [29]. Table 1 shows summary of most relevant related works on deep learning based video compression. From the literature, it is understood that the conventional compression methods use only few reference frames to compress a video frame which jeopardises the ability to extract temporal correlation among different video frames. It is improved with deep learning models as they support non-linear approach. However, there is need for further research to have more robust approach in video compression using deep learning.

Table 2: Notations Used In The Paper

Notation	**Description**
I	reference frames
P, B	referencing (P-frame and B-frame) frames
E	Encoder
D	Decoder
Cond	conditioning network
M	Mask
L	integer levels
$\vec{V_g}$	ground truth flow
$\vec{V_p}$	the flow vectors derived from the frames
EPE	end-point-error
L_R	reconstruction loss

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

L_B	Loss
⋋	Hyperparameter
L_F	the optical flow losses
α	weighting term

3. PROPOSED FRAMEWORK

We proposed a framework named Artificial Intelligence (AI) enabled Video Compression Framework (AIVCF). It has different mechanism and underlying algorithm for efficient video compassion. The framework has provision for combining conventional architecture and deep learning model such as CNN for non-linear data representation. CNN is used to reconstruct current frames through optical flow estimation for obtaining motion information. Auto-encoder based deep learning model is used to compress information of given video. For compressing given video, it is important to achieve deep motion estimation and frame prediction. Figure 2 shows the architectural overview for predicting P-frames. The input video frames are subjected to different operations including encoding and decoding in order to predict P-frames. The input video frameworks are taken by motion encoder which automatically compresses motion information among the frames. Then binary motion code is generated by the encoder. Each frame in the video input is given in such a way that it contains reference denoted as I and a referencing B or P frame. The binarization process made by motion encoder is based on thresholding. It exploits the binarization function discussed in [30]. In the process of training the outcome of motion encoder is in the form of binary value with noise added. The

value is either -1 or 1. In the process, the estimation of gradients is done using the procedure provided in [31].

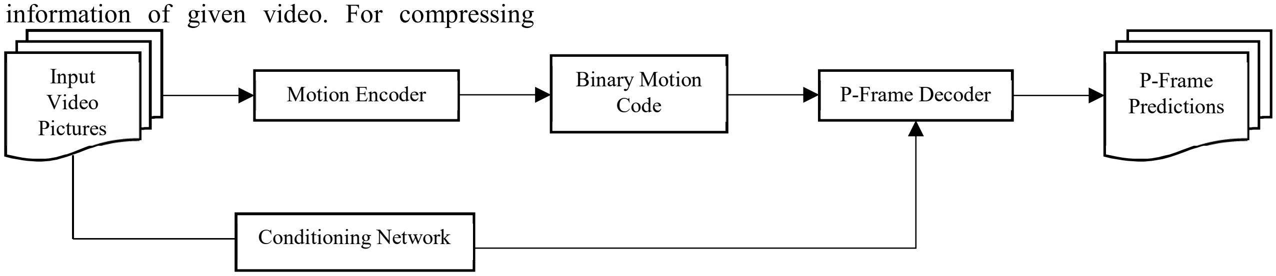

Figure 2: Architectural Overview Of P-Frame Prediction Process

The features of I-frame are extracted at the decoder using conditional network. As per the binarized motion encoding information, the extracted features are exploited to predict P-frames. An existing codec is used for image compression and it is not actually done by the conditional network. The P-frame prediction procedure is expressed as in Eq. 1. Table 2 has details of notations used in this paper.

$$\widehat{P_{1,\ldots,t}} = D(E(I_0, P_{1,\ldots,t}), Cond(I_0)) \quad (1)$$

The decoder denoted as D exploits reference frames in I with the help of conditioning network. Thus it is able to predict sequence of frames to be P-frames. Encoder on the other hand always compresses the inputs. The bit rate in the process of P-frame detection is determined by the output channels used in the encoding layer. In order words, extrapolation is carried out by decoder.

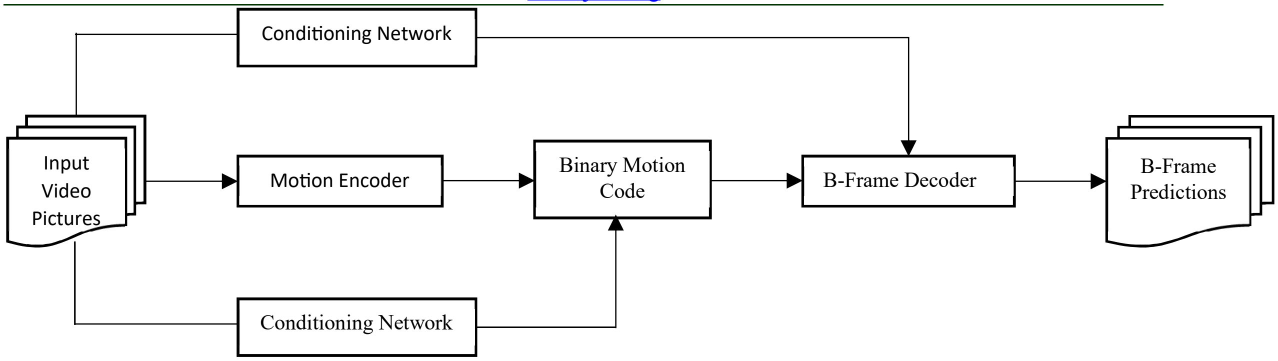

Figure 3: Architectural Overview Of B-Frame Prediction Proces

As presented in Figure 3, it illustrates the process involved in B-frame prediction. The input video frameworks are taken by motion encoder which automatically compresses motion information among the frames. Then binary motion code is generated by the encoder. Each frame in the video input is given in such a way that it contains reference denoted as I and a referencing B or P frame. The binarization process made by motion encoder is based on thresholding. It exploits the binarization function discussed in [30]. In the process of training the outcome of motion encoder is in the form of binary value with noise added. The value is either -1 or 1. In the process, the estimation of gradients is done using the procedure provided in [31]. The features of I-frame are extracted at the decoder using conditional network. As per the binarized motion encoding information, the extracted features are exploited to predict B-frames. An existing codec is used for image compression and it is not actually done by the conditional network. The B-frame prediction procedure is expressed as in Eq. 2.

$$\widehat{B_{1,\ldots,t}} = D(E(I_0, B_{1,\ldots,t}, I_{t+1}), Cond_0(I_0), Cond_t(, I_{t+1}) \quad (2)$$

The decoder denoted as D exploits reference frames in *I* with the help of conditioning network. Thus it is able to predict sequence of frames to be B-frames using interpolation unlike decoder in P-frame prediction process. Encoder on the other hand always compresses the inputs. The bit rate in the process of B-frame detection is determined by the output channels used in the encoding layer. In case of both the processes found in Figure 2 and Figure 3, L2 reconstruction loss is computed in the training phase as expressed in Eq. 3.

$$L_R = \| B - \hat{B} \|^2 \quad \text{or} \quad \| P - \hat{P} \|^2, \quad (3)$$

In the training period, the decoder is given access to I-frame content (represents an entire image in video). However, the at the time of testing encoding and are taken place independently with the help of an image codec. Convolutional layers (multi-scale) discussed in [32] are preferred in the prediction process as the motion in given video occurs differently at different scales. Each convolutional layer has ability to exploit learned "scale invariant feature transform (SIFT)". The conditioning process in the given architectures at the decoder has ability to detect the frame correctly. When compared with raw video frames, the binary motion codes obtained in the prediction process are more compressible. The proposed designs for detection P and B frames support different frame sizes and different number of images/pictures present in the given video.

Algorithm 1: Deep Joint Optimization For Video Compression (DJO-VC)

Algorithm: Deep Joint Optimization for Video Compression (DJO-VC)
Input:
Video denoted ***V*** containing a set of pictures
Output:
Compressed video ***V'***
1. Start
2. Initialize P-Frames vector ***X***
3. Initialize B-Frames vector ***Y***
4. Initialize binary motion code vector ***M***
5. ***I***←GenerateIFrames(***V***)
Detection of P-Frames
6. For each I-frame ***i*** in ***I***

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
For each reference and reference frame r in R
    M←MotionEncoder(r)
    IF CondDecoderExtrapolation(M) →P-Frame Then
        Add M to X
    End If
  End For
End For
Detection of B-Frames
For each I-frame i in I
  For each reference and reference frame r in R
    M←MotionEncoder(r)
    IF CondDecoderInterpolation(M) →B-Frame Then
        Add M to Y
    End If
  End For
End For
V'←GenerateOutput(I, X, Y)
Compute loss functions
Performance evaluation
Display statistics
Return V'
```

As presented in Algorithm 1, it takes given video as input and generates a compressed video with better performance. It has deep CNN based multi-scale convolutional layers used in the prediction of P and B frames. The algorithm reflects prediction of P-frames and also B-frames with automatic compression prior to generating a final compression video which is used for transmission of networks. The motion encoder performs compression of motion information from given video pictures and represents data in the form of -1 or 1. The decoder used in P-frame detection uses extrapolation for detection of P-frames while the decoder used in B-frame detection uses interpolation for detection of B-frames.

In order to bring about flexibility in generation of binary motion codes we incorporate time dimension using the approach presented in [33]. It helps sin adapting bit rate based on different regions of video and the content involved in the regions. The encoder identifies spatio-temporal locations and allocate fixed number of bits. The underlying motion encoder uses number of bit channels based on points in space-time. In the process a bit distribution map, denoted as Bmap is created. The encoder produces bits for each video frame and they are divided into L groups. Each Bmap element is denoted as bt, h, w which is quantized as expressed in Eq. 4.

$$Q_L = \left(b_{t,h,w}\right) = \left[Lb_{t,h,w}\right] \tag{4}$$

For each space-time point, it determines number of bit levels needed. A bit masking is generated further in order to get rid of allocation of non-integer bit numbers. It is expressed as in Eq. 5.

$$m_{c,t,h,w} = \begin{cases} 1, & if\ c \le \frac{c_{bnd}}{L} Q_L(b_{t,h,w}) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

In order to ensure that the decoder ascertains bit stream correctly, an additional loss term is computed as in Eq. 6.

$$L_B = \sum_{t,h,w} b_{t,h,w} \tag{6}$$

This loss term is used to prevent bit assignment to video regions that are stationary that can be ignored from the given I-frame. The operations in Eq. 4 and Eq. 5 are non-differentiable. In order to achieve final dynamic bit assignment approximation is made as expressed in Eq. 7.

$$\frac{\partial m_{c,t,h,w}}{\partial b_{t,h,w}} = \begin{cases} L, & if\ Lb_{t,h,w} - 1 \le \frac{[cL]}{c_{bnd}} \le Lb_{t,h,w} + 2 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

We also explored a loss term based on optimal flow for improving motion compression process. Between two frames of video, optical flow reflects the pixel movement as discussed in [34]. The optical flow based loss function in terms of end point error is as in Eq. 8 and cosine similarity is expressed in Eq. 9.

$$L_{EPE} = \sqrt{\| \overrightarrow{V_g} - \overrightarrow{V_p} \|^2}, \tag{8}$$

$$L_{cosine} = 1 - \frac{\overrightarrow{V_g} . \overrightarrow{V_p}}{\|\overrightarrow{V_g}\| \ \|\overrightarrow{V_p}\|} . \tag{9}$$

The two measures such as L_{EPE} and L_{cosine} functions differently as the latter penalizes directional deviations between predicted vectors and ground truth. After training the models in Figure 2 and Figure 3 (after getting pre-trained models), further training is carried out to gain knowledge on dynamic bit assignment. This optimization function with 150 additional epochs is expressed as in Eq. 10.

$$L_{RB}= L_R + \lambda L_B \quad (10)$$

It combines two kinds of losses computed in Eq. 3 and Eq. 6 in order to improve the evaluation process. In order to strike balance between compression rate and reconstruction quality we introduced a hyper parameter known as λ.

$$L_{RF}= L_R + \propto L_F \quad (11)$$

The loss function expressed in Eq. 11 is used in order to minimize difference between predicted frame's and input frame's optical flow. Here the optimal flow loss is denoted by L_F and distortion loss is denoted by L_R. The performance of the proposed framework is evaluated using three objective metrics. Peak Signal to Noise Ratio (PSNR) is one of the metrics used to know quality of predicted video frames. Video Multi-Method Assessment Fusion (VMAF) [35] is another metric used for evaluation. The third metric is known as Structural SIMilarity index (SSIM) [36].

4. EXPERIMENTAL SETUP

Python data science platform with Python 3 is used for application development and algorithm implementation. The deep neural network architectures for P-Frame and B-Frame detection procedures are built using Pytorch 1.0.1. Other important Python libraries used for implementation are OpenCV, ScikitImage and ScikitVideo. The deep neural networks involved in P and B frame detection procedures are trained using Hallywood dataset [57]. The dataset has 475 diversified video clips in AVI format. To be compatible with data loader in the implementation, each clip is transcoded with H.264 [5] codec. Out of 475 video clips, we used 435 for training and 40 for validation. Initial learning rate for deep learning architectures is set to 0.0001. The optimizer is known as Adam and the number of epochs used in the empirical study is 150.

5. RESULTS AND DISCUSSION

The proposed learned video compression technique using deep learning is evaluated and compared with conventional codecs. Different performance metrics used for evaluation are PSNR, VMAF and SSIM.

Figure 4: Result Of Pre-Processing To Obtain Set Of Pictures From Video

As presented in Figure 4, the given video is subjected to pre-processing and it has resulted in a set of pictures that are used further to achieve learned video compression. The resultant pictures are used as input to the proposed deep learning approach and the compression process is based on learning which is found to have better performance.

Figure 5: Compressed Frames With Bit Rate Per Pixel 0.2121

It is observed from the empirical study that the bit rate per pixel has its influence on the visual quality of the compressed frames. As presented in Figure 5, the pictures acquired from a video are subjected to deep learning based compression. The visual quality visible here is with bit rate per pixel 0.2121.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 6: Compressed frames with bit rate per pixel 0.2176

As presented in Figure 6, the pictures acquired from a video are subjected to deep learning based compression. The visual quality visible here is with bit rate per pixel 0.2176.

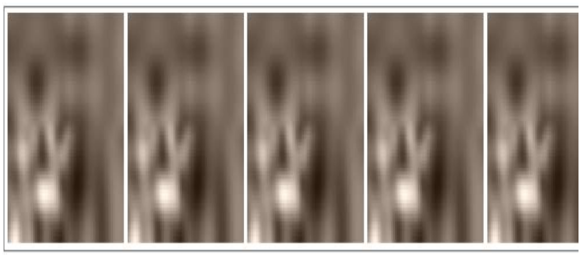

Figure 7: Compressed frames with bit rate per pixel 0.2597

As presented in Figure 7, the pictures acquired from a video are subjected to deep learning based compression. The visual quality visible here is with bit rate per pixel 0.2597.

5.1 Compression Performance with P-Frame Prediction

This section presents results of empirical study using the proposed framework AIVCF considering P-Frame prediction for video compression. It is also compared with video compression using B-Frame detection with optimization. The optimized version exploits dynamic bit assignment for improving compression efficiency. Experiments are made with different bits-per-pixel and the performance is evaluated in terms of PSNR, SSIM and VMAF. In other words, rate-distortion analysis is made and observations are recorded.

Table 3: PSNR comparison between video compression with B-Frame detection and its optimized variant

Bits-Per-Pixel	PSNR	
	AIVCF (B-Frame Detection)	**AIVCF (B-Frame Detection) with Optimization**
0.02	29.85	31.28
0.04	30.05	31.45
0.06	30.2	31.55
0.08	30.4	31.55
0.1	30.45	31.55
0.12	30.48	31.55

As presented in Table 3, video compression performance of B-Frame detection process and its optimized variant is compared against bit rate in terms of PSNR.

Table 4: SSIM comparison between video compression with B-Frame detection and its optimized variant

Bits-Per-Pixel	SSIM	
	AIVCF (B-Frame Detection)	**AIVCF (B-Frame Detection) with Optimization**
0.02	0.844	0.878
0.04	0.849	0.883
0.06	0.852	0.884
0.08	0.857	0.884
0.1	0.86	0.884
0.12	0.864	0.884

As presented in Table 4, video compression performance of B-Frame detection process and its optimized variant is compared against bit rate in terms of SSIM.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5: VMAF comparison between video compression with B-Frame detection and its optimized variant

Bits-Per-Pixel	VMAF	
	AIVCF (B-Frame Detection)	AIVCF (B-Frame Detection) with Optimization
0.02	71.4	75
0.04	72.5	75.7
0.06	72.9	76.2
0.08	73.1	76.2
0.1	73.4	76.2
0.12	73.7	76.2

As presented in Table 5, video compression performance of B-Frame detection process and its optimized variant is compared against bit rate in terms of VMAF.

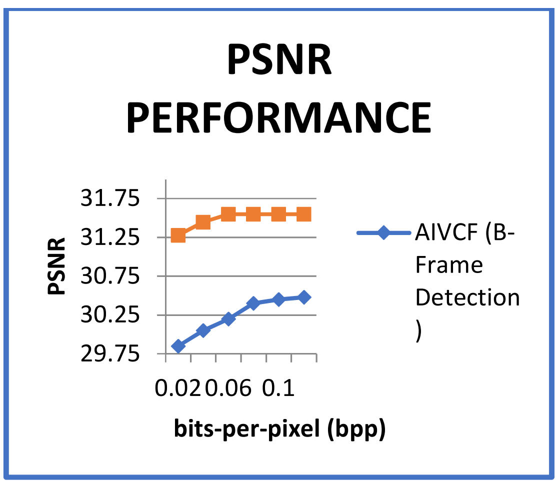

Figure 8: Rate-distortion analysis in terms of PSNR

As presented in Figure 8, bits-per-pixel rate is used for experimentation. Different rates of bits-per-pixel are provided in horizontal axis. With the given rate, PSNR is computed to ascertain video compression performance. Higher in PSNR value indicates less distortion and higher quality in compression. An important observation is that bits-per-pixel (rate) has its influence on PSNR. Another observation is that the optimized version of B-Frame prediction process used for video compression is found to have better performance over its un-optimized variant. When rate is 0.02 the proposed framework with B-Frame prediction process has achieved PSNR 29.85 while its optimized version that exploits dynamic bit assignment achieved PSNR 31.28. This trend is true with all rates with which experiments are made for deep learning based video compression. Therefore, it can be concluded that the optimized version of B-Frame prediction process shows significantly better performance over its un-optimized counterpart.

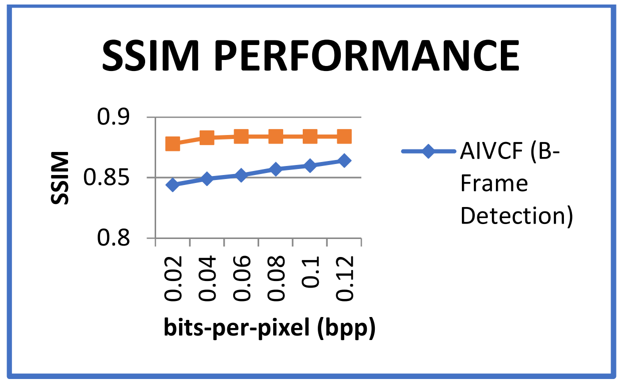

Figure 9: Rate-distortion analysis in terms of SSIM

As presented in Figure 9, bits-per-pixel rate is used for experimentation. Different rates of bits-per-pixel are provided in horizontal axis. With the given rate, SSIM is computed to ascertain video compression performance. Higher in SSIM value indicates less distortion and higher quality in compression. An important observation is that bits-per-pixel (rate) has its influence on SSIM. Another observation is that the optimized version of B-Frame prediction process used for video compression is found to have better performance over its un-optimized variant. When rate is 0.02 the proposed framework with B-Frame prediction process has achieved SSIM 0.844 while its optimized version that exploits dynamic bit assignment achieved SSIM 0.878. This trend is true with all rates with which experiments are made for deep learning based video compression. Therefore, it can be concluded that the optimized version of B-Frame prediction process shows significantly better performance over its un-optimized counterpart.

Journal of Theoretical and Applied Information Technology
15th May 2024. Vol.102. No 9
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

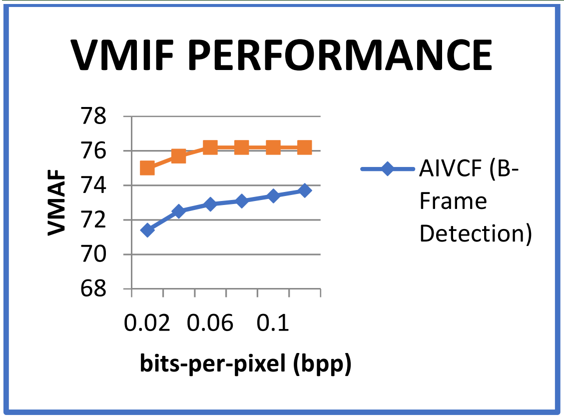

Figure 10: Rate-distortion analysis in terms of VMIF

As presented in Figure 10, bits-per-pixel rate is used for experimentation. Different rates of bits-per-pixel are provided in horizontal axis. With the given rate, VMIF is computed to ascertain video compression performance. Higher in VMIF value indicates less distortion and higher quality in compression. An important observation is that bits-per-pixel (rate) has its influence on VMIF. Another observation is that the optimized version of B-Frame prediction process used for video compression is found to have better performance over its un-optimized variant. When rate is 0.02 the proposed framework with B-Frame prediction process has achieved VMIF 71.4 while its optimized version that exploits dynamic bit assignment achieved VMIF 75. This trend is true with all rates with which experiments are made for deep learning based video compression. Therefore, it can be concluded that the optimized version of B-Frame prediction process shows significantly better performance over its un-optimized counterpart.

5.2 Performance Evaluation of P-Frame Detection Process

This section evaluates per performance of proposed learning based video compression using P-Frame detection process against standard codecs such as H.265 and H.264. Rate-distortion analysis is made with different performance metrics such as PSNR, SSIM and VMIF. Sampling of video clips is made using VTL dataset [40] where each clip is of 64x64 with 17 frames. There are 16 referencing frames and an I-frame in each clip. Experiments are made with the proposed framework and existing codecs aforementioned.

Table 6: PSNR performance comparison of P-Frame detection against H.264 and H.265

Bits-Per-Pixel	PSNR		
	AIVCF (P-Frame Detection)	H.264	H.265
0.1	20	0	0
0.15	26.5	0	0
0.2	28	0	0
0.25	28.3	24.5	0
0.3	28.5	27.8	25.8
0.35	28.6	31	28.3
0.4	28.7	33.5	31.5

As presented in Table 6, PSNR performance of proposed framework AIVCF with P-Frame detection is compared against H.264 and H.265. Rate-distortion analysis is made with different bits-per-pixel values.

Table 7: SSIM performance comparison of P-Frame detection against H.264 and H.265`

Bits-Per-Pixel	SSIM		
	AIVCF (P-Frame Detection)	H.264	H.265
0.1	0.5	0	0
0.15	0.82	0	0
0.2	0.83	0	0
0.25	0.84	0.75	0
0.3	0.85	0.87	0.78
0.35	0.86	0.93	0.88
0.4	0.87	0.95	0.92

As presented in Table 7, SSIM performance of proposed framework AIVCF with P-Frame detection is compared against H.264 and H.265. Rate-distortion analysis is made with different bits-per-pixel values.

Table 8: VMAF performance comparison of P-Frame detection against H.264 and H.265

	VMAF

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Bits-Per-Pixel	**AIVCF (P-Frame Detection)**	**H.264**	**H.265**
0.1	30	0	0
0.15	69	0	0
0.2	71	0	0
0.25	70	56	0
0.3	71	81	62
0.35	72	85	81
0.4	73	88	85

As presented in Table 8, VMAF performance of proposed framework VMAF with P-Frame detection is compared against H.264 and H.265. Rate-distortion analysis is made with different bits-per-pixel values.

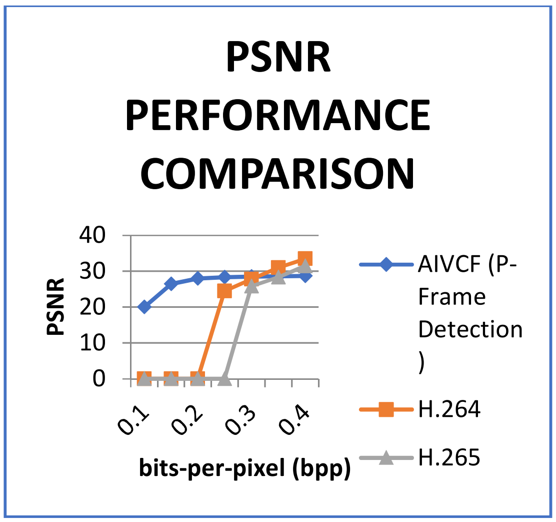

Figure 11: Performance comparison of P-Frame detection with existing codecs H.264 and H.265

As presented in Figure 11, the observations are made with different rates as given in horizontal axis. The perceived quality of video due to compression techniques is measured using PSNR as given in vertical axis. It is observed that the bits-per-pixel has its influence on PSNR. Each compression technique has shown different level of performance due to the underlying mechanisms. However, the proposed learning based approach using P-Frame detection has significant performance improvement over the conventional techniques. However, P-Frame detection process outperforms other techniques only at low bit rates. At higher bit rates, the P-Frame detection process has performance less than that of H.264 and H.265. The rationale behind this is that the proposed framework does not consider compression of residual information but focuses on motion estimation. Only the inter-frame prediction approach in the proposed framework has resulted in performance improvement.

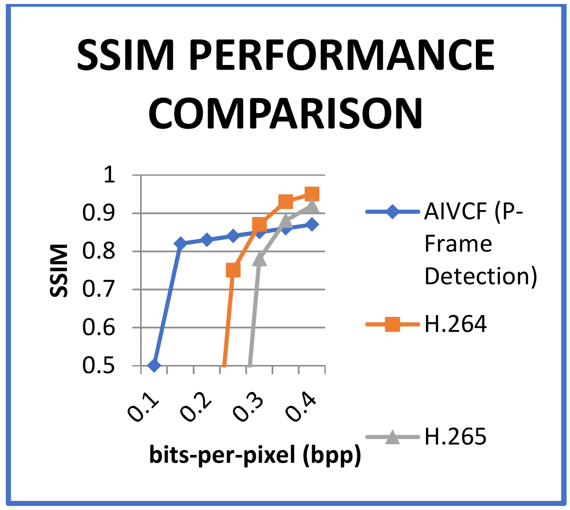

Figure 12: SSIM performance comparison of P-Frame detection with existing codecs H.264 and H.265

As presented in Figure 12, the observations are made with different rates as given in horizontal axis. The perceived quality of video due to compression techniques is measured using SSIM as given in vertical axis. It is observed that the bits-per-pixel has its influence on SSIM. Each compression technique has shown different level of performance due to the underlying mechanisms. However, the proposed learning based approach using P-Frame detection has significant performance improvement over the conventional techniques. However, P-Frame detection process outperforms other techniques only at low bit rates. At higher bit rates, the P-Frame detection process has performance less than that of H.264 and H.265. The rationale behind this is that the proposed framework does not consider compression of residual information but focuses on motion estimation. Only the inter-frame prediction approach in the proposed framework has resulted in performance improvement.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

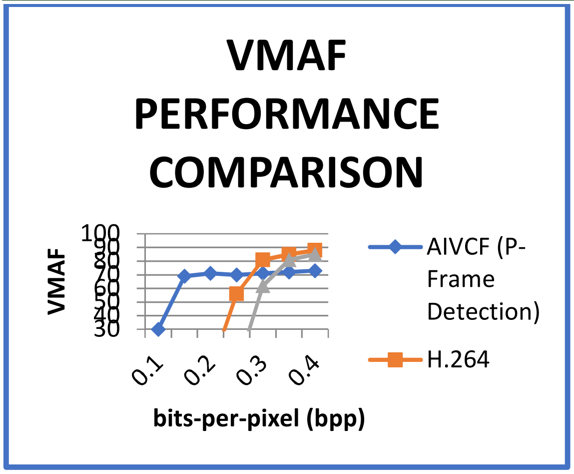

Figure 13: VMAF performance comparison of P-Frame detection with existing codecs H.264 and H.265

As presented in Figure 13, the observations are made with different rates as given in horizontal axis. The perceived quality of video due to compression techniques is measured using VMAF as given in vertical axis. It is observed that the bits-per-pixel has its influence on VMAF. Each compression technique has shown different level of performance due to the underlying mechanisms. However, the proposed learning based approach using P-Frame detection has significant performance improvement over the conventional techniques. However, P-Frame detection process outperforms other techniques only at low bit rates. At higher bit rates, the P-Frame detection process has performance less than that of H.264 and H.265. The rationale behind this is that the proposed framework does not consider compression of residual information but focuses on motion estimation. Only the inter-frame prediction approach in the proposed framework has resulted in performance improvement.

5.3 Performance Evaluation of B-Frame Detection Process

This section evaluates per performance of proposed learning based video compression using B-Frame detection process against standard codecs such as H.265 and H.264. Rate-distortion analysis is made with different performance metrics such as PSNR, SSIM and VMIF. Sampling of video clips is made using VTL dataset [40] where each clip is of 64x64 with 17 frames. There are 16 referencing frames and an I-frame in each clip. Experiments are made with the proposed framework and existing codecs aforementioned.

Table 9: PSNR performance comparison of B-Frame detection against H.264 and H.265

Bits-Per-Pixel	PSNR		
	AIVCF (B-Frame Detection)	**H.264**	**H.265**
0.15	0	0	0
0.2	23.5	0	0
0.25	27	23.5	0
0.3	28.2	26.9	26.2
0.35	28.4	29	29.3
0.4	28.6	31.1	31.8

As presented in Table 9, PSNR performance of proposed framework AIVCF with B-Frame detection is compared against H.264 and H.265. Rate-distortion analysis is made with different bits-per-pixel values.

Table 10: SSIM performance comparison of B-Frame detection against H.264 and H.265

Bits-Per-Pixel	SSIM		
	AIVCF (B-Frame Detection)	**H.264**	**H.265**
0.15	0	0	0
0.2	0.68	0	0
0.25	0.82	0.72	0
0.3	0.85	0.85	0.82
0.35	0.86	0.9	0.9
0.4	0.87	0.93	0.94

As presented in Table 10, SSIM performance of proposed framework AIVCF with B-Frame detection is compared against H.264 and H.265. Rate-distortion analysis is made with different bits-per-pixel values.

Table 11: VMAF performance comparison of B-Frame detection against H.264 and H.265

Bits-Per-Pixel	VMAF		
	AIVCF (B-Frame Detection)	H.264	H.265
0.15	0	0	0
0.2	50	0	0
0.25	75	58	0
0.3	76	75	72
0.35	77	83	84
0.4	78	85	86

As presented in Table 11, VMAF performance of proposed framework AIVCF with B-Frame detection is compared against H.264 and H.265. Rate-distortion analysis is made with different bits-per-pixel values.

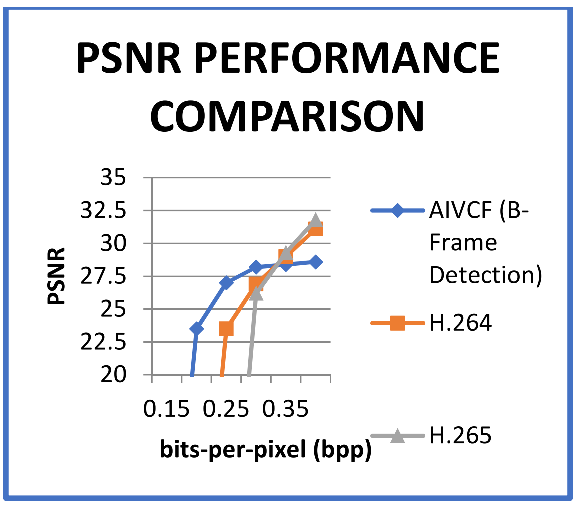

Figure 14: PSNR performance comparison of B-Frame detection with existing codecs H.264 and H.265

As presented in Figure 14, the observations are made with different rates as given in horizontal axis. The perceived quality of video due to compression techniques is measured using PSNR as given in vertical axis. It is observed that the bits-per-pixel has its influence on PSNR. Each compression technique has shown different level of performance due to the underlying mechanisms. However, the proposed learning based approach using B-Frame detection has significant performance improvement over the conventional techniques. However, B-Frame detection process outperforms other techniques only at low bit rates. At higher bit rates, the B-Frame detection process has performance less than that of H.264 and H.265. The rationale behind this is that the proposed framework does not consider compression of residual information but focuses on motion estimation. Only the inter-frame prediction approach in the proposed framework has resulted in performance improvement.

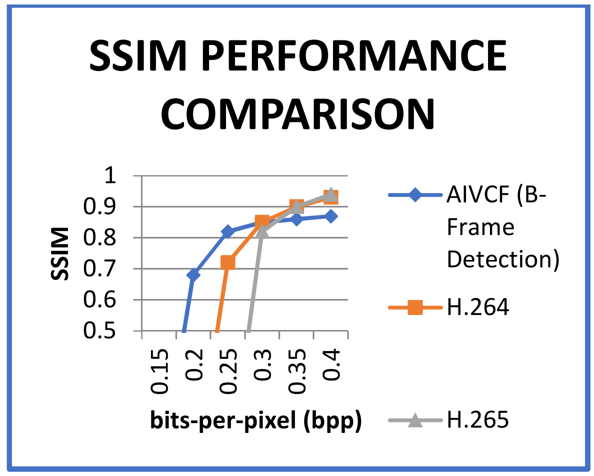

Figure 15: SSIM performance comparison of B-Frame detection with existing codecs H.264 and H.265

As presented in Figure 15, the observations are made with different rates as given in horizontal axis. The perceived quality of video due to compression techniques is measured using SSIM as given in vertical axis. It is observed that the bits-per-pixel has its influence on SSIM. Each compression technique has shown different level of performance due to the underlying mechanisms. However, the proposed learning based approach using B-Frame detection has significant performance improvement over the conventional techniques. However, B-Frame detection process outperforms other techniques only at low bit rates. At higher bit rates, the B-Frame detection process has performance less than that of H.264 and H.265. The rationale behind this is that the proposed framework does not consider compression of residual information but focuses on motion estimation. Only the inter-frame prediction approach in the proposed framework has resulted in performance improvement.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

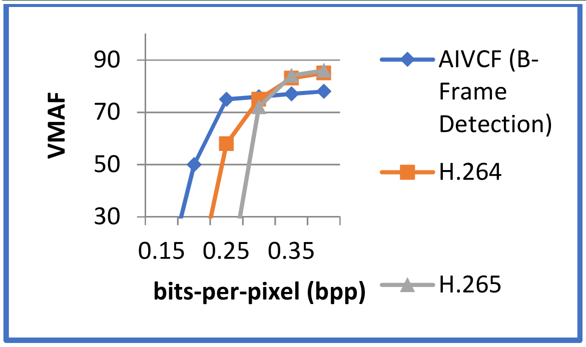

Figure 16: VMAF performance comparison of B-Frame detection with existing codecs H.264 and H.265

As presented in Figure 16, the observations are made with different rates as given in horizontal axis. The perceived quality of video due to compression techniques is measured using VMAF as given in vertical axis. It is observed that the bits-per-pixel has its influence on VMAF. Each compression technique has shown different level of performance due to the underlying mechanisms. However, the proposed learning based approach using B-Frame detection has significant performance improvement over the conventional techniques. However, B-Frame detection process outperforms other techniques only at low bit rates. At higher bit rates, the B-Frame detection process has performance less than that of H.264 and H.265. The rationale behind this is that the proposed framework does not consider compression of residual information but focuses on motion estimation. Only the inter-frame prediction approach in the proposed framework has resulted in performance improvement.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a framework named Artificial Intelligence (AI) enabled Video Compression Framework (AIVCF) which exploits the traditional classical architecture and combines it with a deep learning model for non-linear data representation. This framework has ability to have joint optimization of underlying components. Convolutional Neural Network (CNN) is used to reconstruct current frames by getting motion information through a process known as optical flow estimation. The information of given video is compressed using deep learning models in auto-encoder fashion. The framework strikes balance between quality and compression ability. An algorithm named Deep Joint Optimization for Video Compression (DJO-VC) is proposed to realize the AIVCF. The proposed framework is evaluated with empirical study. The experimental results, in terms of PSNR and SSIM revealed that the proposed framework outperforms existing models such as H.264. However, the proposed framework AIVCF showed better performance only when there are low bit rates. When bit rate is high, its performance is not better than the conventional methods. he rationale behind this is that the proposed framework does not consider compression of residual information but focuses on motion estimation. Only the inter-frame prediction approach in the proposed framework has resulted in performance improvement. In future work, we intend to improve the framework to overcome this drawback besides considering other deep learning approaches.

REFERENCES

[1] Tekalp, A. M., Covell, M., Timofte, R., & Dong, C. (2021). Editorial: Introduction to the Issue on Deep Learning for Image/Video Restoration and Compression. IEEE Journal of Selected Topics in Signal Processing, 15(2), p157–161.

[2] Ma, Siwei; Zhang, Xinfeng; Jia, Chuanmin; Zhao, Zhenghui; Wang, Shiqi; Wanga, Shanshe (2019). Image and Video Compression with Neural Networks: A Review. IEEE Transactions on Circuits and Systems for Video Technology, p1–16.

[3] Yang, R., Mentzer, F., Van Gool, L., & Timofte, R. (2021). Learning for Video Compression with Recurrent Auto-Encoder and Recurrent Probability Model. IEEE Journal of Selected Topics in Signal Processing, 15(2), p388–401.

[4] Lu, Guo; Zhang, Xiaoyun; Ouyang, Wanli; Chen, Li; Gao, Zhiyong; Xu, Dong (2020). An End-to-End Learning Framework for Video Compression. IEEE Transactions on Pattern Analysis and Machine Intelligence, p1–18.

[5] Chen, Zhuo; Fan, Kui; Wang, Shiqi; Duan, Ling-Yu; Lin, Weisi; Kot, Alex (2019). [ACM Press the 27th ACM International Conference - Nice, France (2019.10.21-2019.10.25)] Proceedings of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the 27th ACM International Conference on Multimedia - MM '19 - Lossy Intermediate Deep Learning Feature Compression and Evaluation. , p2414–2422.

[6] Liu, Dong; Chen, Zhenzhong; Liu, Shan; Wu, Feng (2019). Deep Learning-Based Technology in Responses to the Joint Call for Proposals on Video Compression with Capability beyond HEVC. IEEE Transactions on Circuits and Systems for Video Technology, p1–14.

[7] Xu, D., Lu, G., Yang, R., & Timofte, R. (2020). Learned image and video compression with deep neural networks. 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP). P1-3

[8] Westland, Natasha; Dias, Andre Seixas; Mrak, Marta (2019). [IEEE 2019 IEEE International Conference on Image Processing (ICIP) - Taipei, Taiwan (2019.9.22-2019.9.25)] 2019 IEEE International Conference on Image Processing (ICIP) - Decision Trees for Complexity Reduction in Video Compression. , p2666–2670.

[9] Friedland, Gerald; Jia, Rouxi; Wang, Jingkang; Li, Bo; Mundhenk, Nathan (2020). [IEEE 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) - Shenzhen, Guangdong, China (2020.8.6-2020.8.8)] 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) - On the Impact of Perceptual Compression on Deep Learning. , p219–224.

[10] Pessoa, Jorge; Aidos, Helena; Tomas, Pedro; Figueiredo, Mario A. T. (2020). [IEEE 2020 IEEE Workshop on Signal Processing Systems (SiPS) - Coimbra, Portugal (2020.10.20-2020.10.22)] 2020 IEEE Workshop on Signal Processing Systems (SiPS) - End-to-End Learning of Video Compression using Spatio-Temporal Autoencoders. , p1–6.

[11] Poyser, M., Atapour-Abarghouei, A., & Breckon, T. P. (2021). On the Impact of Lossy Image and Video Compression on the Performance of Deep Convolutional Neural Network Architectures. 2020 25th International Conference on Pattern Recognition (ICPR). P1-8

[12] Valenzise, Giuseppe; Purica, Andrei; Hulusic, Vedad; Cagnazzo, Marco (2018). [IEEE 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP) - Vancouver, BC, Canada (2018.8.29-2018.8.31)] 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP) - Quality Assessment of Deep-Learning-Based Image Compression. , p1–6.

[13] Zhang, F., Ma, D., Feng, C., & Bull, D. R. (2021). Video Compression with CNN-based Post Processing. IEEE MultiMedia, p1–11.

[14] Chen, Ziqian; Wang, Shiqi; Wu, Dapeng Oliver; Huang, Tiejun; Duan, Ling-Yu (2018). [ACM Press 2018 ACM Multimedia Conference - Seoul, Republic of Korea (2018.10.22-2018.10.26)] 2018 ACM Multimedia Conference on Multimedia Conference - MM '18 - From Data to Knowledge. , p1625–1633.

[15] Liu, Huanhua; Zhang, Yun; Zhang, Huan; Fan, Chunling; Kwong, Sam; Kuo, C.-C. Jay; Fan, Xiaoping (2019). Deep Learning based Picture-Wise Just Noticeable Distortion Prediction Model for Image Compression. IEEE Transactions on Image Processing, p1–16.

[16] Birman, Raz; Segal, Yoram; Hadar, Ofer (2020). Overview of Research in the field of Video Compression using Deep Neural Networks. Multimedia Tools and Applications, p1-24.

[17] Nagaraj, P.; Rao. J, Surendra; Muneeswaran, V.; Sudheer Kumar, A.; Muthamil sudar, K. (2020). [IEEE 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) - Madurai, India (2020.5.13-2020.5.15)] 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) - Competent Ultra Data Compression By Enhanced Features Excerption Using Deep Learning Techniques. , p1061–1066.

[18] Krishnaraj, N.; Elhoseny, Mohamed; Thenmozhi, M.; Selim, Mahmoud M.; Shankar, K. (2019). Deep learning model for real-time image compression in Internet of Underwater Things (IoUT). Journal of Real-Time Image Processing, p1-15.

[19] Das, Nilaksh; Shanbhogue, Madhuri; Chen, Shang-Tse; Hohman, Fred; Li, Siwei; Chen, Li; Kounavis, Michael E.; Chau, Duen Horng (2018). [ACM Press the 24th ACM

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SIGKDD International Conference - London, United Kingdom (2018.08.19-2018.08.23)] Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18 - SHIELD. , p196–204.

[20] Chen, Ziqian; Duan, Ling-Yu; Wang, Shiqi; Lou, Yihang; Huang, Tiejun; Wu, Dapeng Oliver; Gao, Wen (2019). Towards Knowledge as a Service over Networks: A Deep Learning Model Communication Paradigm. IEEE Journal on Selected Areas in Communications, p1–15.

[21] Prakash, Aaditya; Moran, Nick; Garber, Solomon; Dilillo, Antonella; Storer, James (2017). [IEEE 2017 Data Compression Conference (DCC) - Snowbird, UT, USA (2017.4.4-2017.4.7)] 2017 Data Compression Conference (DCC) - Semantic Perceptual Image Compression Using Deep Convolution Networks. , p250–259.

[22] Wiedemann, Simon; Wiegand, Thomas; Marpe, Detlev; Samek, Wojciech; Kirchhoffer, Heiner; Matlage, Stefan; Haase, Paul; Marban, Arturo; Marinc, Talmaj; Neumann, David; Nguyen, Tung; Schwarz, Heiko (2020). DeepCABAC: A universal compression algorithm for deep neural networks. IEEE Journal of Selected Topics in Signal Processing, p1–16.

[23] Vega, Maria Torres; Mocanu, Decebal Constantin; Famaey, Jeroen; Stavrou, Stavros; Liotta, Antonio (2017). Deep Learning for Quality Assessment in Live Video Streaming. IEEE Signal Processing Letters, 24(6), p736–740.

[24] Duan, Lingyu; Liu, Jiaying; Yang, Wenhan; Huang, Tiejun; Gao, Wen (2020). Video Coding for Machines: A Paradigm of Collaborative Compression and Intelligent Analytics. IEEE Transactions on Image Processing, 29, p8680–8695.

[25] Chen, Zhuo; Fan, Kui; Wang, Shiqi; Duan, Lingyu; Lin, Weisi; Kot, Alex C. (2019). Intermediate Deep Feature Compression: Toward Intelligent Sensing. IEEE Transactions on Image Processing, p1–14.

[26] Kuanar, Shiba; Conly, Christopher; Rao, K. R. (2018). [IEEE 2018 Picture Coding Symposium (PCS) - San Francisco, CA, USA (2018.6.24-2018.6.27)] 2018 Picture Coding Symposium (PCS) - Deep Learning Based HEVC In-Loop Filtering for Decoder Quality Enhancement. , p164–168.

[27] Li, Binglin; Akbari, Mohammad; Liang, Jie; Wang, Yang (2020). [IEEE 2020 Data Compression Conference (DCC) - Snowbird, UT, USA (2020.3.24-2020.3.27)] 2020 Data Compression Conference (DCC) - Deep Learning-Based Image Compression with Trellis Coded Quantization. , p13–22.

[28] Zhang, Zheng-Teng; Yeh, Chia-Hung; Kang, Li-Wei; Lin, Min-Hui (2017). [IEEE 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) - Kuala Lumpur (2017.12.12-2017.12.15)] 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) - Efficient CTU-based intra frame coding for HEVC based on deep learning. , p661–664.

[29] Sinha, Abhishek Kumar; Mishra, Deepak (2020). [IEEE 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) - Kharagpur, India (2020.7.1-2020.7.3)] 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) - T3D-Y Codec: A Video Compression Framework using Temporal 3-D CNN Encoder and Y-Style CNN Decoder. , p1–7.

[30] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, R. Sukthankar, Variable rate image compression with recurrent neural networks, International Conference on Learning Representations (ICLR) (2015).

[31] T. Raiko, M. Berglund, G. Alain, L. Dinh, Techniques for learning binary stochastic feedforward neural networks, arXiv preprint arXiv:1406.2989 (2014).

[32] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, International Conference on Learning Representations (ICLR) (2016).

[33] M. Li, W. Zuo, S. Gu, D. Zhao, D. Zhang, Learning convolutional networks for content-weighted image compression, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2018).

[34] I. E. G. Richardson, The H.264 advanced video compression standard, 2.0 ed., Wiley,

[35] Chichester, West Sussex, 2010.
Netflix, VMAF: the journey continues,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

available at https://medium.com/ netflix-techblog/vmaf-the-journey-continues-44b51ee9ed12, 2018.

[36] Z. Wang, A. C. Bovik, H. Rahim Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions On Image Processing (TIP) 13 (2004).

[37] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, “Video enhancement with task-oriented flow,” Int. J. Comput. Vis., vol. 127, no. 8, pp. 1106–1125, 2019.

[38] S. Wan, M.-Z. Wang, H. Gong, C.-Y. Zou, Y.-Z. Ma, J.-Y. Huo, Y.-F. Yu, and Y. Liu, “CE10: Integrated in-loop filter based on CNN (Tests 2.1, 2.2 and 2.3),” in the JVET meeting, no. JVET-O0079. ITU-T, ISO/IEC, 2019.

[39] Y. Wang, Z. Chen, Y. Li, L. Zhao, S. Liu, and X. Li, “CE13: Dense residual convolutional neural network based in-loop filter (CE13-2.2 and CE13-2.3),” in the JVET meeting, no. JVET-N0254. ITU-T, ISO/IEC, 2019.

[40] A. S. University, Video trace library YUV video sequences, available at http://trace.kom.aau.dk/yuv/index.html, 2000.