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

A HYBRID APPROACH FOR OPTIMIZED VIDEO COMPRESSION USING DEEP RECURRENT AUTO ENCODERS (DRAE) TECHNIQUE

Mr. KOMMERLA SIVA KUMAR [1], Dr. P. BINDHU MADHAVI [2], Dr.K. JANAKI [3]

[1] Assistant Professor, CSE Department, R.V.R & J.C College of Engineering, Andhra Pradesh.
[2] Professor & HOD, Department of AI & ML, The Oxford College of Engineering, Karnataka.
[3] Associate Professor, Department of CSE-AI, Faculty of Engineering, Jain Deemed to be University, Karnataka.

kommerlasivakumar@gmail.com. bindhu.prathipati@gmail.com. karur.janaki@gmail.com

ABSTRACT

Presently, the data traffic is increasing for video conferencing, online education, gaming and watching videos on Netflix, Amazon Prime, YouTube and other OTT platforms. And, the service users are always demanding high definition and high-quality video facilities day by day. However, in order to transmit video data across the Internet's constrained bandwidth effectively, video compression is a necessary task. In last few decades, various video compression algorithms, such as non-learning and learning were standardized. But still some improvements are needed for effective video related services. We propose a deep learning based Deep Recurrent Auto Encoders (DRAE) approach which contain various modules for implementing an efficient video compression technique. The experimental outcome shows our model achieves state-of-the-art learned video compression performance in terms of both PSNR and MS-SSIM.
Keywords: *Video, Compression, Deep Neural Networks, Recurrent Auto Encoders.*

1. INTRODUCTION

Nowadays, video content contributes to more than 80% internet traffic, and the percentage is expected to increase even further. Therefore, it is critical to build an efficient video compression system and generate higher quality frames at given bandwidth budget.

Modern digital video looks as impressive as it does is because of the sheer amount of information digital cameras can capture. This informational data is what creates the crisp details and vivid saturation of modern video. The problem is that it takes a ton of data to capture these beautiful videos. So much data that you may find your computers and hard drives filling up quickly due to the hefty storage demands of your video, not to mention the extremely long wait times for uploading or sharing these videos to online platforms. Luckily, compression offers the solution of taking the vast amounts of data that cameras generate and interpreting it in a way that is more efficient, creating new files that are only a fraction of the file size! The only way you'll be able to share, upload, stream and store all of your great video content with any regularity is by compressing it. The trick is to know "good" compression from "bad" compression. The objective of "good" compression is to minimize the file size as much as possible with the least amount of image quality reduction by removing things like redundant or non-functional data from your video file.

Internet traffic has recently been dominated by video-related applications including video on demand (VOD), live streaming, and ultra-low latency real-time communications.

Due to the ever-increasing demands for resolution ([1] and [2]), and fidelity, more effective video compression is required for content transmission and storage, and therefore for successful implementation of networked video services ([3], [4]). Video compression systems develop suitable techniques to reduce end-to-end reconstruction distortion within a specific bit rate budget. In the conventional sense, this is a rate-distortion (R-D) optimization problem. An abundance of well-liked standards and recommendation specifications (e.g., ISO/IEC MPEG series [5]– [7], ITU-T H.26x series [8]– [11], AVS series [12]) were produced as a result of the majority of previous effort being focused on development and standardization of video coding tools for optimized R–D performance. These tools include intra prediction/inter prediction, transform, and entropy coding. All of these standards have been widely adopted in market, allowing organizations and consumers to access innovative and high-performing services. They are now widely utilized for telemedicine, distance learning, video conferencing, broadcasting, e-commerce, online gaming, and short-form video platforms, covering all major video situations from VOD to live streaming to ultralow latency interactive real-time communications. Convolutional and recurrent neural networks may be used by traditional codecs, particularly those that utilize deep neural networks, as part of proposed strategy to improve their performance.

Researchers working on these fields can use our DRAE model as a key point for their future research.

2. RELATED WORK

In [13,14,15,16], The proposed learning-based end-to-end deep video compression scheme outperforms h.264 [17] and h.265 [18] in experimental testing, because it consists of the different modules optical flow net, motion vector encoder-decoder net, motion compensation net, and residual encoder-decoder net.

In [19], propose a two-step recurrent learning video compression (RLVC) model. Utilizing a recurrent auto encoder model in the first phase will allow you to take advantage of temporal correlation between the vast arrays of video frames. The second stage involves repeatedly estimating the temporally conditional probability mass function of latent representations using a recurrent probability model. To assess performance of proposed model against currently used learnt video compression models and to confirm the efficiency of each recurrent component in their framework, improved experiment results should then be obtained.

In [20], suggest a three-hierarchical-layer, a hierarchical-quality, and a recurrent-enhancement learning video compression technique. All frames are compressed in top layer utilizing the best image compression technique. For further, high-quality compression, use a bi-directional deep compression network at second layer. The least efficient additional compression will be achieved in the third layer by using a single motion deep motion compression network. Create a weighted recurrent enhancement network from the decoder side with inputs for compressed frames, bitrate data, and quality data for multi-frame enhancement. The HLVC model was shown to be effective through experiments.

In [21], to eliminate perceptual redundancy in HD video and improve video quality while also speeding up video encoding, we offer a unique saliency-based HD video compression technique that is on top of HEVC video compression standard. Additionally, an enhanced video - saliency - model is proposed, which utilizes different deep learning techniques to obtain superior performance.

In [22], propose, perceptual quality of final reconstructed information can be greatly improved over [23].

In [24], A learning-based video compression model is proposed and is used for inter frame coding. A motion predictor-net is used to anticipate motion vectors for the target frame and to deliver the differential motion vectors while reducing the transmitted motion information. Additionally, a refine-net working with the residual codec was built. The evaluation findings show that this model is quite competitive in terms of coding performance when compared to existing learning-based video codecs.

In [25], To suggest a feature-space video coding network, perform the operations motion-estimation, motion-compression, motion compensation, and residual-compression in feature space (FVC). According to experimental findings, this framework performs at the cutting edge on four datasets, which are HEVC, UVG, VTL, and MCL-JCV.

In [26], propose Deep-coder based video compression framework. For predictive and residual signals, we use different CNN nets. The quantized feature maps (fMaps) are encoded into a binary stream using scalar quantization and Huffman coding. In this study, we employ the fixed 32x32 block to illustrate our ideas, and

Journal of Theoretical and Applied Information Technology
31st October 2022. Vol.100. No 20

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

compare results to well-known H.264/AVC video coding standard.

In [27], a technique for video compression that performs better than H.264/AVC video coding. The proposed neural network architecture is composed of multiple layers and is split into two parts: an encoder and a decoder. In order to be used as any other compression encoding and decoding modules, encoder and decoder are separated during testing. The two components of the model are trained simultaneously. The entire model was created to attempt and take advantage of spatial and temporal relationships between video frames. In [28], the idea of the Pixel Motion CNN (PMCNN), which incorporates hybrid prediction networks and motion extension. In order to efficiently perform predictive coding within the learning network, PMCNN can simulate spatiotemporal coherence. Results from experiments show how effective the suggested plan is. Despite not using entropy coding or complicated settings, we nonetheless demonstrate better performance than MPEG-2 and H.264 codec.

In [29], various methods of video compression were discussed, and deep learning techniques were used to advance the subject of research. In [30], Standard deep learning algorithms for video compression are still required. In [31,33], discussed the utilization, advantages and limitations of neural network models.

3. PROPOSED METHODOLOGY:

The main objectives of our proposed DRAE are:

To lessen the amount of storage the video takes up.

To decrease the amount of time that video transfers take.

To improve the compression ratio and thus video quality.

And, the proposed DRAE approach is mainly compared with our previous DCNN [32] approach. For motion estimation & motion compensation, we use same procedure in [32] and for motion compression & residual compression, we develop motion compression recurrent encoder – decoder and residual encoder – decoder, shown in Fig: 1. The proposed DRAE approach getting better results than compared with our previous DCNN [data2 - 32] approach and other data 1[19] approach which are shown in results section. The following notations are used in our proposed DRAE approach.

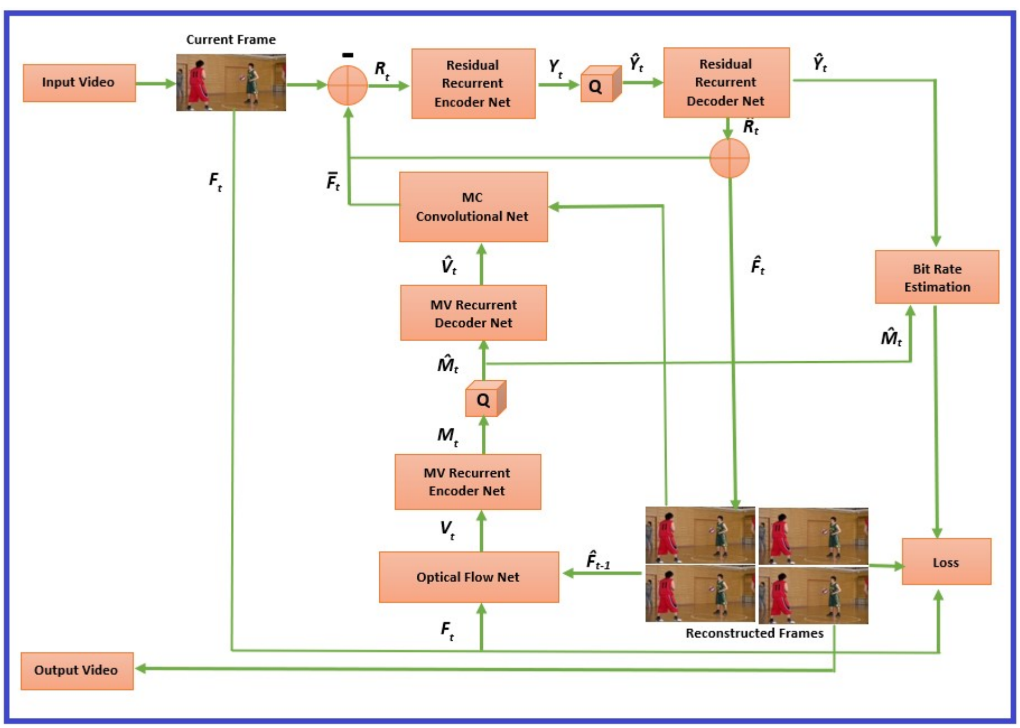

Fig _1: Our Proposed Deep Recurrent Auto Encoder (DRAE) Approach

Table_1: Corresponding Variable Names And Notations Used In Our Proposed Approach.

Definition	Notation
Reference frame	$\hat{F}_{t-1}$
Current frame	F_t
Estimated motion	V_t
Latent representation of motion	M_t
Quantized latent representation of motion	$\hat{M}_t$
Compressed motion	$\hat{V}_t$
Motion compensated reference frame	$\overline{F}_t$
Residual	R_t
Latent representation of residual	Y_t
latent Quantized latent representation of residual	$\hat{Y}_t$
Compressed residual	$\hat{R}_t$
Compressed frame	$\hat{F}_t$

Similarly, this section provides an overview of our Deep Recurrent Auto Encoders (DRAE) approach, which is seen in Fig: 1.
Similar to that, we employ a few notations for this implementation, and each notation's definition is shown in Table 1. And the following is a description of each module in our suggested strategy.

3.1 Motion Estimation:

We employ the motion estimation network, which Tensor Flow built based on a PyTorch implementation of pyramid network [32], to calculate motion data between current frame and previous compressed frame. We employ 5-level pyramid network and adhere to the parameters outlined in [32]. Five convolutional layers in each level, each having a 7x7 kernel and corresponding filter numbers of 32, 64, 32, 16, and 2. The estimated motion V_t is produced by the optical flow estimation network, as shown in Fig: 1.

3.2 Motion Compression:

In this, the proposed DRAE, which is depicted in the following Fig: 2, compresses the estimated motion. Recurrent cells are present in both the encoder and decoder of the proposed DRAE. Similarly, Fig: 2 provide an illustration of the DRAE network's architecture. In the DRAE encoder, we employ four convolutional layers with a 2 x down sampling and a GDN activation function. We place a ConvLSTM cell in the center of the four convolutional layers to create the recurrent structure. As a result, the ConvLSTM's hidden states allow information from earlier frames to flow into encoder network of current frame. In light of both current and past inputs, the suggested DRAE creates latent representation. The recurrent decoder in the DRAE also features a ConvLSTM cell in the center of the four 2 x up-sampling convolutional layers with IGDN, and as a result, it reconstructs F_t using both the most recent and earlier latent representations. Our DRAE approach may utilize the information in a wide range of frames since all prior frames can be thought of as reference frames for compressing present frame.

3.3 Motion Compensation:

In this case, reference frame is initially warped by compressed motion before being fed as inputs to the motion compensation network to produce the motion-corrected frame. The motion compensation network of our method is depicted in Fig. 3, where each layer has a 3-by-3 filter. Each layer's filter number is set to 64, with the exception of the final layer, whose filter number is 3. ↑ 2 and ↓ 2 indicate up- and down-sampling with the stride of 2, respectively, and ⊕ denotes the element-wise addition.

3.4 Residual Compression:

The difference between current raw frame and corrected reference frame following motion compensation is the residual. We compress residual in our DRAE using the same technique as the motion compression in fig 2. The sole distinction is that for residual compression, we utilize filters with a size of 5x5 in the auto-encoder rather than 3x3 for motion compression. The explanation is that residual requires more bitrate and includes more information than motion, and a larger filter size enhances auto encoder's capacity to represent data. The residual can then be added to the corrected reference frame to produce the reconstructed compressed frame.

3.5 Bit Rate Estimation:

During testing phase, quantized motion data from Step 2 and residual data from Step 4 are encoded as bits and transferred to the decoder.

3.6 Frame reconstruction:

The reconstructed frame is obtained by combining Steps 3 and 4.

3.7 Quantization:

The essential premise underlying quantization is that by converting the weights and inputs into integer types, we may reduce memory use and speed up computations on certain hardware. It is necessary to use latent representations like residual and motion representation.

4. TRAINING AND TESTING

In [19, 32], provides the training (viemo) and testing (UVG & JCT-VC) datasets. Our proposed DRAE technique trained the around 29k frames or images (viemo) and tested 9k frames or images in Ultra-Video-Group dataset (UVG) and Joint Collaborative Team-Video Coding dataset (JCT-VC). In this paper the trained data sets link and tested data sets link are given in reference section.

5. RESULTS AND PERFORMNACE ANALYSIS

The Average is calculated by PSNR and bpp in Tables 2, 3, 4 and 5. And drawn the corresponding (We have taken the best of any four values from the Tables) Fig. 4(a) - Table 2, 4(b) -Table 3, 4(c)-Table 4 and 4(d) - Table 5, compared with existing methods like data1 & data2 and produce better average psnr and average bpp values. The Average is calculated by MS-SSIM and bpp in Tables 6, 7, 8 and 9. And drawn the corresponding Fig. 5(a) - Table 6, 5(b)-Table 7, 5(c) - Table 8, and 5(d) – Table 9, compared with existing methods like data1 & data2 and produce better average psnr and average bpp values. And also, the proposed approach (in terms of PSNR metric) proves to reduce the error rate with the following MSE Table 10, 11, 12 and 13 (mean square error rate) for the tested data sets are also given. The compression ratio and transmission time of the of the input/original videos and output/compressed videos are also one of the important objectives of our proposed approach. Our proposed DRAE approach also getting better compression ration than compared with existing video compression approaches. The transmission time of output video is obviously reducing than input video.

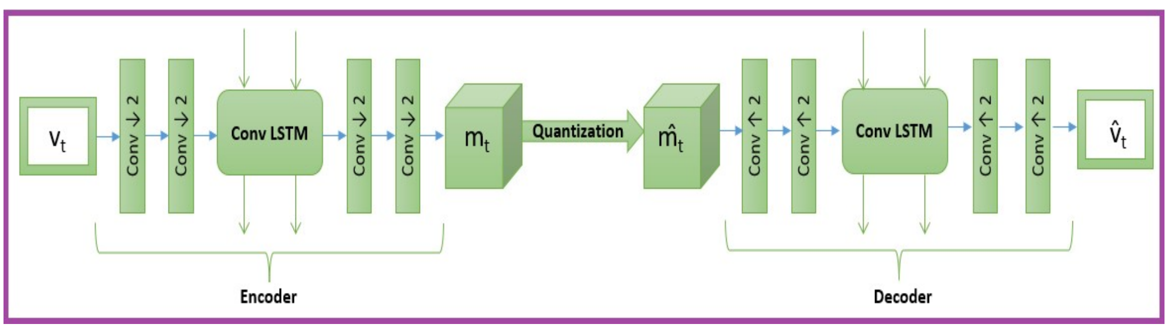

Fig_2: The Motion compression Network.

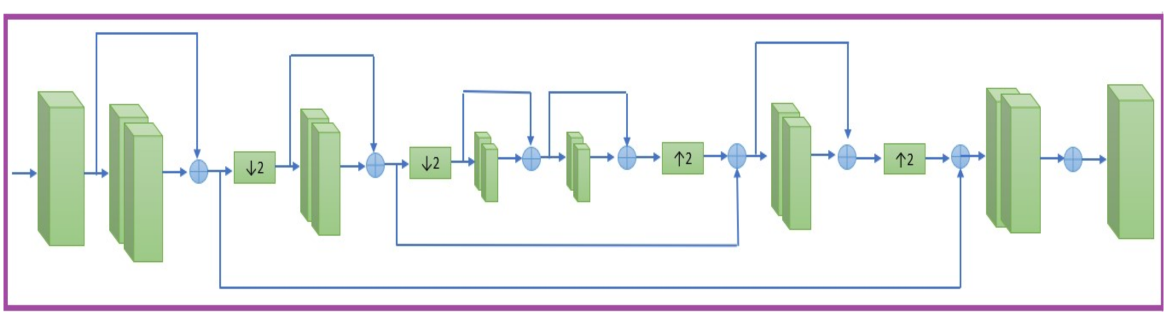

Fig_3: The Motion Compensation Network

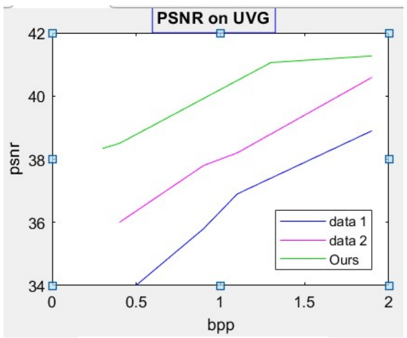

Fig. 4(A): PSNR On UVG

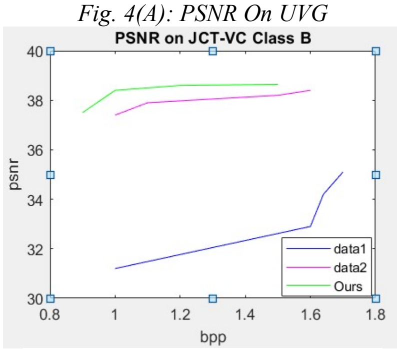

Fig. 4(B): PSNR On JCT-VC Class B

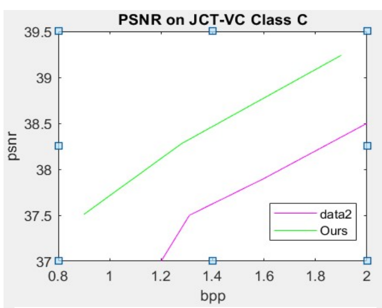

Fig. 4(C): PSNR On JCT-VC Class C

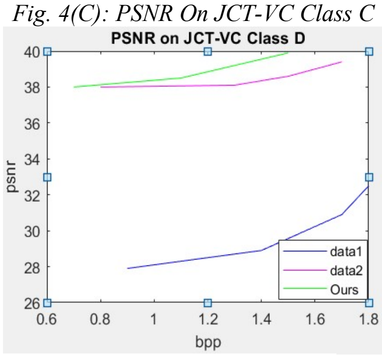

Fig. 4(D): PSNR On JCT-VC Class D

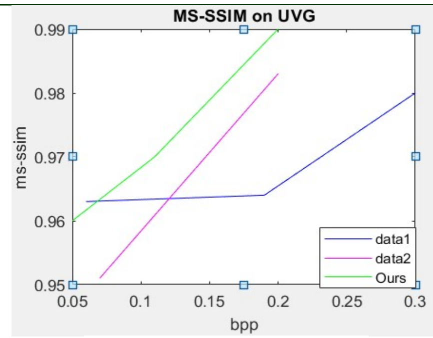

Fig. 5(A): MS-SSIM On UVG

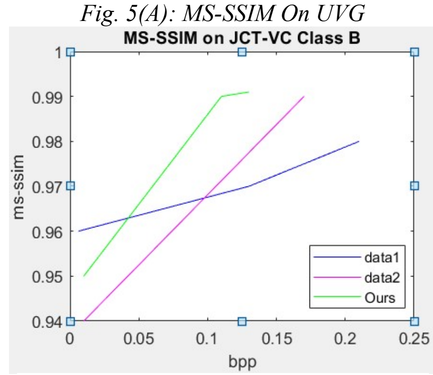

Fig. 5(b): MS-SSIM on JCT-VC Class B

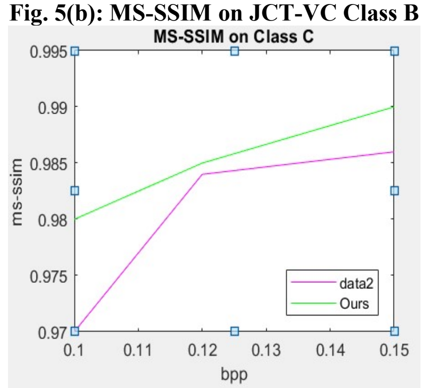

Fig. 5(C): MS-SSIM On JCT-VC Class C

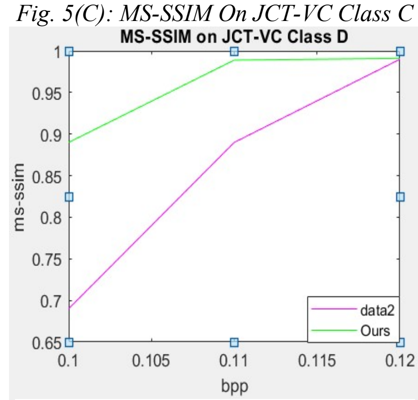

Fig. 5(D): MS-SSIM On JCT-VC Class D

Similarly, the proposed approach (in terms of MS-SSIM metric) proves to reduce error rate with following MSE Table 14, 15, 16 and 17 (mean square error rate) for tested data sets are

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

also given. Regarding figures our DRAE approach is compared with data 1(19) and data 2(32) except Fig 4(c) and 5(c) are compared only with data 2(32). Based on the outcomes and functionality, this study serves as a bridge for academics working in a variety of areas, including deep model creation, computer vision, and video compression. And the basic goals of reducing the amount of storage needed for video, the amount of time video takes to transfer, error rate, and increasing video quality with a better compression ratio are all met by our proposed DRAE technique.

Table 2: PSNR on UVG

Data Set	Video	Data 1(19)		Data 2(32)		Proposed DRAE	
		Avg PSNR	Avg bpp	Avg PSNR	Avg bpp	Avg PSNR	Avg bpp
UVG	Beauty	34.01	0.5	40.59	0.4	**41.06**	0.42
	Bosphorus	35.8	0.95	38.2	0.94	**38.5**	0.83
	HoneyBee	36.9	1.98	36.46	1.99	**36.49**	1.98
	Jockey	-	-	37.83	1.05	**41.27**	0.35
	ReadySetGo	-	-	37.04	1.8	**38.34**	1.31
	ShakeNDry	-	-	36.05	2	**36.23**	1.91
	YachtRide	38.9	1.15	38.05	1.12	**38.14**	1.1

Table 3: PSNR on JCT-VC Class B

Data Set	Video	Data 1(19)		Data 2(32)		Proposed DRAE	
		Avg PSNR	Avg bpp	Avg PSNR	Avg bpp	Avg PSNR	Avg bpp
JCT-VC Class B	BasketballDrive	31.2	1.05	38.26	1.02	**38.6**	0.93
	BQTerrace	32.9	1.6	38.22	1.53	**38.64**	1.22
	Cactus	34.2	1.64	38.41	1.63	37.53	1.57
	Kimono	-	-	37.94	1.18	**38.43**	1.08
	ParkScene	35.1	1.73	37.4	1.568	**37.48**	1.52

Table 4: PSNR on JCT-VC Class C

Data Set	Video	Data 1(19)		Data 2(32)		Proposed DRAE	
		Avg PSNR	Avg bpp	Avg PSNR	Avg bpp	Avg PSNR	Avg bpp
JCT-VC Class C	BasketballDrill	-	-	37.53	1.31	**37.57**	1.28
	BQMall	-	-	38.52	1.27	**39.24**	0.98
	PartyScene	-	-	37.09	2.34	**37.51**	1.9
	RaceHorses	-	-	37.91	1.6	**38.28**	1.27

Table 5: PSNR on JCT-VC Class D

Data Set	Video	Data 1(19)		Data 2(32)		Proposed DRAE	
		Avg PSNR	Avg bpp	Avg PSNR	Avg bpp	Avg PSNR	Avg bpp
JCT-VC Class D	BasketballPass	27.9	0.9	39.41	0.86	**39.92**	0.73
	BlowingBubbles	28.95	1.7	38.01	1.51	**38.02**	1.51
	BQSquare	30.99	1.82	38.65	1.76	**38.82**	1.51

	RaceHorses	32.5	1.41	38.15	1.37	**38.5**	1.18

Table 6: MS-SSIM on UVG

Data Set	Video	Data 1(19)		Data 2(32)		Proposed DRAE	
		Avg MS-SSIM	Avg bpp	Avg MS-SSIM	Avg bpp	Avg MS-SSIM	Avg bpp
UVG	Beauty	0.964	0.06	0.951	0.07	**0.96**	0.05
	Bosphorus	0.963	0.13	0.983	0.1	**0.985**	0.1
	HoneyBee	0.98	0.19	0.913	0.14	**0.991**	0.1
	Jockey	-	-	0.888	0.11	0.88	0.11
	ReadySetGo	-	-	0.981	0.12	**0.99**	0.12
	ShakeNDry	-	-	0.983	0.15	0.97	0.2
	YachtRide	0.98	0.36	0.95	0.1	**0.96**	0.1

Table 7: MS-SSIM on JCT-VC Class B

Data Set	Video	Data 1(19)		Data 2(32)		Proposed DRAE	
		Avg MS-SSIM	Avg bpp	Avg MS-SSIM	Avg bpp	Avg MS-SSIM	Avg bpp
JCT-VC Class B	BasketballDrive	0.961	0.006	0.813	0.1	0.72	0.1
	BQTerrace	0.97	0.13	0.993	0.11	**0.99**	0.11
	Cactus	0.978	0.21	0.481	0.17	**0.991**	0.11
	Kimono	*0.978*	*0.21*	0.941	0.12	**0.95**	0.12
	ParkScene	0.985	0.4	0.396	0.13	**0.951**	0.13

Table 8: MS-SSIM on JCT-VC Class C

Data Set	Video	Data 1(19)		Data 2(32)		Proposed DRAE	
		Avg MS-SSIM	Avg bpp	Avg MS-SSIM	Avg bpp	Avg MS-SSIM	Avg bpp
JCT-VC Class C	BasketballDrill	-	-	0.986	0.12	**0.99**	0.12
	BQMall	-	-	0.977	0.1	**0.98**	0.1
	PartyScene	-	-	0.984	0.15	**0.985**	0.15
	RaceHorses	-	-	0.703	0.12	0.629	0.12

Table 9: MS-SSIM on JCT-VC Class D

Data Set	Video	Data 1(19)		Data 2(32)		Proposed DRAE	
		Avg MS-SSIM	Avg bpp	Avg MS-SSIM	Avg bpp	Avg MS-SSIM	Avg bpp
JCT-VC Class D	BasketballPass	0.961	0.004	0.89	0.1	0.79	0.1
	BlowingBubbles	0.972	0.12	0.28	0.12	0.78	0.1
	BQSquare	0.981	0.19	0.99	0.11	**0.99**	0.1
	RaceHorses	0.987	0.32	0.69	0.11	**0.989**	0.11

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

Table 10: MSE on UVG (PSNR Metric)

Data Set	Video	MSE
UVG	Beauty	8.4
	Bosphorus	0.001
	HoneyBee	0.002
	Jockey	0.001
	ReadySetGo	0.002
	ShakeNDry	0.002
	YachtRide	0.001

Table 11: MSE on JCT-VC Class B (PSNR Metric)

Data Set	Video	MSE
JCT-VC Class B	BasketballDrive	0.001
	BQTerrace	0.001
	Cactus	0.001
	Kimono	0.001
	ParkScene	0.001

Table 12: MSE on JCT-VC Class C (PSNR Metric)

Data Set	Video	MSE
JCT-VC Class C	BasketballDrill	0.001
	BQMall	0.001
	PartyScene	0.001
	RaceHorses	0.001

Table 13: MSE on JCT-VC Class D (PSNR Metric)

Data Set	Video	MSE
JCT-VC Class D	BasketballPass	7.15
	BlowingBubbles	0.001
	BQSquare	0.001
	RaceHorses	0.001

Table 14: MSE on UVG (MS-SSIM Metric)

Data Set	Video	MSE
UVG	Beauty	0,09
	Bosphorus	0.12
	HoneyBee	0.07
	Jockey	0.07
	ReadySetGo	0.09
	ShakeNDry	0.06
	YachtRide	0.13

Table 15: MSE on JCT-VC Class B (MS-SSIM Metric)

Data Set	Video	MSE
JCT-VC Class B	BasketballDrive	0.03
	BQTerrace	0.10
	Cactus	0.09
	Kimono	0.08
	ParkScene	0.08

Table 16: MSE on JCT-VC Class C (MS-SSIM Metric)

Data Set	Video	MSE
JCT-VC Class C	BasketballDrill	0.03
	BQMall	0.05
	PartyScene	0.05
	RaceHorses	0.06

Table 17: MSE on JCT-VC Class D (MS-SSIM Metric)

Data Set	Video	MSE
JCT-VC Class D	BasketballPass	0.03
	BlowingBubbles	0.07
	BQSquare	0.17
	RaceHorses	0.06

6. CONCLUSION

This paper has proposed a Deep Recurrent Auto Encoders (DRAE) video compression approach. Specifically, we proposed Deep Recurrent Auto Encoders to compress motion and residual, fully exploring the temporal correlation in video frames. The proposed Deep Recurrent Auto Encoders model significantly expands the range of reference frames, which has not been achieved in previously learned as well as handcrafted standards. The experiments validate that the proposed approach outperforms all previous learned approaches in terms of both PSNR and MS-SSIM. Moreover, the proposed method can inspire traditional codecs, particularly the methods that integrate deep networks in traditional codecs, to adopt recurrent networks to improve their performance.

REFERENCES

[1]. D. J. Brady et al., "Multi scale giga pixel photography," Nature, vol. 486, no. 7403, pp. 386–389, 2012.

[2]. M. Cheng et al., "A dual camera system for high spatiotemporal resolution video acquisition," IEEE Trans. Pattern Anal. Mach. Intel., 2020.

[3]. F. Dufaux, P. Le Callet, R. Mantiuk, and M. Mrak, High Dynamic Range Video: From Acquisition, to Display and Applications. New York, NY, USA: Academic, 2016.

[4]. M. Winken, D. Marpe, H. Schwarz, and T. Wiegand, "Bit-depth scalable video coding," in Proc. IEEE Int. Conf. Image Process., vol. 1, pp. I-5–I-8, 2007.

[5]. P. Tudor, "Mpeg-2 video compression," Electron. & Commun. Eng. J., vol. 7, no. 6, pp. 257–264, 1995.

[6]. B. G. Haskell, A. Puri, and A. N. Netravali, Digital Video: An Introduction to MPEG-2. New York, NY, USA: Springer, 1996.

[7]. T. Sikora, "The MPEG-4 video standard verification model," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 1, pp. 19–31, 1997.

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

[9]. G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wieg and, "Overview of the high efficiency video coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, 2012.

[10]. V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (HEVC)," in Integrated Circuits and Systems, Algorithms and Architectures, vol. 39. New York, NY, USA: Springer, 2014, pp. 49–90.

[11]. G. J. Sullivan, P. N. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," Proc. SPIE, vol. 5558, pp. 454–474, 2004.

[12]. L. Yu, S. Chen, and J. Wang, "Overview of AVS-video coding standards," Signal Process., Image Commun., vol. 24, no. 4, pp. 247–262, 2009.

[13]. Simon Niklaus, "A reimplementation of SPyNet using PyTorch", https://github.com/sniklaus/ pytorch-spynet, 2018.

[14]. Anurag Ranjan and Michael J Black, "Optical flow estimation using a spatial pyramid network", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4161–4170, 2017.

[15]. R. Yang, F. Mentzer, L. Van Gool, and R. Timofte, "Learning for video compression with hierarchical quality and recurrent enhancement," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[16]. G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An end-to-end deep video "compression framework," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11 006–11 015, 2019.

[17]. T. Wieg and, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/AVC video coding standard" TCSVT, 13(7):560–576, 2003.

[18]. G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wieg and, et al, "Overview of the high efficiency video coding (HEVC) standard", TCSVT, 22(12):1649–1668, 2012.

[19]. Ren Yang," Learning for Video Compression with Recurrent Auto-Encoder and Recurrent Probability Model", IEEE Journal of Selected Topics in Signal Processing, 388 – 401, 2021.

[20]. Ren Yang, Fabian Mentzer, Luc Van Gool, Radu Timofte," Learning for Video Compression with Hierarchical Quality and Recurrent Enhancement", 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 13-19, 2020.

[21]. Shiping Zhu, Qinyao Chang, Qinghai Li, "Video saliency aware intelligent HD video compression with the improvement of visual quality and the reduction of coding complexity", springer publications, 2022.

[22]. Fan Zhang, Di Ma, Chen Feng, and David R. Bull, "Video Compression With CNN-Based Post processing", Published by the IEEE Computer Society, 2021.

[23]. F. Zhang, C. Feng, and D. R. Bull, "Enhancing VVC through CNN-based post-

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

processing," in Proc. IEEE Int. Conf. Multimedia Expo., pp. 1–6, 2020.

[24]. David Alexandre, Hsueh-Ming Hang, Wen-Hsiao Peng, Marek Domański, "deep video compression for inter frame coding", IEEE Xplore, 2022.

[25]. Zhihao Hu, Guo Lu, Dong Xu, "FVC: A New Framework towards Deep Video Compression in Feature Space", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Year: 2021.

[26]. Tong Chen, Haojie Liu, Qiu Shen, Tao Yue, Xun Cao, Zhan Ma," Deep Coder: A Deep Neural Network Based Video Compression", IEEE Visual Communications and Image Processing (VCIP), 2017.

[27]. Alan Jacob, Vedanta Pawar, Vinay Vishwakarma, Anand Mane, "Deep Learning Approach to Video Compression", IEEE Bombay Section Signature Conference (IBSSC) , 2019.

[28]. Zhibo Chen; Tianyu He; Xin Jin; Feng Wu, "Learning for Video Compression", IEEE Transactions on Circuits and Systems for Video Technology (Volume: 30, Issue: 2), 2020.

[29]. T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman. Video enhancement with task-oriented flow. ArXiv preprint arXiv: 1711.09078, 2017.

[30]. Kamal, S., Shende, V. G., Swaroopa, K., Bindhu Madhavi, P., Akram, P. S., Pant, K., & Sahile, K. (2022). FCN Network-Based Weed and Crop Segmentation for IoT-Aided Agriculture Applications. Wireless Communications and Mobile Computing.vol. 2022, Article, 2770706, 10 p, 2022. doi.org/10.1155/2022/2770706.

[31]. Meda, S. & Bhogapathi, R. (2022). An Integrated Machine Learning Model for Heart Disease Classification and Categorization. Journal of Computer Science, 18(4), 257-265. https://doi.org/10.3844/jcssp.2022.257.265.

[32]. Kumar, K. S., Madhavi, P. B. & Janaki, K. (2022). An Efficient Video Compression Framework using Deep Convolutional Neural Networks (DCNN). Journal of Computer Science, 18(7), 589-598.

[33]. https://doi.org/10.3844/jcssp.2022.589.598.

[34]. Dr. Sreelatha, Dr.R. Lakshmi Tulasi, Mr.K. Siva Kumar "An Optimized Framework of Video Compression Using Deep Convolutional Neural Networks (DCNN)", IJCSNS International Journal of Computer Science and Network Security, VOL.22 No.5, May 2022.

[35]. DOI: https://doi.org/10.22937/IJCSNS.2022.22.5.71.