Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

Video Compression for Surveillance Application using Deep Neural Network

Prasanga Dhungel, Prashant Tandan, Sandesh Bhusal, Sobit Neupane, Subarna Shakya,
Department of Electronics and Computer Engineering,
Central Campus, Institute of Engineering, Pulchowk,
Tribhuvan University, Pulchowk
Lalitpur Nepal - 44600
Email: drss@ioe.edu.np

Abstract: We present a new approach to video compression for video surveillance by refining the shortcomings of conventional approach and substitute each traditional component with their neural network counterpart. Our proposed work consists of motion estimation, compression and compensation and residue compression, learned end-to-end to minimize the rate-distortion trade off. The whole model is jointly optimized using a single loss function. Our work is based on a standard method to exploit the spatio-temporal redundancy in video frames to reduce the bit rate along with the minimization of distortions in decoded frames. We implement a neural network version of conventional video compression approach and encode the redundant frames with lower number of bits. Although, our approach is more concerned toward surveillance, it can be extended easily to general purpose videos too. Experiments show that our technique is efficient and outperforms standard MPEG encoding at comparable bitrates while preserving the visual quality.

Keywords: Video compression, Motion Estimation, Auto-encoder, Rate-Distortion minimization, Bitrate Estimation

1. Introduction

The use of surveillance cameras is increasing rapidly in countries all over the world. There are an estimated 770 million surveillance cameras installed around the world today and the number is estimated to rise to around a billion in 2021. The huge interest in this technology can be attributed to its utility in public safety, reduction of crime rate and so on. However, the system has some caveats as well.

A large amount of temporal redundancy can be noticed in most of the surveillance recordings as the surroundings are mostly static and only a small part of the frame changes contributed by moving objects. So, a large percentage of bandwidth is expended transmitting scenes of little or no interest because they do not contain objects of semantic interest (e.g. people, vehicles, animals). Systems with a large number of surveillance cameras typically stream raw video feeds from all cameras to a central server for observation, analysis and processing. It puts a high processing load on the central server. Moreover, the streams need to be archived in

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

most of the cases for future use which consumes a huge amount of storage. All of these issues of the surveillance system can be solved by using a video compression mechanism which exploits the spatio-temporal redundancies prevalent in the streams preserving the important details to render their further analysis and storage useful.

There are many existing video compression standards which are used widely such as MPEG [1]. H.264[2] that uses mathematical techniques for video compression. While they have been very well engineered and thoroughly tuned, they are designed to be used in a general sense, which limits their ability to be specific on surveillance applications.

Deep learning approaches have been applied to solve many problems including the field of image compression which has made huge advancements. ML-based image compression approaches [3, 4] have been surpassing the commercial codecs by significant margins, and are still making significant progress. The advancements in the hardware field to accommodate the machine learning based applications have further boosted this process. But image compression is mainly based on spatial estimation but for video compression, motion information should also be considered.

Deep learning approaches for video compression are being developed for the whole compression pipeline [5, 6, 7] or for some parts of the traditional pipeline [8, 9, 10]. One good approach to our problem can be to detect interesting portions of video frames and allocate more bits in the compressed representation to them. The rest of the frame is allocated fewer bits in the compressed stream by smoothing away details that would otherwise be unnecessarily encoded in the transmitted video. Despite having robust and accurate object detection algorithms, this approach will make the solution highly specific.

Our solution is to apply an end-to-end deep learning model to compress the video streams of surveillance systems with robustness and low memory requirements.

1.1 Brief Introduction of Video Coding

- **Motion Estimation**: Motion between current frame and previous frame is estimated (as optical flow or motion vector).
- **Motion Compensation**: Based on the motion vector or optical flow obtained at motion estimation step, pixels on the previous frames are moved to obtain the motion compensated frame.

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

- **Residual Compression**: The leftover difference between the targeted frame and motion compensated frame is obtained and compressed.
- **Motion Compression**: Motion information obtained from the motion estimation step is compressed and sent for encoding.
- **Encoding**: Residue obtained after compression and motion information obtained after compression are encoded and sent to the decoder.
- **Frame Reconstruction**: Motion compensated frame from motion compensation step and residue at residue compression steps are added to obtain a reconstructed frame.

2. Related Works

2.1 Traditional Video Compression Pipeline

Historically, video used to be stored on magnetic tapes. Since the introduction of Discrete Cosine Transform by Nassir Ahmed, et. al. [11] in 1974, it has been widely used in video compression tasks, leading to the development of H.261, which can be attributed as the first practical video coding standard. The most popular and widely used video compression standard today is H.264, which was developed by a number of prominent players in the video coding field.

Video compression methods generally exploit the redundant responses of the Human Visual System (HVS). Traditional video compression methods make usage of the Y:Cr:Cb Color space that transforms the sampled spatial sequence of Red, Green and Blue into Luma and 3 Chroma components. The HVS is highly responsive to the luma components, but gives less of a preference to the chroma components. Only two chroma components can be transferred in the encoded signal, as the third component can be derived as a combination of the remaining components and luma.

2.2 Neural Network Models in Video Compression

Neural Networks are being tested and tried focusing on the task of video compression. However, not much breakthrough has been attained till date. The Neural Network based models aim at optimizing portions of the traditional video compression pipeline. Traditional video compression has many components that can be optimized through the usage of Neural Networks, since they are excellent tools because of their nonlinear activation function that help to predict the different types of data required while compressing videos. The research activity is primarily focused into two groups:

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

- Learning-based optimization modules combined with traditional video codecs
- End to End learned codecs

The neural networks that are primarily used for video compression are Fully connected (FC – suited due to their regression capabilities), Convolutional Neural Networks (CNN – suited due to their ability to find patterns in data) and LSTMs, which are suited due to their capability of exploiting historical data. Although neural networks have a massive computation requirement, this is primarily in the training phase, and hence the codec versions of the models have only a forward pass phase, which can have bounded reasonable complexity. In the following subsection, we explore some of the major progresses in the field of video compression using deep learning.

2.2.1 Intra prediction modes

Video frames in traditional codecs are divided as Group of Pictures (GOP), which are composed of I, P and B Frames, which denote the first image still, the frame estimated by using motion compensation among blocks, and the frame produced by bidirectional correlation computation between temporally separated frames. The input video frame is initially partitioned into Macro-Blocks and then Intra-Prediction coding is used, contributing to the reduction of Intra frames (I-frames), which, in general, are the most demanding frames in terms of bitrate. This is achieved by finding correlations between previously scanned pixels and the next block in the same frame. Once correlation is found, the next block pixels are predicted and only the differences (Residual Errors) are sent, thus improving compression efficiency. Intra- Prediction algorithms apply different modes to predict block pixel values. The modes essentially predict block pixel values by applying different directional functions on the surrounding pixels, functions that take advantage of frame image gradients [12].

In practice, all modes are being evaluated per block and the one that provides the lowest Mean-Square-Error (MSE), e.g., best prediction, is selected. The decoder is notified of the selected mode using signaling. Some new modes and revised scanning algorithms have been proposed by Ofer Hadar et al. in [13].

Intra prediction is the task of predicting the current block based on the blocks that are already decoded. In a traditional raster-scan order, the macroblocks are decoded top to bottom left to right order. Neural Networks excel in prediction tasks, and therefore in the last three years researchers have been exploring the potential of using them for better predictions.

Improving Intra-Prediction with Deep Learning has taken few different possible approaches:

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

- Predicting the most suitable standard mode per block to prevent extensive MSE calculations at the encoder [14]. The authors use a classification Convolutional Neural Networks (CNN) and supervised learning to analyze image blocks and train the network to predict the most likely optimal HEVC mode. The considered modes are the 33 angular Intra-Prediction modes, the DC mode and the Planar mode. The network is trained on 32 × 32 blocks labeled according to the best selected mode.
- Using neighboring blocks for predicting block residual errors and correcting the values to accomplish an improved (lower) residual error [15]. The authors train CNNs for predicting residual error between HEVC predicted blocks and the original pixel values. The block prediction is performed for 8 × 8 Prediction Unit (PU) blocks. The network input is the 3 adjacent HEVC Intra-Prediction 8 × 8 blocks (top and left of the predicted block). The network is trained to predict the residual error between the target predicted block and the original values, thus providing additional prediction accuracy correction to standard calculated modes. The trained network is used for correcting prediction residual error.

2.2.2 Inter prediction with Neural Network models

The most prevalent research direction for improving Inter-Prediction has focused on using Neural Networks and in particular CNNs, to capture matching blocks features and using them for improving the predicted block, thus reducing the Inter-Prediction residual error. The prediction of B-Frame is using a bi-directional average of forward and backward frame blocks. A method was proposed to use a CNN network that performs a more accurate weighing for bi-directional motion compensation [16]. Pair of past and future frames are used for training a CNN network to combine them and accomplish a more accurate predicted frame than the one traditionally calculated using simple averaging between them. The proposed method is claimed to have accomplished up to 10.5% BD-rate savings and an average of 3.1% BD-rate savings compared to HEVC. Another method to improve Inter-Prediction accuracy has combined temporal and spatial redundancies. CNN and FC networks were trained to predict any motion compensated (Inter- Prediction) block pixel values from the standard identified motion compensated block in a previous frame as well as neighboring block pixels of the same frame [17]. Combining the pixels from temporal and spatial domains into a network input layer and training the network, has yielded better prediction results than using the simple motion compensated block alone. The proposed method is reported to have accomplished an average of 5.2% BD-Rate reduction compared to HEVC.

Another research direction that has been proposed is improving the calculation efficiency of fractional/sub-pixel motion estimation. The conventional codecs improve temporal block predictions by calculating sub-pixel matches between blocks, as opposed to integer pixel matches. Neural Networks have been used for predicting the best matched Prediction Block (PB) in a reference frame with integer as well as with sub-pixel pixel

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

precisions, thus eliminating the need to perform extensive interpolations that are required by conventional codecs for searching and finding the best sub-pixel match and obtaining better predictions despite the video signal not being low-passed nor stationary by nature. [18, 19]

2.2.3 End to End Deep Learning models

In some cases, an end-to-end scheme for video compression is using RD as a loss function, thus optimizing the network for best tradeoff between bitrate and quality. Few research works have followed this direction [19, 20]. [19], takes this interesting approach to image compression by devising optimization criteria that maximizes Rate while also minimizing Distortion.

Another distinctive approach has used several networks for mimicking the standard codec structure, by replacing complete functional blocks with Neural Networks. They further replace the MV scheme by Optical Flow (OF), which is expected to be more accurate, while avoiding the extensive data volumes for pixel-by-pixel movement representation by extracting distinctive features of the OF map using CNNs [6].

Video codecs for surveillance tasks are varied and different manufacturers use different codecs according to their requirements. Due to the low processing power of the common surveillance systems, the desired system should have low computational cost while producing a better output. JPEG, M-JPEG are used, which produce videos with blocky artifacts and large memory footprint. H.263 and H.261 are used but they produce inefficient performance in the face of high bandwidth availability, while MPEG-2 and MPEG-4 have an extremely large bandwidth requirement; when the bitrate is limited, they produce blocky artifacts in the output stream. H.264 is a state-of-the-art method used to compress videos, but has a large computational requirement. In this way, all available systems have their own drawbacks and strengths.

3. Proposed Work

We replace all of the components in traditional video compression with the neural network and jointly optimize the whole network with a single loss function. Fig. 1 gives an overview of our framework.

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

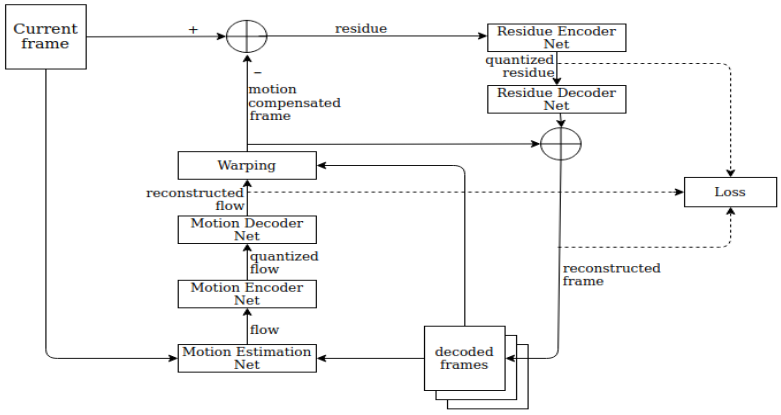


Figure 1: Proposed end-to-end video compression framework. There is one-to-one mapping between the traditional video compression pipeline and our proposed framework

3.1 Flow Estimation

A video can be thought of as a collection of frames $x_1, \ldots, x_T \in \mathbb{R}^{3\times H\times W}$. Estimating optical flow can be thought of as an optimization problem, and several machine learning techniques have been applied for its estimation. We estimate a learnable flow $f_t \in \mathbb{R}^{2\times H\times W}$ between the current frame x_t and previously reconstructed frame $\hat{x}_{t-1}$ such that copying the corresponding pixels in $\hat{x}_{t-1}$ according to f_t would resemble x_t as much as possible. Encouraged by Lu et al.,[6], we compute the motion using $\hat{x}_{t-1}$, instead of x_{t-1} so that we can transmit an initial frame to the decoder, and the decoder can reconstruct the rest of the frame from the previously reconstructed frame in its site. Estimating flow from $\hat{x}_{t-1}$ results lower distortion at the decoder as it eliminates the accumulation of the error at each encoding-decoding step.Traditional video codecs mathematically compute the optical flow and make a variety of assumptions about the frames, from brightness constancy to spatial smoothness and small movement. They are not only slow but also have high memory requirements. Along with preciseness, time and memory efficiency, video compression task requires the optical flow and the residue it generates to be more compressible.Optical flow estimation in [9] and it's later version [21] are precise but they have large model size and are slow with high memory usage consequently. Taking these things into account, we use a CNN model in [10] to estimate the optical flow.

3.2 Flow Compression

Classical frameworks of video compression either directly encode the raw optical flow. Directly compressing optical flow values will significantly increase the number of bits required for storing motion information. Other

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

conventional methods linearly transform the motion information after splitting the motion information into fixed size blocks.

Operations such as DCT, multi-scale orthogonal wavelet decomposition are used for the linear transformation. Some architectures however use variable size blocks [22] at the encoder and can more efficiently compress large homogeneous blocks. Such systems need to convey the partition structure to the decoder as a side information. Block based system approach also introduces blockness artifacts in the output.

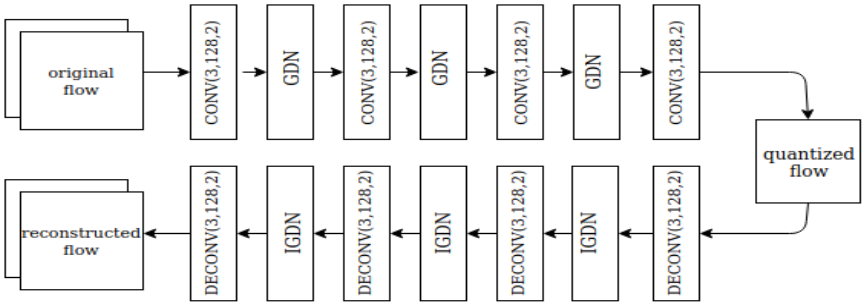


Figure 2: Our motion encoder-decoder network. CONV(a,b,c): affine convolution with filter support ($a \times a$), b output channels and downsampling factor of c in encoder and decoder respectively. The upper portion is the part of the encoder and lower portion is the part of the decoder. DECONV(a,b,c): Inverse of Convolution, but here c is an upsampling factor. GDN/IGDN: Generalized Divisive Normalization [23] across channels, and its inverse.

The model we present in this paper replaces the conventional linear transform with a highly non-linear flow encoder network that maps f_t into latent representation v_t. Our work is inspired by [24] with a minor change in number of channels. Our approach can better exploit the power of nonlinear transform and can achieve better compression efficiency compared to the linear transforms. In surveillance recordings, the consecutive frames are almost similar with the small motion of a few pixels. Also, the movement of many pixels are somewhat similar. The proposed transformation can identify such motion and utilize it for better compression. Entropy coding relies on discrete entropy so the cascaded transformation is followed by a quantization step after which the latent representation v_t is quantized (each element is rounded to the nearest integer) into $\hat{v}_t$.However, the quantization operation being discontinuous is not differentiable and as a consequence produces zero gradients backward, which makes end-to-end training impossible. To resolve the problem, we use the method in [24] and replace the quantization operation by adding uniform noise of width 1 and centered around 0. The quantized representation $\hat{v}_t$ is approximated as: $\hat{v}_t$ = U(vt,1), where U(m,n) is the uniform density of width n and centered on m. Later, in the inference stage, we use the rounding operation explicitly. The swapping of operation is possible because rounding operation can be thought of as adding a deterministic form of U(0,1). The quantized motion information can be compressed with lossless entropy code and the resulting bit stream $b_t \in \{0,1\}^{l(b)}$ is transmitted to the decoder. The number and size of filters, downsampling factors, and connectivity from layer

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

to layer are provided in Fig. 2. Currently, we do not provide the entropy coding module. It is straightforward to carry out entropy coding by using a framework, such as CABAC [25].

3.3 Motion Compensation

Motion information $\hat{f}_t$ is reconstructed from $\hat{v}_t$. We rely on standard warping function with bilinear interpolation to obtain the predicted frame $\bar{x}_t$ i.e $\bar{x}_t = w(\hat{x}_{t-1}, \hat{f}_t)$. The goal of the step is to leverage the temporal redundancy in the consecutive frames. The reconstructed flow ft is used instead of original flow ft to make sure that error in optical flow encoding-decoding is compensated in residue. Residue $r_t = x_t - \bar{x}_t$ is the difference between actual frame and predicted frame. The obtained residue has low energy and most of the pixels are zero and thus easily compressible. This phenomenon is more apparent in surveillance work.

3.4 Residual Compression

Similar to motion information we non-linearly transform the residual with autoencoder style in [24]. Lu et al.,[6] uses the CNN in [26] that uses variational autoencoder [27] style. The model in [26] regardless of producing better results, is large thus slow and requires large amounts of memory which doesn't make it suitable for surveillance purposes. Instead, we rely on [24] to transform r_t into latent space y_t. The encoder-decoder network is similar to Fig. 2 with a change in number of input and output channels. Quantization operation is carried out similar to 3.2 to obtain y_t , and the resulting bit stream $e_t \in \{0,1\}^{l(e)}$ is sent to the decoder

3.5 Frame Reconstruction

Motion information $\hat{f}_t$ and residue $\hat{r}_t$ are reconstructed from the bit streams b_t and e_t using their respective decoder network. Previously reconstructed frame $\hat{x}_{t-1}$ is warped with the reconstructed flow that results the predicted frame which is added to reconstructed residue to obtain the current reconstructed frame.i.e

$$\hat{x}_t = w(\hat{x}_{t-1}, \hat{f}_t) + \hat{r}_t$$

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

3.6 Experimental setup

3.6.1 Loss Function

Our objective is to minimize a weighted sum of rate and distortion, $R + \lambda D$, where governs the tradeoffs between the two terms. Our loss function is: $H(\hat{v}_t) + H(\hat{y}_t) + d(x_t, \hat{x}_t)$ where $d(x_t, \hat{x}_t)$ is the reconstruction loss and $H(.)$ represents the number of bits used for encoding the representations. The actual rates achieved by a properly designed entropy code are only slightly larger than the entropy [28], and thus we define the rate directly in terms of entropy. We estimate the probability distribution of $\hat{f}_t$ and $\hat{r}_t$ using technique in [24]. There are various metrics to penalize discrepancies between the reconstructed frame $\hat{x}_t$ and its target x_t such as

$$MSE = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}[x(i,j)-\hat{x}(i,j)]^2}{H \times W}$$

$$PSNR = 20 \log_{10}\left(\frac{1}{\sqrt{MSE}}\right)$$

$$MS-SSIM(x,y) = [l_m(x,y)]^{\alpha}\prod_{m=1}^{M}[c_m(x,y)]^{\beta}[s_m(x,y)]^{\gamma}$$

Where l_m, c_m, s_m are luminance similarity, contrast similarity and structural similarity components.

Like [7], we use Multi-Scale Structural Similarity Index (MS-SSIM) [29] which is been designed for and is known to match the human visual system significantly better than alternatives such as Peak Signal to Noise Ratio (PSNR) or l_p type losses such as MSE.

3.6.2 Datasets

Our training set comprises realistic and natural frames from VIRAT video dataset [30]. We spatially downsampled the frames to 256 × 256 for the training.

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

3.6.3 Training Procedure

We transmit the x_1 as it is and reconstruct $\hat{x}_2$ using x_1, $\hat{f}_1$ and $\hat{r}_1$. At the encoder, we estimate f_2 using $\hat{x}_2$ and x_3, transmit corresponding $\hat{v}_2$and $\hat{y}_2$ to the decoder. Decoder side already has $\hat{x}_2$ and using $\hat{v}_2$and $\hat{y}_2$, it can obtain $\hat{x}_3$. In other words, both encoding and decoding is carried out at the encoder. Previous reconstructed frame being used in encoding-decoding current frame means that the graph used in the previous frame is still in the memory. During optimization, gradients can flow in the graph of the previous frame i.e previous flows and residues are changed in order that it reduces the distortion and bit-rate not only in the previous frame but also in the current frame. So, for frames x_t: x_1, $\hat{x}_2$, $\hat{x}_3$, . . . , $\hat{x}_{t-1}$ are required which is not time and memory efficient. To resolve this problem, we detach the graph of the previous frame and only use the value of the previous reconstructed frame for current frame so that only $\hat{x}_{t-1}$ is in buffer during the training procedure of $\hat{x}_t$.

3.6.4 Implementation Details

We have trained the model with λ = 1024. We optimize the model with Adam [31] with an initial learning rate of 0.001 and default parameters β_1 = 0.9 and β_2 = 0.999. The learning rate is reduced by a factor of 10, thrice during training based on stability of loss. The whole system is implemented in PyTorch.

4. Results and Discussion

In our dataset we obtained an average MS-SSIM of 0.98 and average PSNR of 36dB along with 0.48 average bitrate. Our approach outperforms standard MPEG in terms of MS-SSIM and PSNR and is comparable to standard H.264 in terms of MS-SSIM if the frames are similar to the frames in our dataset.

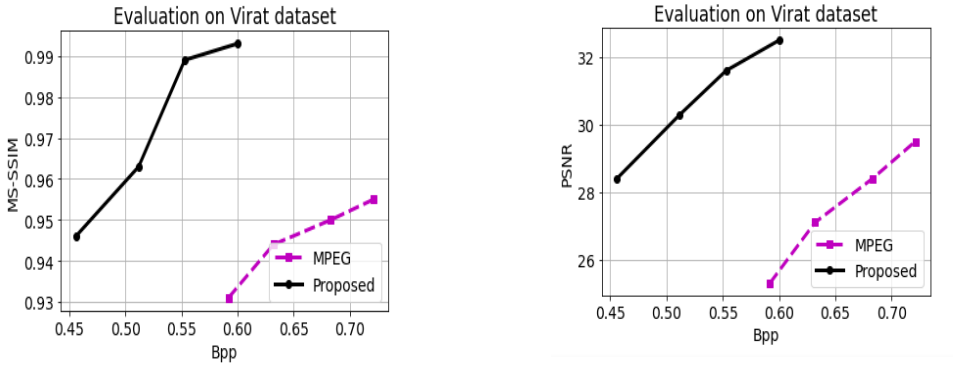
ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

. (a) (b)

Figure 3(a): Original(target) frame, 3(b): Reconstructed frame using proposed work

Although trained on MS-SSIM, our proposed model outperforms the standard MPEG model in terms of PSNR as well. More importantly, we observe an improvement in visual quality of reconstructed frames at a lower bitrate. Comparison between our model and MPEG is provided in Fig 4.



(a) (b)

Figure 4(a): Comparison between our model and MPEG based on MS-SSIM, 4(b): Comparison between our model and MPEG based on PSNR

Due to the small model and less parameters, our framework is faster and more memory efficient than [19, 24].

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

Because of one-to-one correspondence to conventional video compression approach,our proposed framework can be used as a reference model for video compression using deep learning. With the exploration of bigger dataset, our model can be used for compressing not just surveillance videos but also the videos comprising other varieties of scenes.

5. Conclusion

This paper has provided an overview of a learned video compression pipeline geared towards the compression of surveillance videos. To reduce the errors introduced in the pipeline by partial conversion of traditional video compression components, we have proposed a deep end-to-end model that can generalize well on the video surveillance dataset. The network is optimized to control the rate vs distortion problem occurring in video compression methodologies.

Our deep codec is simple and performs on par with H.264 codec and outperforms MPEG codec. However, we have not yet considered this system through a full implementation on a generalized video compression task. That work remains as a future research and extension.

References

[1] Le Gall, D. (1991). Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4), 46–58.

[2] Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7), 560–576.

[3] Liu, H., Chen, T., Shen, Q., Yue, T., & Ma, Z. (2018). Deep image compression via end-to-end learning., In *Cvpr workshops*.

[4] Toderici, G., Vincent, D., Johnston, N., Jin Hwang, S., Minnen, D., Shor, J., & Covell, M. (2017). Full resolution image compression with recurrent neural networks, In *Proceedings of the ieee conference on computer vision and pattern recognition*

[5] Kim, S., Park, J. S., Bampis, C. G., Lee, J., Markey, M. K., Dimakis, A. G., & Bovik, A. C. (2020). Adver-sarial video compression guided by soft edge detection, In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE

[6] 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 conference on computer vision and pattern recognition.

[7] Rippel, O., Nair, S., Lew, C., Branson, S., Anderson, A. G., & Bourdev, L. (2018). Learned video compression, arXiv 1811.06981

ISSN: 2582-2012 (online)

Journal of Artificial Intelligence and Capsule Networks (2020)
Vol.02/ No. 02
Pages: 131-145
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2020.2.006

[8] Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion, In *Scandinavian conference on image analysis*. Springer.

[9] Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., Van der Smagt, P., Cremers, D., & Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv*:*1504.06852*

[10] 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*.

[11] N. Ahmed, T. N., & Rao, K. R. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1), 90–93.

[12] Sullivan, G. J., Ohm, J., Han, W., & Wiegand, T. (2012). Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1649–1668.

[13] Hadar, O., Shleifer, A., Mukherjee, D., Joshi, U., Mazar, I., Yuzvinsky, M., Tavor, N., Itzhak, N., & Birman, R. (2017). Novel modes and adaptive block scanning order for intra prediction in AV1 (A. G. Tescher, Ed.). In A. G. Tescher (Ed.), *Applications of digital image processing xl*, SPIE. International Society for Optics and Photonics. https://doi.org/10.1117/12.2274035

[14] Laude, T., & Ostermann, J. (2016). Deep learning-based intra prediction mode decision for hevc, In 2016 *picture coding symposium (pcs)*.

[15] Cui, W., Zhang, T., Zhang, S., Jiang, F., Zuo, W., & Zhao, D. (2018). Convolutional neural networks based intra prediction for hevc.

[16] Zhao, Z., Wang, S., Wang, S., Zhang, X., Ma, S., & Yang, J. (2018). Cnn-based bi-directional motion compen-sation for high efficiency video coding, In 2018 *ieee international symposium on circuits and systems (iscas)*.

[17] Lee, J. K., Kim, N., Cho, S., & Kang, J. (2018). Convolution neural network based video coding technique using reference video synthesis, In 2018 *asia-pacific signal and information processing association annual summit and conference (apsipa asc).*

[18] Liu, J., Xia, S., Yang, W., Li, M., & Liu, D. (2019). One-for-all: Grouped variation network-based fractional interpolation in video coding. *IEEE Transactions on Image Processing*, 28(5), 2140–2151.

[19] Ibrahim, E. M., Badry, E., Abdelsalam, A. M., Abdalla, I. L., Sayed, M., & Shalaby, H. (2018). Neural networks based fractional pixel motion estimation for hevc, In 2018 *ieee international symposium on multimedia (ism).*

[20] Jiang, F., Tao, W., Liu, S., Ren, J., Guo, X., & Zhao, D. (2018). An end-to-end compression framework based on convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10), 3007–3018.

[21] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks, In *Proceedings of the ieee conference on computer vision and pattern recognition*

[22] Schwarz, H., Marpe, D., & Wiegand, T. (2007). Overview of the scalable video coding extension of the h. 264/avc standard. *IEEE Transactions on circuits and systems for video technology*, 17(9), 1103–1120.

[23] Ballé, J., Laparra, V., & Simoncelli, E. P. (2015). Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*.

[24] Ballé, J., Laparra, V., & Simoncelli, E. P. (2016). End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704.*

ISSN: 2582-2012 (online)

[25] Marpe, D., Schwarz, H., & Wiegand, T. (2003). Context-based adaptive binary arithmetic coding in the h. 264/avc video compression standard. *IEEE Transactions on circuits and systems for video technology*, 13(7), 620–636.
[26] Ballé, J., Minnen, D., Singh, S., Hwang, S. J., & Johnston, N. (2018). Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436.*
[27] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114.*
[28] Rissanen, J., & Langdon, G. (1981). Universal modeling and coding. *IEEE Transactions on Information Theory*, 27(1), 12–23
[29] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600–612.
[30] Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C.-C., Lee, J. T., Mukherjee, S., Aggarwal, J., Lee, H., Davis, L. Et al. (2011). A large-scale benchmark dataset for event recognition in surveillance video, In *Cvpr2011*. IEEE.
[31] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Authors Biography

The paper presented above is a part of major project by 4th year students (**Prasanga Dhungel, Prashant Tandan,Sandesh Bhusal,** and **Sobit Neupane)** at Pulchowk Campus, IOE. They belong to the Department of Electronics and Communication Engineering. The project was successfully supervised by Prof. Dr. Subarna Shakya who is a professor at Pulchowk Campus. Through there major project, they thought of helping the society in technical way. With this idea, they came across the concept standard method to exploit the spatio-temporal redundancy in video frames to reduce the bit rate along with the minimization of distortions in decoded frames toward the surveillance and other general purpose videos. The supervisor was helpful enough throughout this journey of making the concept turn into application which may help the targeted audience in some way

SUBARNA SHAKYA has received the MSc and PhD degrees in Computer Engineering from the Lviv Polytechnic National University, Ukraine, 1996 and 2000 respectively. He is the Professor of Computer Engineering, Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Pulchowk, Tribhuvan University, Nepal. He has served as Executive Director at National Information Technology Centre, Government of Nepal.His areas of research interests include E-Government system, Computer Systems &Simulation, Cloud computing& security, Software Engineering & Information System, Computer Architecture, Multimedia system**.**

ISSN: 2582-2012 (online)