Journal of Computer Science

Original Research Paper

An Efficient Video Compression Framework using Deep Convolutional Neural Networks (DCNN)

[1]Kommerla Siva Kumar, [2]P. Bindhu Madhavi and [3]K. Janaki

[1]Department of Computer Science and Engineering, R.V.R and J.C College of Engineering, Andhra Pradesh, India
[2]Department of AI and ML, The Oxford College of Engineering, Karnataka, India
[3]Department of Computer Science and Engineering-AI, Faculty of Engineering, Jain Deemed to be University, Karnataka, India

Article history
Received: 30-03-2022
Revised: 02-06-2022
Accepted: 06-06-2022

Corresponding Author:
Kommerla Siva Kumar
Department of Computer Science and Engineering, R.V.R and J.C College of Engineering, Andhra Pradesh, India
Email: kommerlasivakumar@gmail.com

Abstract: In the current world, video streaming has grown in popularity and now accounts for a large percentage of internet traffic, making it challenging for service providers to broadcast videos at high rates while utilizing less storage space. To follow inefficient analytical coding design, previous video compression prototypes require non-learning-based designs. As a result, we propose a DCNN technique that integrates OFE-Net, MVE-Net, MVD-Net, MC-Net, RE-Net, and RD-Net for getting an ideal collection of frames by linking each frame pixel with preceding and following frames, then finding linked blocks and minimizing un needed pixels. In terms of MS-SIM and PSNR, the proposed DCNN approach produces good video quality at low bit rates.

Keywords: Deep Neural Networks, Encoding, Decoding, Video Compression

Introduction

People who watch videos on the internet are about 90%, this is expected to rise in the near future. As a result, an effective video compression model is required to deliver higher-quality frames while using less bandwidth.

Video codecs compress videos using hand-drawn models. Despite their superb design, the present models are poorly optimized. The video compression process can be improved even more by tweaking the entire codec model.

Deep neural networks have outperformed classic picture codecs like the Joint Photographic Experts Group in video compression. Deep neural network-based models that rely on extremely nonlinear transformations require end-to-end training.

It's not easy to create a model that uses a variety of video compression algorithms. Motion estimation, which creates and compresses motion data, is the most important part. To remove temporal redundancy, video compression significantly relies on motion information. The only way to express motion vectors is to use an optical flow net. Although learning-based optical flow estimation focuses on obtaining exact flow data, proper optical flow isn't always the best solution for specific video applications. Furthermore, the ability of optical flow data is greater than that of existing models, resulting in high bit rate information when optical flow values are directly compressed using existing methods.

Reduced rate-distortion aims to provide higher-quality reconstructed frames at the same bit rate. It is essential for proper video compression to technique.

Rate distortion must be decreased to achieve the benefits of end-to-end training for deep learning-based video compression models. The following are the model's key benefits: All steps of the DCNN model are implemented using deep neural networks. The DCNN model is based on rate-distortion and uses a single loss function to combine all of the steps, resulting in a high compression ratio. This study will aid researchers working on computer vision, video compression, and deep model creation.

Related Work

Kumar and Janaki (2020), the video compression task can be categorized into three types. They are the classical era, the era of generic heuristics, and the era of modern techniques with deep learning. Through the detailed study of the literature through the past decades, it is learned that various schemes have been proposed for video compression. These schemes have contributed a lot of efficient mechanisms in different ways. However, further improvements are also needed towards the same pertaining to the limitations observed as specified.

Birman *et al*. (2020), illustrate and explain various issues for the video compressionprocess in the field of DNNs. ,Still additional investigation is looking to achievethe upcoming generation and neural networks-based codecs.

Ranjan and Black (2017) ,have presented a deep network with afast and lightweight model for the optical flow process. The previous pyramid featureis replaced with a U-shaped networkand this model obtains better results. And thismodel can help computer vision applications.

Dai *et al.* (2009) ,described an optical flow approach that provides the features of deep learning-based optical flow algorithms. This approach gives better accuracy results compared with an existing method and surpasses it in several benchmarks.

Bao *et al.* (2019), the Estimation and Compensation of Motion (MEMC) neural network is proposed for learning and improving video frame interpolation. This model takes advantage of the MEMC framework's capabilities for managing massive amounts of motion data, as well as learning-based methods for extracting features quickly. Many video enhancement activities can be done with this MEMC framework. The qualitative and quantitative evaluation of these methods against state-of-the-art video interpolation and improvement algorithms on various standard data sets demonstrates that they outperform them.

Wu *et al.* (2020), describes a video compression framework based on deep learning which provides MV and RP network. Here the experiment results show that MV and RP networks be able to improve the performance of compression by modeling spatial correlations among the frames accurately.

Chen *et al.* (2017), present an efficient video compression framework based on deep learning. Here comparison of × 264, with this Deep Coder has shown a similar type of coding efficiency (lossy) with the familiar testing series used by the video coding society and video compression on deep learning is an alternative framework for the process of video coding in feature.

Chen *et al*. (2019), propose PMCNN and modeled spatiotemporal to achieve predictive-based coding and a learning-based framework of an effective process for video compression is explored. Even though lack of entropy-based coding and still this achieves a better result for video compression, exhibiting new attainable handling of video compression.

Balle *et al.* (2016), presented a nonlinear transform coding-based image compression method and a framework to optimize it end-to-end for rate-distortion performance. Nevertheless, additional visual improvements might be possible in terms of a perceptual metric like MSE, if the method were optimized.

Balle *et al.* (2018) ,provide a variational auto encoder-based image compression trainable model. When evaluating rate-distortion performance using a traditional metric based on squared error, this model leads to picture compression when using the MS-SSIM index and it outperforms ANN-based techniques when using a traditional metric based on squared error (PSNR).

The limitations of existing work are:

- Lu *et al*. (2019) and (Yang *et al*., 2020a), the encoding procedures are more complex due to the processing of large-size videos
- Lu *et al*. (2019) and (Yang *et al*., 2020b), at decoding, the performance of video quality is degraded with very usual frame drops during the encoding process
- Cheng *et al*. (2019), in the video transmission process, to send a greater number of frames an end-to-end delay has occurred
- Cheng *et al*. (2019), in the video transmission process, to send a smaller number of frames, the compression ratio is decreased
- Cheng *et al*. (2019), the deep learning scheme presented so far solely depends on training sample sets and video frames
- Habibian *et al.* (2019), over-sampling and under-sampling are the most common phenomena affecting these schemes
- Habibian *et al.* (2019), the present schemes need more flexibility
- Meda and Bhogapathi (2022), discussed the utilization of neural network models.
- Kamal *et al.* (2022), still there is a need for standard deep learning mechanisms for video compression

Materials and Methods

The above limitations are overcome in the proposed approach and the objectives of the proposed work are:

- To reduce the storage space occupied by video
- To decrease the time taken for video during transfer
- To enhance the video quality with a better compression ratio

Introducing the symbols: Assume $V = \{F_1, F_2, ...F_{t-1}, F_t ...\}$ represents the sequences of current video and at time step t, F_t is frame. The symbols $\bar{F}_t$ and $\hat{F}_t$ represent predicted frames and reconstructed or decoded frames. The residual information or error information between original frame F_t and predicted frame $\bar{F}_t$ is R_t.

The reconstructed (decoded residual) information is denoted by $\hat{R}_t$. In order, motion information is essential to reduce temporal redundancy. Among them, the optical flow or motion vector value represents V_t and its corresponding reconstructed form is $\hat{V}_t$. To improve the compression efficiency, either linear transform or nonlinear transform techniques can be used. Consequently, residual information R_t is converted to Y_t, motion information V_t converted to M_t, and corresponding quantized versions

$\hat{R}_t$ and $\hat{M}_t$ respectively. The detailed architecture of proposed DCNN approach is shown in Fig. 1 and description of each step is as follows:

Step 1: Motion estimation

We use an OFE Net to estimate the optical flow, which is considered as motion information V_t:

Step 2: Motion compression

The MVE-MVD net is proposed for compressing and decoding optical flow values. A sequence of convolution and nonlinear-transform procedures are used to extract or provide the optical flow V_t. After then, $\hat{M}_t$ is quantized to M_t. The MVD net obtains the quantized representation, which is subsequently used to reconstruct the motion information $\hat{V}_t$. Entropy coding will also be done using the quantized representation $\hat{M}_t$:

Step 3: MC net

The motion compensation network obtains predicted frame $\overline{F}_t$, which is near to the current frame F_t as possible, using both previously reconstructed frame $\hat{F}_{t-1}$ and motion vector $\hat{V}_t$. In beginning, the previous frame $\hat{F}_{t-1}$ warped to the present frame using motion information $\hat{V}_t$. However, there are still artifacts in the warped frame. To remove these artifacts, we send warped frame $W(\hat{F}_{t-1}, V_t)$, reference frame $\hat{F}_{t-1}$, and the motion vector $\hat{V}_t$ into another CNN, which produces the refined predicted frame $\overline{F}_t$. The proposed method follows a pixel-based motion compensation strategy that gives more precise temporal information:

Step 4: RE net and RD net

The residual encoder network encodes the remaining data that exists between the original frame and the forecasted frame. Our method may more effectively harness the potential of non-linear transform and produce higher compression efficiency as compared to the discrete cosine transform used in the conventional video compression system:

Step 5: BRE net

The quantized motion information $\hat{M}_t$ from Step 2 and residual information $\hat{Y}_t$ from Step 4 is coded into bits and transmitted to the decoder during the testing stage. From the training stage, by employing CNNs for the number of bits costs are estimated (BRE Net in figure) and subsequently acquire the probability distribution of each symbol in $\hat{M}_t$ and $\hat{Y}_t$:

Step 6: Frame reconstruction

By adding $\overline{F}_t$ in Step 3 and $\hat{R}_t$ in Step 4, obtains the reconstructed frame $\hat{F}_t$, i.e., $\hat{F}_t = \overline{F}_t + \hat{R}_t$:

Training and Testing

Guo *et al.* (2019), provides the training (Vimeo) and testing (UVG & JCT-VC) datasets. Our proposed DCNN technique trained the 90k dataset (Vimeo) and tested the Ultra-Video-Group dataset (UVG) and Joint Collaborative Team-Video Coding dataset (JCT-VC).

Results and Discussion

The Average is calculated by PSNR and bpp in Tables 1, 2, 3, and 4.

And drawn the corresponding Fig. 2 (a), (b), (c), (d) - Table 1, 2(b) -Table 2, 2(c)-Table 3 and 2(d) - Table 4, compared with existing methods like data1 (Lu *et al*., 2019), data 2 (Yang *et al*., 2020a), data3(Yang *et al*., 2020b) and data4 (Yang *et al*., 2020a) and produce better average psnr and average bpp values.

The Average is calculated by MS-SSIM and bpp in Tables 5, 6, 7, and 8.

And drawn the corresponding Fig. 3 (a), (b), (c), (d) - Table 5, 3(b)- 6, (c) - 7, and (d) -8, compared with existing methods like data1 (Lu *et al*., 2019), data2 (Yang *et al*., 2020b), data 3 (Yang *et al*., 2020a) and data 4 (Yang *et al*., 2020b) 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 9, 10, 11, and 12 (mean square error rate) for the tested data sets are also given.

Similarly, the proposed approach (in terms of MS-SSIM metric) proves to reduce the error rate with the following MSE Table 13, 14, 15, and 16 (mean square error rate) for the tested data sets are also given.

The proposed approach proves for better compression ratio (in terms of PSNR) with the following Tables 17, 18, 19 and 20 for the tested data sets are given.

Similarly, the proposed approach proves for better compression ratio (in terms of MS-SSIM) with the following Tables 21, 22, 23, and 24 for the tested data sets are given.

Similarly, the compressed/output video takes less transmission time than original/input video.

DOI: 10.3844/jcssp.2022.589.598

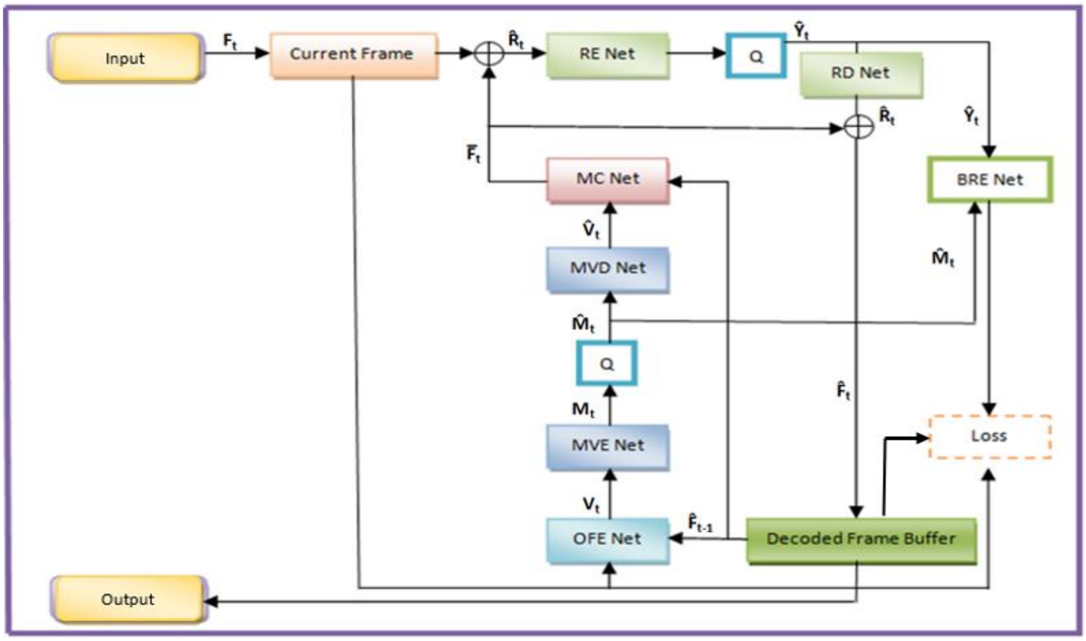

Fig. 1: The architecture of our proposed DCNN approach

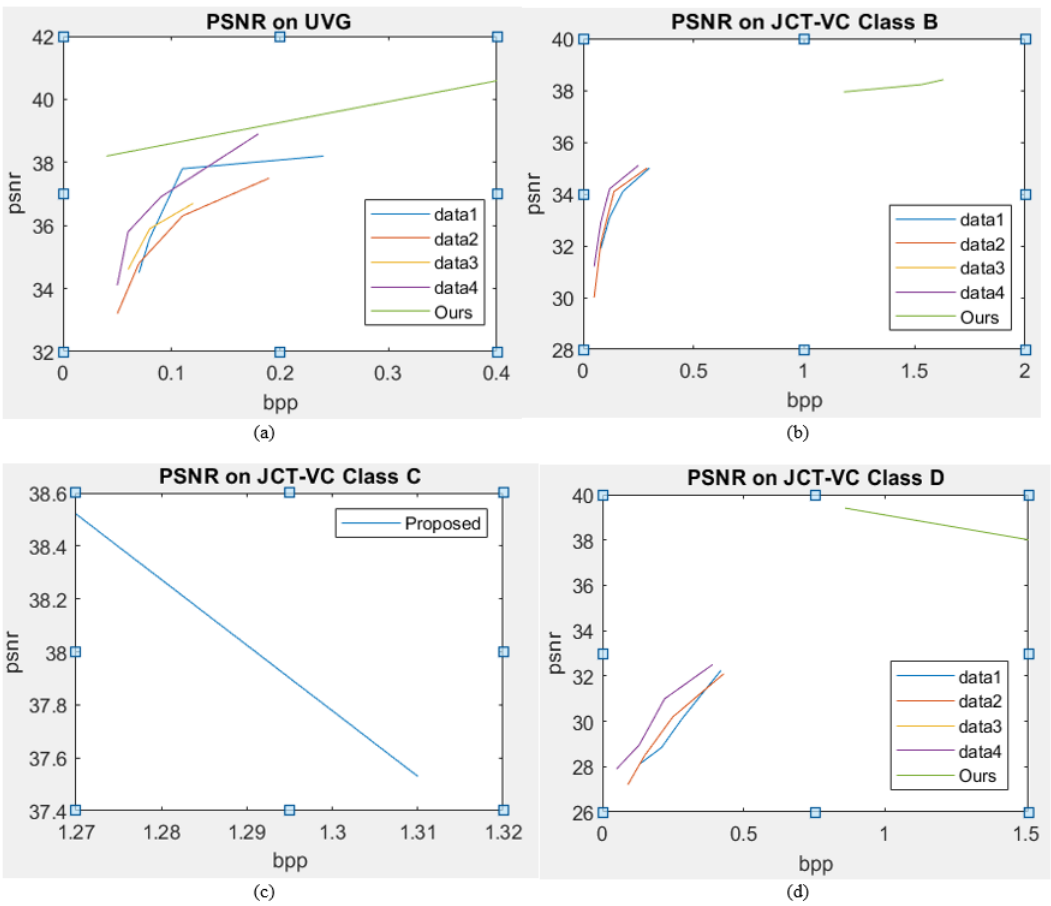

Fig. 2: Average PSNR and bpp

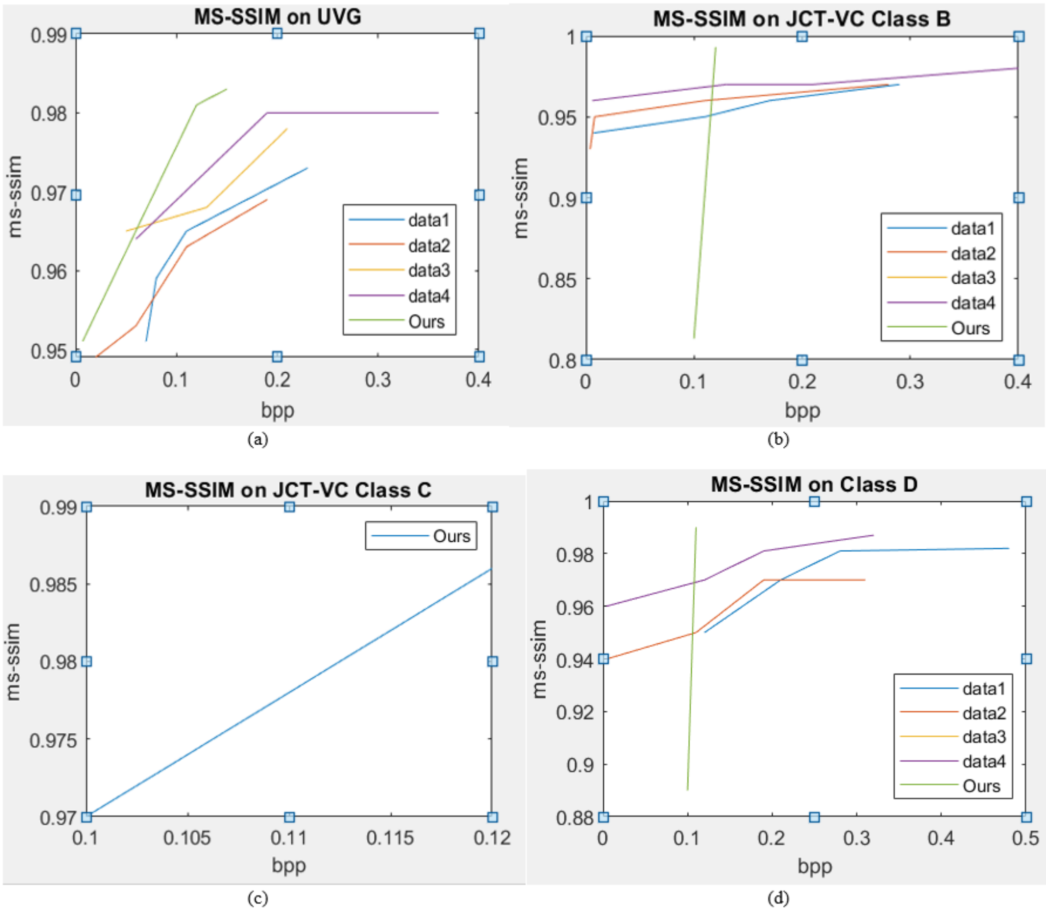

Fig. 3: Average MS-SSIM and bpp

Table 1: PSNR on UVG

Data set	Video	Avg PSNR	Avg bpp
UVG	Beauty	40.59	0.40
	Bosphorus	38.20	0.94
	Honey Bee	36.46	1.99
	Jockey	37.83	1.05
	Ready set go	37.04	1.80
	Shaken dry	36.05	2.00
	Yacht ride	38.05	1.12

Table 2: PSNR on JCT-VC class B

Data set	Video	Avg PSNR	Avg bpp
JCT-VC class B	Basketball drive	38.26	1.020
	BQ terrace	38.22	1.530
	Cactus	38.41	1.630
	Kimono	37.94	1.180
	Park scene	37.40	1.568

Table 3: PSNR on JCT-VC class C

Data set	Video	Avg PSNR	Avg bpp
JCT-VC class C	Basketball drill	37.53	1.31
	BQ mall	38.52	1.27
	Party scene	37.09	2.34
	Race horses	37.91	1.60

Table 4: PSNR on JCT-VC class D

Data set	Video	Avg PSNR	Avg bpp
JCT-VC class D	Basketball pass	39.41	0.86
	Blowing bubbles	38.01	1.51
	BQ square	38.65	1.76
	Race horses	38.15	1.37

Table 5: MS-SSIM on UVG

Data set	Video	Avg MS-SSIM	Avg bpp
UVG	Beauty	0.951	0.07
	Bosphorus	0.983	0.10
	Honey bee	0.913	0.14
	Jockey	0.888	0.11
	Ready set go	0.981	0.12
	Shaken dry	0.983	0.15
	Yacht ride	0.950	0.10

Table 6: MS-SSIM on JCT-VC class B

Data set	Video	Avg MS-SSIM	Avg bpp
JCT-VC class B	Basketball drive	0.813	0.10
	BQ terrace	0.993	0.11
	Cactus	0.481	0.17
	Kimono	0.941	0.12
	Park scene	0.396	0.13

Table 7: MS-SSIM On JCT-VC class C

Data set	Video	Avg MS-SSIM	Avg bpp
JCT-VC class C	Basketball drill	0.986	0.12
	BQ mall	0.977	0.10
	Party scene	0.984	0.15
	Race horses	0.703	0.12

Table 8: MS-SSIM on JCT-VC class D

Data set	Video	Avg MS-SSIM	Avg bpp
JCT-VC class D	Basketball pass	0.89	0.10
	Blowing bubbles	0.28	0.12
	BQ square	0.99	0.11
	Race horses	0.69	0.11

Table 9: MSE on UVG (PSNR metric)

Data set	Video	MSE
UVG	Beauty	8.3600
	Bosphorus	0.0001
	Honey bee	0.0002
	Jockey	0.0001
	Ready set go	0.0001
	Shaken dry	0.0002
	Yacht ride	0.0001

Table 10: MSE on JCT-VC class B (PSNR metric)

Data set	Video	MSE
JCT-VC class B	Basketball drive	0.001
	BQ terrace	0.001
	Cactus	0.001
	Kimono	0.001
	Park scene	0.001

DOI: 10.3844/jcssp.2022.589.598

Table 11: MSE on JCT-VC class C (PSNR metric)

Data set	Video	MSE
JCT-VC class C	Basketball drill	0.0001
	BQ mall	0.0001
	Party scene	0.0001
	Race horses	0.0001

Table 12: MSE on JCT-VC class D (PSNR metric)

Data set	Video	MSE
JCT-VC class D	Basketball pass	0.001
	Blowing bubbles	0.001
	BQ square	0.001
	Race horses	0.001

Table 13: MSE on UVG (MS-SSIM metric)

Data set	Video	MSE
UVG	Beauty	0.09
	Bosphorus	0.12
	Honey bee	0.07
	Jockey	0.06
	Ready set go	0.08
	Shaken dry	0.06
	Yacht ride	0.11

Table 14: MSE on JCT-VC class B (MS-SSIM metric)

Data set	Video	MSE
JCT-VC class B	Basketball drive	0.03
	BQ terrace	0.14
	Cactus	0.10
	Kimono	0.08
	Park scene	0.09

Table 15: MSE on JCT-VC class C (MS-SSIM metric)

Data set	Video	MSE
JCT-VC class C	Basketball drill	0.04
	BQ mall	0.05
	Party scene	0.05
	Race horses	0.07

Table 16: MSE on JCT-VC class D (MS-SSIM metric)

Data set	Video	MSE
JCT-VC class D	Basketball pass	0.00
	Blowing bubbles	0.07
	BQ square	0.16
	Race horses	0.07

Table 17: Compression ratio of UVG (PSNR metric)

Data set	Video	Size of the input (MB's)	Size of the output (MB's)	Compression ratio
UVG	Beauty	13.43	9.800	1.36
	Bosphorus	14.10	8.970	1.57
	Honey bee	19.04	14.560	1.30
	Jockey	15.68	11.410	1.16
	Ready set go	17.17	11.740	1.46
	Shaken dry	19.05	12.240	1.55
	Yacht ride	15.18	9.990	1.51

DOI: 10.3844/jcssp.2022.589.598

Table 18: Compression ratio of JCT-VC class B (PSNR metric)

Data set	Video	Size of the input (MB's)	Size of the output (MB's)	Compression ratio
JCT-VC class B	Basketball drive	15.69	11.80	1.32
	BQ terrace	16.37	11.97	1.36
	Cactus	17.24	11.83	1.45
	Kimono	16.92	11.55	1.46
	Park scene	17.70	11.22	1.57

Table 19: Compression ratio of JCT-VC class C (PSNR metric)

Data set	Video	Size of the input (MB's)	Size of the output (MB's)	Compression ratio
JCT-VC class C	Basketball drill	17.55	10.61	1.65
	BQ mall	15.92	11.50	1.38
	Party scene	19.57	13.03	1.50
	Race horses	17.83	17.02	1.04

Table 20: Compression ratio of JCT-VC class D (PSNR metric)

Data set	Video	Size of the input (MB's)	Size of the output (MB's)	Compression ratio
JCT-VC class D	Basketball pass	12.94	9.54	1.35
	Blowing bubbles	16.33	12.23	1.33
	BQ square	15.66	12.31	1.27
	Race horses	15.96	17.19	0.92

Table 21: Compression ratio of UVG (MS-SSIM metric)

Data set	Video	Size of the input (MB's)	Size of the output (MB's)	Compression ratio
UVG	Beauty	13.43	10.84	1.23
	Bosphorus	14.10	11.03	1.27
	Honey bee	19.04	14.23	1.33
	Jockey	15.68	12.83	1.22
	Ready set go	17.17	13.81	1.24
	Shaken dry	19.05	15.26	1.24
	Yacht ride	15.18	12.19	1.24

Table 22: Compression ratio of JCT-VC class B (MS-SSIM metric)

Data set	Video	Size of the input (MB's)	Size of the output (MB's)	Compression ratio
JCT-VC class B	Basketball drive	15.69	11.38	1.37
	BQ terrace	16.37	13.63	1.20
	Cactus	17.24	14.04	1.22
	Kimono	16.92	13.05	1.29
	Park scene	17.70	13.73	1.28

Table 23: Compression ratio of JCT-VC class C (MS-SSIM metric)

Data set	Video	Size of the input (MB's)	Size of the output (MB's)	Compression ratio
JCT-VC class C	Basketball drill	17.55	13.13	1.33
	BQ mall	15.92	12.95	1.22
	Party scene	19.57	15.41	1.26
	Race horses	17.83	15.77	1.13

Table 24: Compression ratio of JCT-VC class D (MS-SSIM metric)

Data set	Video	Size of the input (MB's)	Size of the output (MB's)	Compression ratio
JCT-VC class D	Basketball pass	12.94	11.54	1.12
	Blowing bubbles	16.33	13.99	1.16
	BQ square	15.66	14.52	1.07
	Race horses	15.96	15.15	1.00

Conclusion

In this study, we propose an DCNN-based efficient video compression framework. We also demonstrate how our DCNN technique outperforms both commonly used traditional video compression standards and more recent deep learning-based video compression solutions. Our proposed DCNN model offers a higher compression ratio and lower error rates because it enhances better video quality while using low bit rates.

Acknowledgment

We would like to express our gratitude to Dr. P. Bindhu Madhavi, Dr. K. Janaki, and the rest of my colleagues for their tremendous technical and spiritual support in helping me complete my task successfully.

Author's Contributions

Kommerla Siva Kumar: Investigation, problem formulation, methodology, formal analysis, software implementation, data curation, data analysis, original draft paper preparation.

P. Bindhu Madhavi: Supervision, design research plan, research administration, problem formulation, methodology, resources, writing review, draft paper correction and editing.

K. Janaki: Supervision, design research plan, research administration, problem formulation, methodology, resources, writing review, draft paper correction and editing.

Ethics

This manuscript substance is the author's original work and has not been previously published somewhere else. Authors already read and approved the manuscript and no potential ethical issues are immersed.

References

Balle, J., Laparra, V., & Simoncelli, E. P. (2016). End-to-end optimized image compression. arXiv preprint arXiv:1611.01704. https://arxiv.org/abs/1611.01704.

Balle, J., Minnen, D., Singh, S., Hwang, S. J., & Johnston, N. (2018). Variational image compression with a scale hyperprior. arXiv preprint arXiv:1802.01436. doi.org/10.48550/arXiv.1802.01436.

Bao, W., Lai, W. S., Zhang, X., Gao, Z., & Yang, M. H. (2019). Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. IEEE transactions on pattern analysis and machine intelligence, 43(3), 933-948. doi.org/10.1109/TPAMI.2019.2941941

Birman, R., Segal, Y., & Hadar, O. (2020). Overview of research in the field of video compression using deep neural networks. Multimedia Tools and Applications, 79(17), 11699-11722. doi.org/10.1007/s11042-019-08572-3

Chen, T., Liu, H., Shen, Q., Yue, T., Cao, X., & Ma, Z. (2017, December). Deep coder: A deep neural network-based video compression. In 2017 IEEE Visual Communications and Image Processing (VCIP) (pp. 1-4). IEEE. doi.org/10.1109/VCIP.2017.8305033

Chen, Z., He, T., Jin, X., & Wu, F. (2019). Learning for video compression. IEEE Transactions on Circuits and Systems for Video Technology, 30(2), 566-576. doi.org/10.1109/TCSVT.2019.2892608

Cheng, Z., Sun, H., Takeuchi, M., & Katto, J. (2019). Learning image and video compression through spatial-temporal energy compaction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10071-10080).

Dai, J., Au, O. C., Yang, W., Pang, C., Zou, F., & Liu, Y. (2009, October). Motion vector coding is based on predictor selection and boundary-matching estimation. In 2009 IEEE International Workshop on Multimedia Signal Processing (pp. 1-5). IEEE. doi.org/10.1109/MMSP.2009.5293331.

Guo, L., Ouyang, W., Xu, D., Zhang, X., Cai, C., & Gao, Z. (2019), For Testing (JCT-VC data set).

Habibian, A., Rozendaal, T. V., Tomczak, J. M., & Cohen, T. S. (2019). Video compression with rate-distortion autoencoders. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 7033-7042).

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. 2022, Article, 2770706, 10 p, 2022. doi.org/10.1155/2022/2770706.

Kumar, K. S., & Janaki. (2020) A review on video compression approaches and utilization of deep learning techniques", International Journal of Advanced Research in Engineering and Technology (IJARET), IAEMA Publication, 1211-1218, 11, 9. doi.org/10.34218/IJARET.11.9.2020.120.

Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., & Gao, Z. (2019). Dvc: An end-to-end deep video compression framework. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11006-11015).

Ranjan, A., & Black, M. J. (2017). Optical flow estimation using a spatial pyramid network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4161-4170).

Wu, X., Zhang, Z., Feng, J., Zhou, L., & Wu, J. (2020). End-to-end optimized video compression with mv-residual prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 156-157).

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

Yang, R., Mentzer, F., Gool, L. V., & Timofte, R. (2020a). Learning for video compression with hierarchical quality and recurrent enhancement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 6628-6637).

Yang, R., Van Gool, L., & Timofte, R. (2020b). Open DVC: An open-source implementation of the DVC video compression method. arXiv preprint arXiv:2006.15862. doi.org/10.48550/arXiv.2006.15862