Comprehensive Analysis of Flow Incorporated Neural Network based Lightweight Video Compression Architecture

Sangeeta[1], Preeti Gulia[2], Nasib Singh Gill[3]
Department of Computer Science and Applications
Maharshi Dayanand University, Rohtak
India

***Abstract*—The increasing video content over the internet motivated the exploration of novel approaches in the video compression domain. Though neural network based architectures have already emerge as de-facto in the field of image compression and analytics, their application in video compression also result in promising outputs. Adaptive and efficient compression techniques are required for video transmission over varying bandwidth. Several deep learning based techniques and enhancements were proposed and experimented but they didn't exhibit full optimal behavior and are not end to end trained and optimized. In the zest of a pure and end to end trainable compression technique, a deep learning based video compression architecture has been proposed comprises of frame autoencoder, flow autoencoder and motion extension network for the reconstruction of predicted frames. The video compression network has been designed incrementally and trained with random emission steps strategy. The proposed work results in significant improvement in visual perception quality measured in SSIM and PSNR when compared to some state-of-art techniques but in trade-off with frame reconstruction time sheet.**

Keywords—Deep learning; video compression; autoencoder; SSIM; PSNR

I. Introduction

The growing video content over the internet motivated the researchers to look for more proficient and efficient video compression techniques. The traditional in-use video compression techniques are manually designed and optimized. In recent years, deep learning based techniques are applied in various domain-specific applications including image and video compression too. The application of deep learning in image compression resulted in satisfactory results [1-5]. These methods focused on producing the quantization based binary representation of the images exploring various techniques like transmission of a subset of the encoded representation, learning variable quantization, training multiple models etc. The enhanced implementation of recurrent approach considerably improved the performance of the compression architectures.

The expanded architectures developed for image compression extended for videos also. But the task of video compression emerged as challenging due to the inclusion of motion information. The training of neural networks emerged with motion information emerged as very challenging. Recently, some developments have been made by the researchers to encode the video information in a trade-off with the complexity [6,7]. Though, some architecture resulted in superior performance in comparison to the traditional codecs, but with increased complexity and computation. This led to the exploration of learning based more efficient and less complex video compression methods.

The proposed method comprises of autoencoder style architecture. The architecture consists of frame compression/decompression, flow vector compression /decompression network, and finally a motion extension/frame reconstruction network. The frame and flow compression/decompression networks are composed of encoder and decoder networks. The encoder and decoder networks comprise of recurrent ConvGRU based frames with varying degrees of compression quality. The architecture has been designed and implemented incrementally. The performance analysis and ablation study reveals the significant improvement in compression quality when measured both in SSIM and PSNR with increased efficiency measured in time taken to generate a single frame, mentioned as TPF.

The work related to the proposed architecture has been described in Section 2. The detailed description of the architecture has been described in Section 3. The experimental details and results are presented in its subsequent section i.e. 3B. Section 4 presents the performance analysis of the proposed architecture with its comparative analysis. The whole work has been concluded in Section 5.

II. Related Work

The superfluous video content is taking a huge share of internet traffic [8]. The technological advancements have brought very high quality video formats and streaming of such formats over the web has brought new challenges to the compression standards. Although the in-use traditional techniques are performing well but doesn't give optimal results with the emerging new formats. Moreover, as the bandwidth is limited and varying, adaptive and highly efficient techniques are required to transmit the quality video content with minimal interruption. Discrete Cosine Transforms are mainly used in the block designed traditional techniques [9,10]. As these blocks based traditional techniques are developed incrementally, they cannot be end to end optimized.

The main focus of compression techniques is to remove redundancies and represent the frames in minimum number of bits. The reconstruction error got increased with the increment in compression rate. Initially designed video compression standards are the extended versions of image compression standards. In such techniques like motion JPEG, individual frames of the video are compressed to achieve whole compression. The exciting results in the field of deep learning based image compression attained the researchers' attention and found some of the autoencoder techniques more potent and proficient than traditional schemes [11-15]. Decreasing rate distortion error is the primary goal of these compression schemes. The use of RNNs in some image compression architectures also improved the performance [16]. RNN based architectures are more suitable for varying compression rate. Adaptive compression techniques are required to transmit the quality and uninterruptable video content over the varying bandwidth. Some variable image size compatible video compression architectures comprising of CNNs were proposed to remove spatial redundancies [11,14,17]. Entropy encoding has been used in such techniques to achieve improved compression. In addition to CNN and RNN based architectures, several different quantization and probability driven adaptive arithmetic coding based schemes were proposed and evaluated [18,19]. Such deep learning based explored techniques resulted in improved performance compared to the standard codecs.

The exciting compression quality achieved in the field of image compression using deep learning based approaches lead to exploration of their video compatible extended versions. As videos includes more redundant information, it is imperative to have rigorous approach in the expanded formats. The widely used traditional codecs like H.264 or H.265 are block designed [20]. Their recent used versions are evolved with time by extensive engineering efforts. Their incremental block based design does not support end to end optimization. Rather, each block can be optimized or extended individually. Their predictive coding is based on the continual prediction of P or B frames from I frames extracting the required information. Initially, extensions to the existing codecs were proposed based on deep learning based schemes.

Later, researchers' explored pure deep learning based end to end optimizable approaches using different architectures and strategies. Some of the video compression architectures based on image interpolation were designed [21-23]. Several flow based techniques were presented for the prediction of the frames and spatial varying data will be learned by the Convolutional kernel. For the slow and small video frames, image extrapolation has performed well in frame prediction [24-26]. The efforts put forth in the design of deep learning based architectures of DVC in [7] and adversarial video compressions in [28] are well appreciated. A number of efficient deep learning based architectures have been developed over the years but each having its own trade off. Some of them suffer from the performance trade off either with complexity or computation.. In addition to the compression sphere, researches have also been extended to the extraction of information from compressed formats without decompressing [27]. Our research is also motivated from the same idea of designing of such compression architecture whose compressed format can also be parsed efficiently for analytics purpose.

III. Proposed Work

The proposed architecture is a neural network based scheme for video compression. A frame auto-encoder based compression network has been designed using CNN and ConvGRU units. The input frames are taken consecutively by the encoder network and presents the encoded form to the corresponding decoder. The reconstructed frames are generated by the decoder from the encoded format. The encoder and decoder networks of the frame autoencoder are trained together. A Flow Autoencoder is also incorporated to compute the optical flow. Optical Flow is used for the motion information lies between consecutive frames of a video. The Motion Extension Network is used to reconstruct the next frames using optical flow and decoded frame from frame autoencoder. The proposed system is modelled in Tensorflow.

A. Network Architecture

The frame autoencoder is the vital part of compression architecture. It comprises of the encoder and decoder networks comprising of CNN and ConvGRU units. The encoder encodes the frames with varying degrees of compression quality. The binary format has been quantized before passing to the decoder. The decoder regenerates the frame from the encoded format according to the degree of compression. Farneback based Flow computation has been used for the motion estimation and prediction among the consecutive frames. Flow autoencoder has been incorporated to compress the computed flow value. Motion extension network reconstructs the frames based on the current frame from the frame autoencoder, the previous frame and decoded flow value as illustrated in Fig. 2. The overview of the proposed architecture has been presented in Fig. 1.

Flow Vector estimation, compression and decompression is done using the traditional Farneback flow estimation method. The flow vectors between every two frames are estimated. The estimated flow vectors are then compressed using a standard CNN based encoder network with Generalized Divisive Normalization (GDN) layers as the nonlinearity (Fig. 3). A CNN based decoder network with Inverse GDN as the nonlinearity is used to decompress the flow vectors.

The structural distortion among the input and output frames has been minimized by following loss function:

$$F(xt, x't) = \lambda 1 \; SSIM(xt, x't) + \lambda 2 \; MSE(xt, x't)$$

where xt and x't represents the input and output videos frames respectively. λ1 is the multiplier and SSIM represents the Structural Similarity Index Metric Loss. λ2 is also the multiplier and MSE denotes the mean square error amid the video input and the output frames.

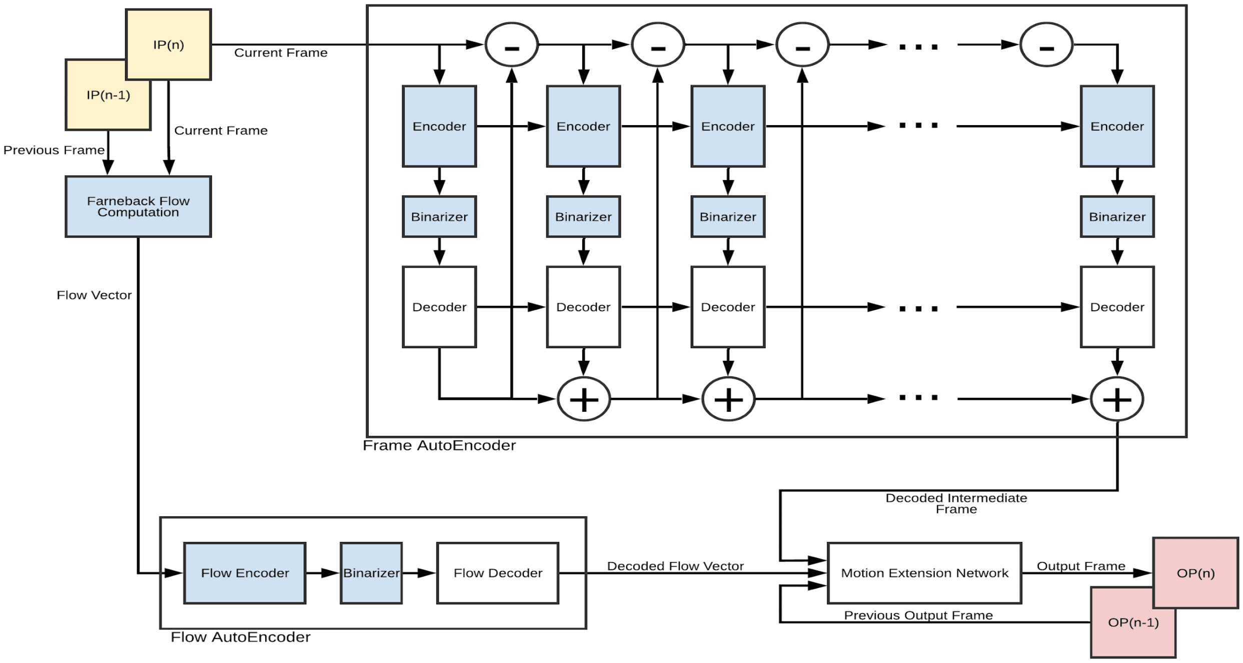

Fig. 1. The Compression Network Architecture.

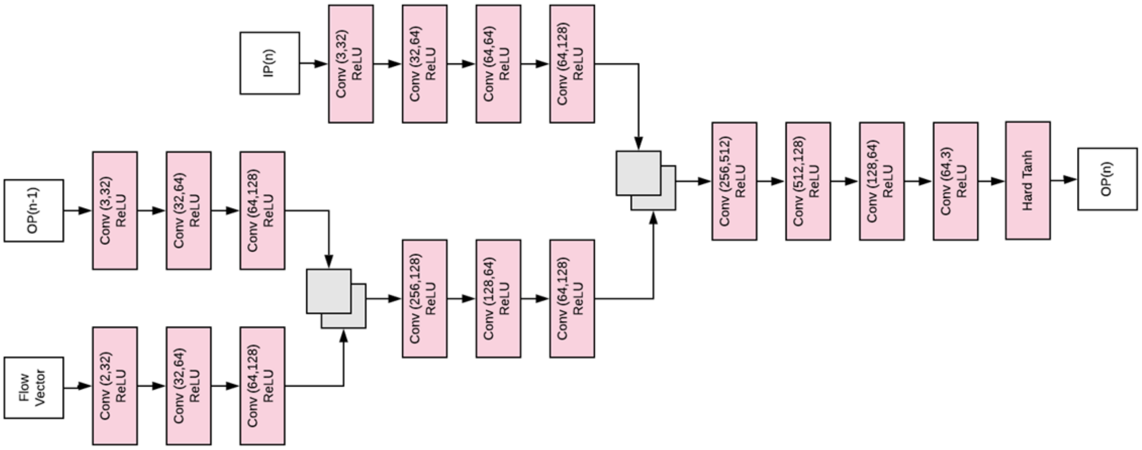

Fig. 2. Motion Extension Network.

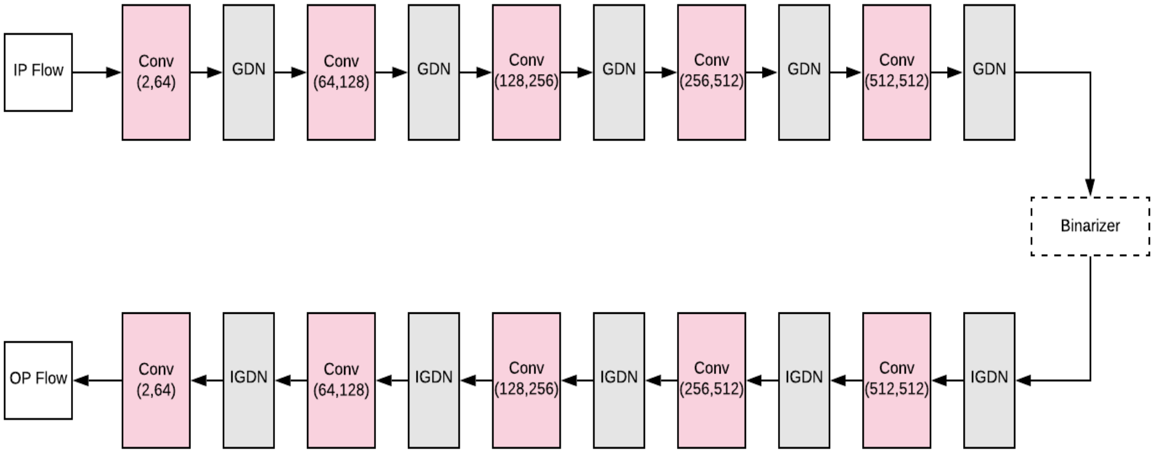

Fig. 3. Flow Autoencoder.

B. *Experiment*

Dataset: A dataset comprising of 20s long 571 small videos out of total 826 videos from Youtube UGC has been used to train the network, remaining clips has been for testing and validation. Videos of varying quality have been chosen i.e. 480p, 360p and 720p. The frame size has been chosen as 64x64, so video clips of all quality firstly rescaled to the chosen format, and then training is performed. Videos frames are taken randomly during training but while testing the clips are chosen from the starting. The model has been trained with randomized emission step training strategy with emission steps varying from 1 to 10. Addition of each emission step improves the output but have an effect on the compression efficiency.

Implementation Details: For the implementation purpose, a single T4, K80 or P100 GPU has been used to train the network on the Google Colaboratory platform. ƛ1 is taken as one and ƛ2 be 10. The frames have been kept to the size of 64 x 64. 10e-4 be the learning rate with Adam Optimizer. During the training of frame encoder with 100 epochs; at 50th, 70th and 90th epoch; the learning rate has been divided by ten. But for the whole model training, only 70 epochs have been used after stacking the framer encoder first and learning rate has been altered at 35th and 55th epoch by dividing ten.

Evaluation: SSIM i.e. Structural Similarity Index and PSNR i.e. Peak Signal to Noise Ratio has been used to measure the visual quality of the reconstructed frames. The temporal distortion encountered among the frames has been evaluated by Flow EPE i.e. End Point Error. Moreover, the reconstruction time of individual frames has been measured by the TPF i.e. Time per Frame parameter.

IV. Experimental Results and Discussion

The proposed architecture has been evaluated in terms of perception quality, residual error and efficiency. The experimental results of the network have been obtained for four performance parameters i.e. SSIM, PSNR, EPE and TPF. SSIM, Structural Similarity Index is a good measure of visual perception. Higher the SSIM value, good is the quality of image/video frames. PSNR presents Peak Signal to Noise Ratio. It represents the image quality in term of mean square error. Lower the value of PSNR better will be the image. Flow-EPE, Flow- End Point Error is used to measure the quality of video frames reconstructed in terms of residual error between consecutive frames of a video. The efficiency of the architecture is observed in terms of time required to generate a frame. The Green cell represents Highest achieved value and Red cell represents lowest achieved value.

A. *Performance Analysis*

The performance of the proposed architecture is measured in terms of both visual perception and efficiency. The experimental values obtained for the performance parameters namely SSIM, PSNR, Flow EPE and Time per frame are given in the Tables I to IV respectively. Moreover, the corresponding change in the parameters' values with increment of each additional emission step has shown in Fig. 4 to 7. The proposed network has been designed incrementally. Firstly, the results were obtained with simple frame autoencoder trained with randomized training strategy. Secondly, Motion Extension Network has been incorporated with Frame Autoencoder named as MotionNet Randomized. The values of all four parameters are obtained for each emission step. The graphical representation shows a significant rise in SSIM, PSNR and TPF with each additional emission step in all three randomized architectures. Incorporation of Optical flow and Motion Extension Network results in improved visual quality. The same can be observed by 0.044 rises in SSIM with 3.3 increments in PSNR value.

TABLE I. SSIM Values Obtained Per Emission Step

SSIM	Baseline	ConvGRU Randomized	MotionNet Randomized	Flow-MotionNet Randomized
1.	0.67	0.652	0.706	0.709
2.	0.67	0.768	0.813	0.819
3.	0.67	0.823	0.866	0.874
4.	0.67	0.864	0.902	0.91
5.	0.67	0.883	0.924	0.932
6.	0.67	0.893	0.938	0.948
7.	0.67	0.916	0.948	0.957
8.	0.67	0.917	0.951	0.961
9.	0.67	0.92	0.953	0.963
10.	0.67	0.919	0.954	0.963

TABLE II. PSNR Values Obtained Per Emission Step

PSNR	Baseline	ConvGRU Randomized	MotionNet Randomized	Flow-MotionNet Randomized
1.	18.9	18.3	20	20
2.	18.9	21.1	22.2	22.5
3.	18.9	22.4	23.5	24.1
4.	18.9	23.7	24.8	25.5
5.	18.9	24.3	25.8	26.7
6.	18.9	24.6	26.6	27.8
7.	18.9	25.8	27.2	28.4
8.	18.9	25.7	27.6	28.9
9.	18.9	26	27.8	29.1
10.	18.9	25.9	27.8	29.2

TABLE III. Flow EPE Values Obtained Per Emission Step

Flow EPE	Baseline	ConvGRU Randomized	MotionNet Randomized	Flow-MotionNet Randomized
1.	1.154	1.251	0.826	0.822
2.	1.154	0.613	0.477	0.577
3.	1.154	0.555	0.383	0.368
4.	1.154	0.409	0.35	0.276
5.	1.154	0.311	0.273	0.226
6.	1.154	0.319	0.248	0.253
7.	1.154	0.273	0.173	0.189
8.	1.154	0.221	0.199	0.17
9.	1.154	0.201	0.175	0.148
10.	1.154	0.173	0.189	0.17

TABLE IV. TIME PER FRAME VALUES OBTAINED PER EMISSION STEP

TPF	Baseline	ConvGRU Randomized	MotionNet Randomized	Flow-MotionNet Randomized
1.	0.015	0.0182	0.0208	0.0243
2.	0.015	0.0186	0.0214	0.0248
3.	0.015	0.0191	0.0218	0.0254
4.	0.015	0.0195	0.0224	0.0258
5.	0.015	0.0201	0.023	0.0263
6.	0.015	0.0205	0.0234	0.0268
7.	0.015	0.0211	0.0239	0.0274
8.	0.015	0.0216	0.0245	0.0279
9.	0.015	0.0221	0.025	0.0285
10.	0.015	0.0226	0.0255	0.029

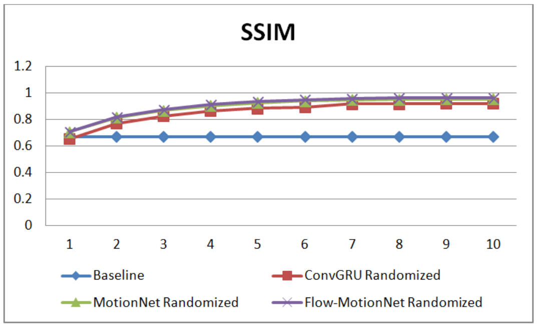

Fig. 4. SSIM Values per Emission.

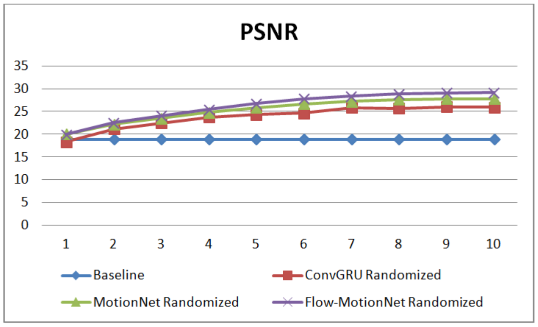

Fig. 5. PSNR Values per Emission.

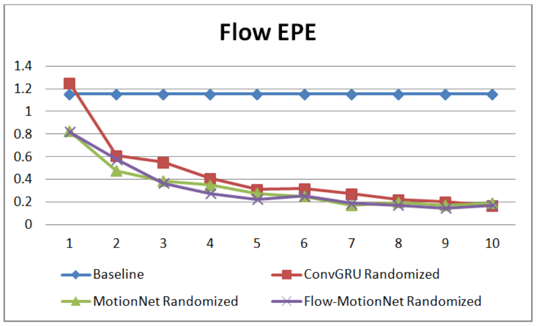

Fig. 6. Flow EPE Values per Emission.

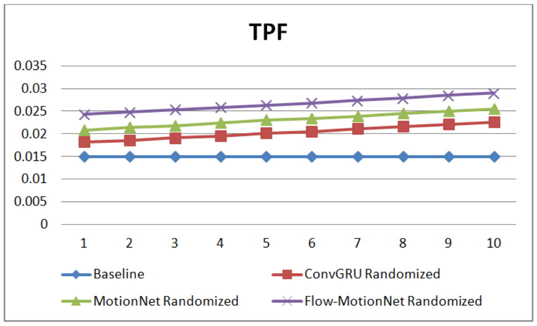

Fig. 7. TPF Values per Emission.

The error in consecutive frames of the video has been measured by Flow-EPE. In general, the EPE values are decreasing with each additional emission step but some fluctuations have been observed in some emission steps like the smallest value of EPE has been obtained after 9th emission step instead of 10th step. But in comparison to the simple frame autoencoder, a slight reduction of 0.003 EPE value has been observed if compared for last emission step. The efficiency of the network has been observed in terms of time required for the network to regenerate a single frame. As the proposed network comprises of optical flow and Motion Extension Network, the increase in computation resulted in slight increase in TPF value, so increased value of TPF has been observed for the proposed network. The analysis of the outcomes reveals that the proposed architecture shows a significant improvement in visual quality but with slight cost of regeneration time. This architecture can be further enhanced by plugging other optimized networks like optical flow, entropy coding etc.

The performance of adaptive bit rate video compression has been analyzed with the average values of performance parameters obtained for all emission steps. The below Table V show the average values of the performance parameters. Here also, the proposed architecture shows a significant improvement in SSIM and PSNR values eventually leading to better video quality frames. But the average TPF value has been increased by 0.00628 units. The incorporation of optical flow and motion extension network, though contributed in improving the visual quality of frames but increased the computation of the network leading to enhanced time in frame regeneration.

TABLE V. AVERAGE PERFORMANCE IN 10 EMISSION STEPS

	Avg. SSIM	Avg. PSNR	Avg. EPE	Avg. TPF
Baseline	0.67	18.9	1.154	0.015
ConvGRU Randomized	0.8555	23.78	0.4326	0.02034
MotionNet Randomized	0.8955	25.33	0.3293	0.02317
Flow-MotionNet Randomized	0.9036	26.22	0.3199	0.02662

B. Comparison with State-of-Art Architectures

The outcomes of the proposed architecture have also been compared with the state-of-art conventional compression techniques like H.264 and H.265 and also with the deep learning based models proposed by authors of DVC [7] and Adversarial video compression [28].

For comparison, SSIM and PSNR metrics are used to relatively measure the perception quality. MS-SSIM correlates better with human perception of distortion. The proposed model outperformed in terms of MS-SSIM metrics. Table VI represents the MS-SSIM and PSNR values of the various architectures. The proposed model achieved good SSIM performance but with a drop in PSNR value.

TABLE VI. MS-SSIM VALUES OF VARIOUS ARCHITECTURES

Architecture	MS-SSIM	PSNR
H.264	0.955	34
H.265	0.96	36
Adversarial video compression [28]	0.9476	28.46
DVC [7]	0.955	35.5
Flow-MotionNet (Proposed)	0.963	29.2

V. CONCLUSION

Deep Learning is becoming a milestone in the field of both compression and analytics. Some deep learning based enhancements and improvements surpass the traditional techniques in both qualitative and quantitative measurements. These positive outcomes motivated the exploration of pure deep learning based video compression strategies which can be end to end trained and optimized. This paper also presents a simple lightweight adaptive deep learning based architecture comprises of optical flow and motion extension network trained with randomized training strategy with ten varying emission steps. A ConvGRU unit has been used in each layer of both the encoder and decoder networks of frame autoencoder. Optical Flow has also been used for the motion depiction which eventually helps in frame regeneration with frame autoencoder decoded output in motion extension network. The performance analysis depicts a significant improvement in visual quality measured in terms of both SSIM and PSNR but in trade-off with frame regeneration time. The performance of the proposed architecture can be further improved by addition of other optimization strategies.

ACKNOWLEDGMENT

The first author expresses her thanks to University Grant Commission, India for providing fellowship in form of JRF to support this work.

REFERENCES

[1] J. Ball´e, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. In Int'l. Conf. on Learning Representations (ICLR2017), Toulon, France, April 2017. Available at http://arxiv.org/abs/1611.01704.

[2] K. Gregor, F. Besse, D. Jimenez Rezende, I. Danihelka, and D.Wierstra. Towards conceptual compression. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 3549–3557. Curran Associates, Inc., 2016.

[3] L. Theis, W. Shi, A. Cunningham, and F. Huszar. Lossy image compression with compressive autoencoders. In Int'l. Conf. on Learning Representations (ICLR2017), 2017.

[4] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar. Variable rate image compression with recurrent neural networks. ICLR 2016, 2016.

[5] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell. Full resolution image compression b jgwith recurrent neural networks. CVPR, abs/1608.05148, 2017.

[6] Zhibo Chen, Tianyu He, Xin Jin, Feng Wu, "Learning for video compression", arXiv:1804.09869v2 [cs.MM] 9 Jan 2019.

[7] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai and Zhiyong Gao. DVC:An End-to-end Deep Video Compression Framework. arXiv: 1812.00101v3 [eess.IV] 7Apr 2019.

[8] C.V. Networking Index, "Forecast and methodology," 2016-2021 CISCO White paper, 2016.

[9] I.E. Richardson, "Video codec design: developing image and video compression systems" John Wiley & Sons, 2002.

[10] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," in TCSVT, 2007.

[11] J. Ball´e, V. Laparra, E.P. Simoncelli, "End-to-end optimized image compression" in ICLR, 2017.

[12] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S.J. Hwang, J. Shor, G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," arXiv preprint arXiv:1703.10114, 2017.

[13] O. Rippel, L. Bourdev, "Real-time adaptive image compression" in ICML, 2017.

[14] L. Theis, W. Shi, A. Cunningham, F. Husz´ar, "Lossy image compression with compressive autoencoders," in ICLR, 2017.

[15] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, M. Covell, "Full resolution image compression with recurrent neural networks," in CVPR, 2017.

[16] M.H. Baig, V. Koltun, L. Torresani, "Learning to inpaint for image compression," in NIPS, 2017.

[17] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, L. Van Gool, "Conditional probability models for deep image compression" arXiv preprint arXiv:1801.04260, 2018.

[18] A.v.d. Oord, N. Kalchbrenner, K. Kavukcuoglu, "Pixel recurrent neural networks" in ICML, 2016.

[19] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, L.V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in NIPS, 2017.

[20] D. Le Gall, "MPEG: A video compression standard for multimedia applications," Communications of the ACM, 1991.

[21] X. Jia, B. De Brabandere, T. Tuytelaars, L.V. Gool, "Dynamic filter networks," in NIPS, 2016.

[22] Z. Liu, R. Yeh, X. Tang, Y. Liu, A. Agarwala, "Video frame synthesis using deep voxel flow" in ICCV, 2017.

[23] S. Niklaus, L. Mai, F. Liu, "Video frame interpolation via adaptive separable convolution" in ICCV, 2017.

[24] C. Vondrick, H. Pirsiavash, A. Torralba, "Generating videos with scene dynamics" in NIPS, 2016.

[25] T. Xue, J. Wu, K. Bouman, B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in NIPS, 2016.

[26] M. Mathieu, C. Couprie, Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in ICLR, 2016.

[27] Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A.J. Smola, P. Kr¨ahenb¨uhl, "Compressed video action recognition," in CVPR, 2018.

[28] Sungsoo Kim, Jin Soo Park, Christos G. Bampis, Jaeseong Lee,Mia K. Markey, Alexandros G. Dimakis, Alan C. Bovik.: Adversarial Video Compression Guided by Soft Edge Detection. arXiv:1811.10673v1 [eess.IV] 26 Nov 2018.