Invertible Neural Network-Based Video Compression

Zahra Montajabi, Vahid Khorasani Ghassab and Nizar Bouguila
Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada

Keywords: Computer Vision, Deep Neural Networks, Invertible Neural Network (INN), Video Coding.

Abstract: Due to the recent advent of high-resolution mobile and camera devices, it is necessary to develop an optimal solution for saving the new video content instead of traditional compression methods. Recently, video compression received enormous attention among computer vision problems in media technologies. Using state-of-the-art video compression methods, videos can be transmitted in a better quality requiring less bandwidth and memory. The advent of neural network-based video compression methods remarkably promoted video coding performance. In this paper, an Invertible Neural Network (INN) is utilized to reduce the information loss problem. Unlike the classic auto-encoders which lose some information during encoding, INN can preserve more information and therefore, reconstruct videos with more clear details. Moreover, they don't increase the complexity of the network compared to traditional auto-encoders. The proposed method is evaluated on a public dataset and the experimental results show that the proposed method outperforms existing standard video encoding schemes such as H.264 and H.265 in terms of peak signal-to-noise ratio (PSNR), video multimethod assessment fusion (VMAF), and structural similarity index measure (SSIM).

1 INTRODUCTION

Nowadays, many web activities and web-based applications such as real-time communications and live streaming include a large number of video content. Video contents contribute to about 80% of the Internet traffic (Networking, 2016). As high-quality video content such as 4k videos become more prevalent, more sophisticated video compression methods are required to save the communication bandwidth/storage space while offering high-quality video encoding with less loss. Moreover, the higher quality of encoded videos enhances computer vision schemes such as object tracking and action recognition.

Among different recent methods developed for compression, deep learning-based compression methods gained more interest and attention. There are several traditional compression methods like JPEG (Wallace, 1992) and JPEG 2000 (Skodras et al., 2001), which are not optimized; because each module is optimized independently. In these methods, they map the input to latent feature representation linearly. In contrast, deep neural network-based methods are capable of using highly non-linear transformations and end-to-end training on a large scale to optimize compression. Some of the recent deep learning-based models are reviewed comprehensively in (Liu et al., 2021).

Recently, INNs got more popular than previous auto-encoder frameworks and they concentrate on learning the forward process, using additional latent output variables to capture the information that would otherwise be lost, in contrast to classical neural networks that attempt to solve the ambiguous inverse problem directly (Kingma and Dhariwal, 2018), (Dinh et al., 2016), (Ardizzone et al., 2018). Although autoencoders are very capable of choosing the significant information for reconstruction, some amount of information is completely lost; while using INN for encoding and decoding helps to preserve the information. Specific features of INN are: it has a bijective mapping between input and output and its inverse exists; it is possible to compute both forward and inverse mapping efficiently; and there is a tractable Jacobian of both mappings that makes it possible to calculate posterior probabilities explicitly. The work in (Ardizzone et al., 2018) demonstrated theoretically and practically that INNs are an effective analysis tool by using both synthetic data and real-world issues from astronomy and medicine categories. Another work on image generation (Ardizzone et al., 2019) used conditional INN architecture which performs better than variational autoencoders (VAEs) and generative adversarial networks (GANs). In (Lugmayr et al., 2020), INN is employed to more effectively address the ill-posed issue of super-resolution com-

Montajabi, Z., Ghassab, V. and Bouguila, N.
Invertible Neural Network-Based Video Compression.
DOI: 10.5220/0011807600003411
In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2023)*, pages 558-564
ISBN: 978-989-758-626-2; ISSN: 2184-4313

pared to GAN-based frameworks. Similarly, for image rescaling in (Xiao et al., 2020), an invertible bijective transformation is used to reduce the ill-posed nature of image upscaling.

In each image or frame of a video, there are some parts that are less important, and by compressing them more than other parts and using fewer bits for them, it is not easy to notice the difference with the original one unless by pixel by pixel comparison. Therefore, each image/video frame can be grouped into perceptually significant and perceptually insignificant areas. Sorting the areas based on significance is called texture analyzer which is used in many compression methods, and the inverse process is called texture synthesizer, which restores the pixels (Ding et al., 2021). This type of analysis/synthesis method is used in our video compression model. The proposed method, inspired by (Minnen et al., 2018), (Kingma and Dhariwal, 2018), (Xie et al., 2021), (Shi et al., 2016), (Cheng et al., 2020) incorporates four different modules: feature enhancement which is used to improve the nonlinear representation, INN which helps to reduce information loss, attention squeeze which is used to stabilize the training process instead of using unstable sampling technique (Wang et al., 2020), and hyperprior which performs analysis/synthesis and entropy coding.

To validate the performance of our method, we tested the model on the YouTube UGC video compression dataset (Wang et al., 2019) and reported the results by using PSNR, VMAF, and SSIM quality metrics under a similar setting. The results are reported and compared numerically and visually. The visual results demonstrate that under the same BPP, reconstructed video frames using our method has more clear details.

The paper is organized as follows. The video coding method is described in detail in Section 2. Experimental results are presented in Section 3, followed by concluding remarks in Section 4.

2 APPROACH

The proposed method consists of four modules: feature enhancement, INN, squeeze module, and main module. The architecture is shown in Fig. 1. Let $V = \{f_1, f_2, ..., f_T\}$ be the video sequences and f_t be the video frame f at time t. The input frame f has dimensions of (3, H, W). The first step, feature enhancement, adds non-linearity to each input frame and turns it to j with the same dimension of (3, H, W). The second step, INN, turns j to q with dimensions of $(3\times4^4, \frac{H}{2^4}, \frac{W}{2^4})$ in the forward pass. Third step, attention-squeeze module, leads q to turn into y with dimensions of $(\frac{3\times4^4}{\alpha}, \frac{H}{2^4}, \frac{W}{2^4})$ in which α is the compression ratio. For the rest of the model, the hyperprior of (Minnen et al., 2018) paper is used in which Minnen presented an autoregressive context model with a mean and scale hyperprior; In the hyperprior, using a mean and scale gaussian distribution, the quantized latent features $\hat{y}$ is parameterized with an analysis transform hyper encoder and a synthesis transform hyper decoder. The hyper encoder consists of three convolution layers with Leaky Relu activation between them. Analysis hyper encoder takes y to generate side information z and synthesis hyper decoder takes quantized side information $\hat{z}$. Asymmetric numeral system (ANS) (Duda, 2009) is used for entropy coding. Each frame loss is calculated based on the following equation (Minnen et al., 2018):

$$L = R(\hat{y}_t) + R(\hat{z}_t) + \lambda \cdot D(f_t, \hat{f}_t) = E[-\log_2 p_{\hat{y}}(\hat{y}_t)] + E[-\log_2 p_{\hat{z}}(\hat{z}_t)] + \lambda \cdot D(f_t, \hat{f}_t) \quad (1)$$

In which rate R is the entropy of quantized latent features, λ is the Lagrange multiplier and its different values correspond to different bit rates, D is the distortion term which can represent mean squared error (MSE) for MSE optimization, or $1 - MS\text{-}SSIM$ for MS-SSIM optimization (Wang et al., 2003), and here it represents mean squared error.

The inverse for the decoding process is as follows: squeeze module copies $\hat{y}$ for alpha times and reshapes it to $\hat{q}$. Then $\hat{q}$ is passed to the INN inversely and turned to $\hat{j}$ and finally, after the inverse of feature enhancement, the reconstructed video with frames $\hat{f}$ is generated. The detail of each module is elaborated in this section:

2.1 Feature Enhancement

Feature enhancement module is used to improve the nonlinear representation of the network because INNS are not often capable of nonlinear representation (Dinh et al., 2015). This module includes a Dense block (Huang et al., 2016) with input channel size of 3 and output channel size of 64, three convolution layers, and another Dense block with input channel size of 64 and output channel size of 3. Convolution layers have input channel size of 64, output channel size of 64, stride 1, and kernel sizes of 1, 3, and 1.

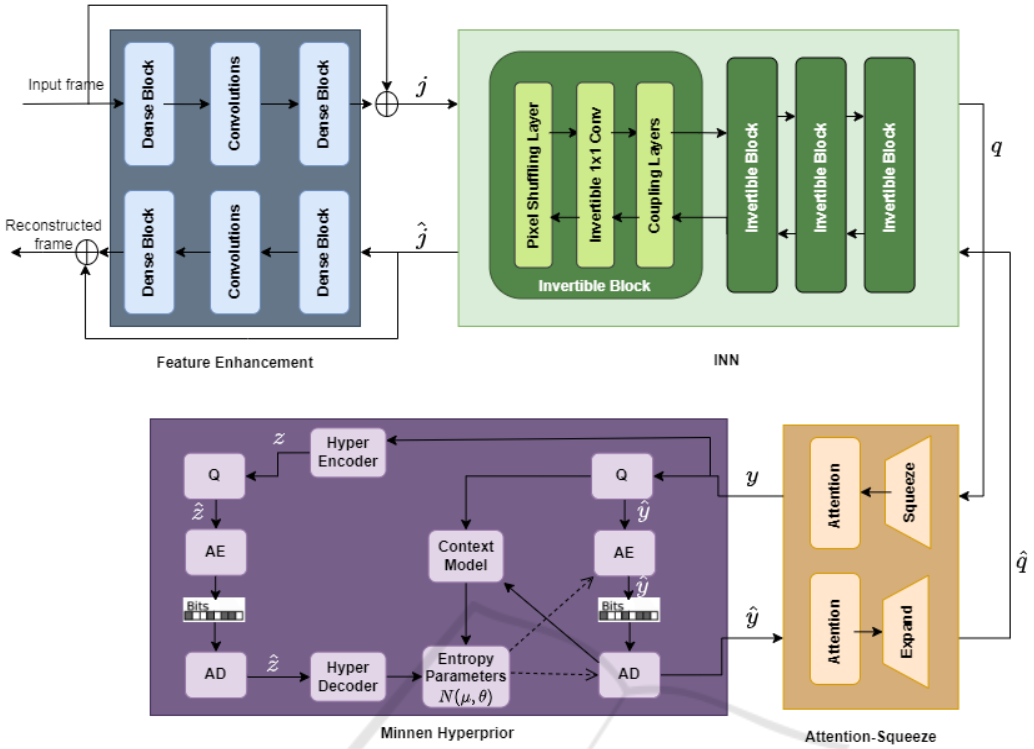


Figure 1: Structure of our video compressor.

2.2 INN

There are two invertible layers in the INN module which are the down-sampling layer and the coupling layer. A down-sampling layer includes a pixel shuffling layer (Shi et al., 2016) and an invertible 1x1 convolution layer (Kingma and Dhariwal, 2018). Four invertible blocks are utilized for down-sampling and up-sampling similar to the method proposed in (Minnen et al., 2018). Each of them consists of one down-sampling layer and three coupling layers. Using these four blocks, the input is down-sampled by 16 times (the down-sampling factor in each pixel shuffling layer is 2). Affine coupling layer (Dinh et al., 2016) is defined as following equations:

$$j^{i+1}_{t,1:d} = j^{i}_{t,1:d} \odot exp(\sigma_c(g_2(j^{i}_{t,d+1:D}))) + h_2(j^{i}_{t,d+1:D}) \quad (2)$$

$$j^{i+1}_{t,d+1:D} = j^{i}_{t,d+1:D} \odot exp(\sigma_c(g_1(j^{i+1}_{t,1:d}))) + h_1(j^{i+1}_{t,1:d}) \quad (3)$$

In the above equations, $j^{i}_{t,1:D}$ is the D dimensional input at time frame t to the i_{th} coupling layer which is divided into two parts with dimensions d and D-d. The functions h_1, h_2, g_1, g_1 are Convolutions with stride one with activation functions. Each of them consists of a convolution with kernel size 3, leaky relu, convolution with kernel size 1, leaky relu, and another convolution layer with kernel size 3. Also, $\odot$, exp, and σ_c show the Hadamard product, exponential, and center sigmoid functions respectively.

The inverse process is similar;

$$\hat{j}^{i}_{t,d+1:D} = (\hat{j}^{i+1}_{t,d+1:D} - h_1(\hat{j}^{i+1}_{t,1:d}) \odot exp(-\sigma_c(g_1(\hat{j}^{i+1}_{t,1:d})))) \quad (4)$$

$$\hat{j}^{i}_{t,1:d} = (\hat{j}^{i+1}_{t,1:d} - h_2(\hat{j}^{i}_{t,d+1:D})) \odot exp(-\sigma_c(g_2(\hat{j}^{i}_{t,d+1:D}))) \quad (5)$$

2.3 Attention-Squeeze Module

INN does not change the size of input while many of the pixels are useless for compression. So, the channel dimension of the INN's output is reduced through the Squeeze layer. If the output tensor of INN has the size of (D, H, W), it is reshaped into (r, $\frac{D}{r}$, H, W) in which r is the compression ratio. Then, it takes the average on the first dimension to turn the tensor into size ($\frac{D}{r}$, H, W) and passes it to the attention module; Attention module helps the model to pay more attention to challenging parts and reduce the bits of simple parts (Cheng et al., 2020). The inverse module in the decompression phase has an attention module

and then it copies the quantized tensor r times and reshapes it to the size of (D, H, W).

3 EXPERIMENTS

Experiments are performed on a Tesla V100-SXM2-16GB GPU. The network is trained for 550 epochs using a batch size of 16, learning rates of 0.0001 for the first 450 epochs and 0.00001 for the rest, with Adam optimizer (Kingma and Ba, 2015) on Pytorch framework.

3.1 Dataset

To train our video compression model, the Vimeo-90k dataset is used (Xue et al., 2019) which is a large dataset developed for different kinds of video processing tasks. The dataset includes 89800 clips with different contents in different categories. 5000 video clips from different categories are used to train our model. The resolution of the videos is cropped to 256 x 256 pixels.

To test the method and report the results, YouTube UGC Dataset (Wang et al., 2019) is used. There are 1500 video clips with the length of 20 seconds in different categories such as gaming, sports, animation, and lecture. Each clip is available in 4:2:0 YUV format and resolutions of 360P, 480P, 720P, and 1080P. To test the method, we selected 40 videos in different categories with 720P resolution. Also, another public dataset of Ultra Video Group (UVG) (Mercat et al., 2020) is used which includes 16 versatile 4K (3840×2160) video sequences. Each video is 50 or 120 frames per second available in 4:2:0 YUV format. To test the videos, the 1920 x 1080 resolution of the dataset is used.

3.2 Evaluation Method

To measure the performance of the proposed framework, peak-signal-to-noise ratio (PSNR), video multimethod assessment fusion (VMAF), and structural similarity index measure (SSIM) quality metrics are used. A higher PSNR value shows a higher image quality (Horé and Ziou, 2010), and SSIM (Dosselmann and Yang, 2005) measures the similarity between two images and is better correlated with the human perception of distortion. Compared to PSNR and SSIM, VMAF is a subjective measure of the human eye perception and so it is a pivotal measure in real-world applications (Rassool, 2017).

Table 1: Performance comparison of the methods applied to the YouTube UGC dataset. The proposed method outperforms others in terms of PSNR, VMAF, and SSIM.

Quality Metric	PSNR	VMAF	SSIM
Proposed	**45.5**	**98.9**	**0.982**
H.264	43.9	97.5	0.975
H.265	40.6	93.4	0.96

Table 2: Performance comparison of the methods applied to the UVG dataset. The proposed method outperforms others in terms of PSNR, VMAF, and SSIM.

Quality Metric	PSNR	VMAF	SSIM
Proposed	**43.9**	**97.4**	**0.971**
H.264	43.1	94.5	0.96
H.265	41.5	91.5	0.955

3.3 Results

The average PSNR, VMAF, and SSIM of the proposed model on the UGC dataset are 45.5, 98.9, and 0.982, respectively, which outperforms H.264 (Wiegand et al., 2003) and H.265 (Sullivan et al., 2012) as can be seen in Table 1. The average bit per pixel (BPP) for all test sets in the UGC dataset is 0.6.

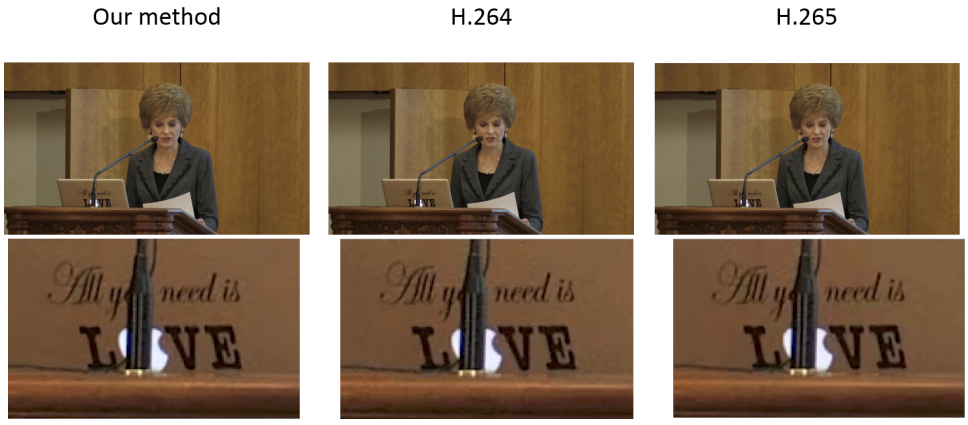
Fig. 2 shows samples of the outputs of our framework and H.264 and H.265 on the UGC dataset. In the first example of the lecturer 2a, the text on the laptop is sharper and more clear using our method. In the second example 2b, the edge of the shapes is more quantized in H.264 and more blurry in H.265. Also, in the third example of a television clip 2c, the face of the person is less quantized, sharper, and more clear using our method.

The average PSNR, VMAF, and SSIM of the proposed model on the UVG dataset are 43.9,97.4, and 0.971, respectively, which outperforms H.264 and H.265 as can be seen in Table 2. The average bit per pixel (BPP) for all test samples in the UVG dataset is 0.4.

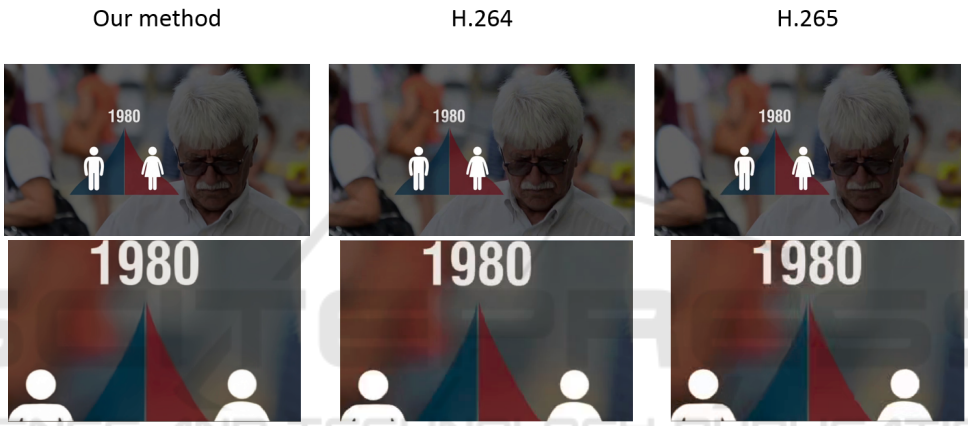
Fig. 3 shows samples of the outputs of our framework and H.264 and H.265 on the UVG dataset. In the first example of the honey bee among flowers 3a, the output of H.264 is more quantized and the honey bee in the output of H.265 is more blurry while using our method it is more clear and sharper. In the second example 3b, the numbers are more quantized and more blurry in H.264 and H.265 than in our method.

4 CONCLUSION

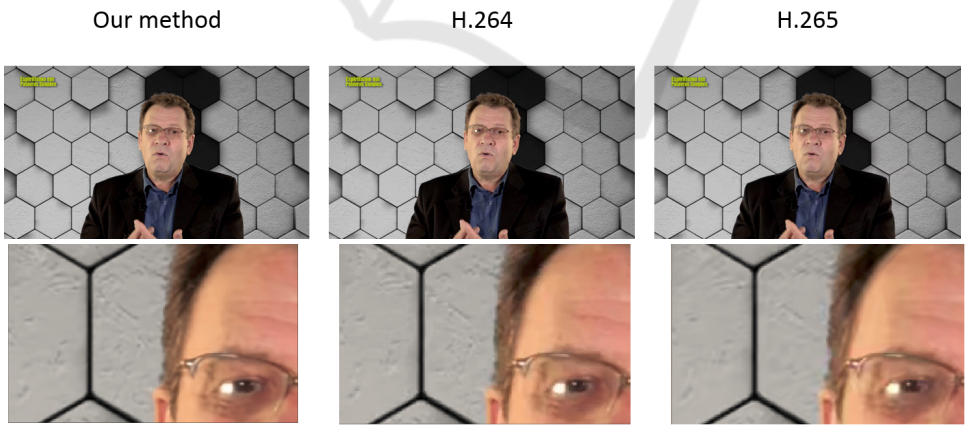
A video compression method based on INN has been introduced. First, a feature enhancement method is used for enhancing the nonlinear representation.



(a)



(b)



(c)

Figure 2: Samples of the proposed method, H.264, and H.265 outputs. The data used here are from the UGC dataset. (a) The text on the laptop is sharper and more clear using our method. (b) The edge of the shapes is more quantized in H.264 and more blurry in H.265. (c) The face of the person is less quantized, sharper, and more clear using our method.

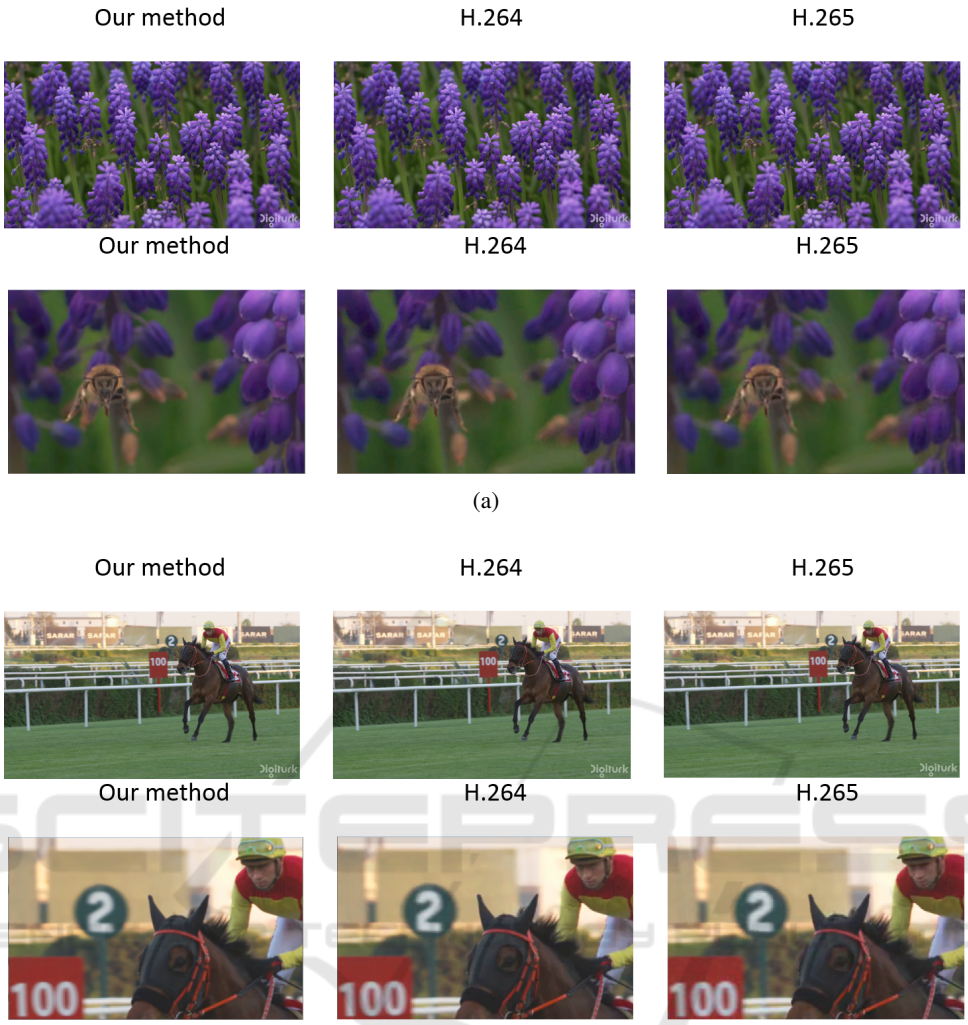


Figure 3: Samples of the proposed method, H.264, and H.265 outputs. The data used here are from the UVG dataset. (a) The output is sharper and more clear using our method than H.264 and H.265. (b) The numbers are more quantized and more blurry in H.264 and H.265 than in our method.

Then, INN is used to decrease the information loss problem. Compared with traditional auto-encoders which lose information in the encoding process, INN can persevere the information and leads to reconstructed videos with more clear details without making the network more complex. To solve the problem of unstable training in INNs, attention-squeeze module is used which makes the feature dimension adjustment stable and tractable.

The results of the method are reported and compared numerically and visually. Evaluations of the proposed method on two standard public datasets show better quality (under the same setting) in terms of PSNR, VMAF, and SSIM as compared to the recognized methods such as H.264 and H.265. The visual comparison of the output of the proposed method shows that it has more clear details than the outputs of H.264 and H.265.

To improve the method in future works, it may be possible to deploy more kinds of layers for feature enhancement module or even replace the newer published hyperpriors instead and use them with INNs. Also, the idea of current model can be used for similar applications such as video denoising.

REFERENCES

Ardizzone, L., Kruse, J., Wirkert, S. J., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C., and Köthe, U. (2018). Analyzing inverse

problems with invertible neural networks. *CoRR*, abs/1808.04730.

Ardizzone, L., Lüth, C., Kruse, J., Rother, C., and Köthe, U. (2019). Guided image generation with conditional invertible neural networks. *CoRR*, abs/1907.02392.

Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. (2020). Learned image compression with discretized gaussian mixture likelihoods and attention modules. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7936–7945.

Ding, D., Ma, Z., Chen, D., Chen, Q., Liu, Z., and Zhu, F. (2021). Advances in video compression system using deep neural network: A review and case studies. *Proceedings of the IEEE*, 109(9):1494–1520.

Dinh, L., Krueger, D., and Bengio, Y. (2015). Nice: Nonlinear independent components estimation. *CoRR*, abs/1410.8516.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using real NVP. *CoRR*, abs/1605.08803.

Dosselmann, R. and Yang, X. D. (2005). Existing and emerging image quality metrics. In *Canadian Conference on Electrical and Computer Engineering, 2005.*, pages 1906–1913.

Duda, J. (2009). Asymmetric numeral systems. *CoRR*, abs/0902.0271.

Horé, A. and Ziou, D. (2010). Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369.

Huang, G., Liu, Z., and Weinberger, K. Q. (2016). Densely connected convolutional networks. *CoRR*, abs/1608.06993.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. *ArXiv*, abs/1807.03039.

Liu, D., Li, Y., Lin, J., Li, H., and Wu, F. (2021). Deep learning-based video coding. *ACM Computing Surveys*, 53(1):1–35.

Lugmayr, A., Danelljan, M., Gool, L. V., and Timofte, R. (2020). Srflow: Learning the super-resolution space with normalizing flow. *CoRR*, abs/2006.14200.

Mercat, A., Viitanen, M., and Vanne, J. (2020). Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. *Proceedings of the 11th ACM Multimedia Systems Conference*.

Minnen, D. C., Ballé, J., and Toderici, G. (2018). Joint autoregressive and hierarchical priors for learned image compression. *ArXiv*, abs/1809.02736.

Networking, C. V. (2016). Cisco global cloud index: Forecast and methodology, 2015-2020. white paper. *Cisco Public, San Jose*, page 2016.

Rassool, R. (2017). Vmaf reproducibility: Validating a perceptual practical video quality metric. In *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–2.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *CoRR*, abs/1609.05158.

Skodras, A., Christopoulos, C., and Ebrahimi, T. (2001). The jpeg 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58.

Sullivan, G. J., Ohm, J.-R., Han, W.-J., and 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.

Wallace, G. (1992). The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv.

Wang, Y., Inguva, S., and Adsumilli, B. (2019). Youtube ugc dataset for video compression research. *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5.

Wang, Y., Xiao, M., Liu, C., Zheng, S., and Liu, T. (2020). Modeling lost information in lossy image compression. *CoRR*, abs/2006.11999.

Wang, Z., Simoncelli, E., and Bovik, A. (2003). Multiscale structural similarity for image quality assessment. volume 2, pages 1398 – 1402 Vol.2.

Wiegand, T., Sullivan, G., Bjontegaard, G., and 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.

Xiao, M., Zheng, S., Liu, C., Wang, Y., He, D., Ke, G., Bian, J., Lin, Z., and Liu, T.-Y. (2020). Invertible image rescaling. *ArXiv*, abs/2005.05650.

Xie, Y., Cheng, K. L., and Chen, Q. (2021). Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 162–170, New York, NY, USA. Association for Computing Machinery.

Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. (2019). Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125.