A LEARNED POST-PROCESSING MODEL WITH QUALITY-GATED CONVLSTM FOR VIDEO COMPRESSION

by

Hilal Güven

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical and Electronics Engineering

September 7, 2023

A LEARNED POST-PROCESSING MODEL WITH QUALITY-GATED CONVLSTM FOR VIDEO COMPRESSION

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Hilal Güven

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Ahmet Murat Tekalp

_____

Assoc. Prof. Aykut Erdem

_____

Assoc. Prof. Erkut Erdem

Date: _____

to my beloved mom, dad, and everyone who believes in me..

ABSTRACT

A LEARNED POST-PROCESSING MODEL WITH QUALITY-GATED CONVLSTM FOR VIDEO COMPRESSION

Hilal Güven

Master of Science in Electrical and Electronics Engineering

September 7, 2023

Recently, significant progress has been shown in applying deep learning to video compression and enhancement in terms of coding efficiency and quality improvement. In this work, a learned post-processing network is proposed for video compression and restoration tasks, which contains Quality-Gated Convolutional Long Short-Term Memory (QG-ConvLSTM) cells to enhance the quality of the compressed video frames considering the relative quality. With the proposed QG-ConvLSTM cell-based post-processing, the network exploits and takes full advantage of the inter-frame correlation and quality fluctuation between neighboring compressed frames. Since high-quality compressed frames provide more helpful information than low-quality compressed frames, the network can adjust the input and forget gate weights in QG-ConvLSTM cells. To show the enhancement of the proposed network that uses relative quality information between frames, video frames given to the network as inputs are compressed hierarchically with different qualities using the standard VVC (H.266) codec. In the proposed network, the weights given to the QG-ConvLSTM network are determined by extracting quality-related features from compressed video frames. Additionally, there is no reference frame for the quality feature extraction, so a no-reference image quality assessment method via Transformers, relative ranking, and self-consistency is suggested. The quality enhancement performance of the proposed network is measured with frequently used metrics, namely PSNR, MS-SSIM, and VMAF.

Keywords: video compression, learned video post-processing, video quality enhancement, quality-gated convolutional LSTM, HLVC, no-reference image quality assessment

ÖZETÇE

Yüksek Lisans Tez Başlığı
Hilal Güven
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
September 7, 2023

Son dönemlerde, kodlama verimliliği ve kalite iyileştirme açısından derin öğrenmenin video sıkıştırma ve iyileştirmede uygulanmasında önemli ilerleme kaydedilmiştir. Bu çalışmada, video sıkıştırma ve yeniden inşa etme için, göreceli kalite dikkate alınarak sıkıştırılmış video karelerinin kalitesini artırmak için Kalite Kapılı Evrişimsel Uzun-Kısa Süreli Bellek (QG-ConvLSTM) hücrelerini içeren öğrenilmiş bir işleme sonrası ağ önerilmektedir. Önerilen QG-ConvLSTM hücre tabanlı son işleme ile ağın, komşu sıkıştırılmış kareler arasındaki kareler arası korelasyon ve kalite dalgalanmasından faydalanmakta ve bundan tam avantaj sağlanmaktadır. Yüksek kaliteli sıkıştırılmış çerçeveler, düşük kaliteli sıkıştırılmış çerçevelerden daha yararlı bilgiler sağladığından, önerilen ağ QG-ConvLSTM hücrelerindeki unut ve giriş kapı ağırlıklarını ayarlayabilmektedir. Çerçeveler arasında göreceli kalite bilgisi kullanan önerilen ağın gelişimini göstermek için, girdi olarak ağa verilen video çerçeveleri, standart VVC (H.266) sıkıştırma algoritması kullanılarak farklı kalite değerleriyle hiyerarşik olarak sıkıştırılmaktadır. Önerilen ağda, QG-ConvLSTM ağına verilen ağırlıklar, sıkıştırılmış video karelerinden kaliteyle ilgili özelliklerin çıkarılmasıyla belirlenmektedir. Ek olarak, kalite özelliği çıkarımı için bir referans çerçevesi bulunmadığından Transformer, göreceli sıralama ve öz-tutarlılık algoritmalarını kullanan referanssız bir görüntü kalitesi değerlendirme yöntemi önerilmektedir. Önerilen ağın kalite iyileştirme performansı, PSNR, MS-SSIM ve VMAF gibi, sıklıkla kullanılan metriklerle ölçülür.

Anahtar Kelimeler: video sıkıştırma, öğrenilmiş video son işleme, video kalitesini iyileştirme, kalite kapılı evrişimsel LSTM, HLVC, referanssız görüntü kalitesi değerlendirmesi

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to Professor Ahmet Murat Tekalp for his guidance, continuous support, and understanding during my master's studies. He has always inspired me with his knowledge, expertise, motivation, and passion for learning and teaching.

I acknowledge the scholarship provided by KUIS AI Center in the scope of the AI Fellowship Program and the Technological Research Council of Turkey (TÜBİTAK) in the scope of the 2210/A Domestic General Master's Scholarship Program during my master's studies.

I would like to thank my special friend Batuhan, with whom I spent very precious, joyful, and also difficult times during this period, for being with me all the time, making me laugh, and making me feel understood. I am also deeply thankful to my dear friends Burak, Canberk, Mert, Müge, Sadra, and Vahideh for being in my life and having a very special, beautiful, and fun time together. I would also like to thank my dear friend Doğukan, who always made me feel his support even though he was far away from me.

I will always be deeply grateful to Kerem, who has had a great place in my life during this period, for all the things he has done for me, for being with me all the time, supporting me in all circumstances, for motivating me during difficult times, and for all his efforts and sacrifices.

Last but not least, I would like to express my genuine and special thanks to my beloved family for always being there for me, believing in me, and supporting me unconditionally. I couldn't have made it without them. I am sincerely grateful to my dear mother, Nezihe, for all her efforts, sacrifices, and dedication and to my dear father, Cumali, for worrying about me and supporting me all the time. I am grateful to my dear brother Çağrı, whose endless support, motivation, and help I

feel constantly throughout my life, especially in this period. I am also grateful to my little and sweet niece Almila, to her precious mother Hatice, and to my cat Ceviz for being in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CODEC	Coder-Decoder
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
MS-SSIM	Multiscale Structural Similarity
VMAF	Video Multi-Method Assessment Fusion
GOP	Group-of-Pictures
AVC	Advanced Video Coding
HEVC	High Efficiency Video Coding
VVC	Versatile Video Coding
ReLU	Rectified Linear Unit
LSTM	Long Short-Term Memory
Bi-LSTM	Bi-directional LSTM
ConvLSTM	Convolutional Long Short-Term Memory
QG-ConvLSTM	Quality-Gated ConvLSTM
FPS	Frames Per Second
RGB	Red, Green and Blue
LDV	Large-Scale Diverse Video
ResNet	Residual Network
NR-IQA	No-Reference Image Quality Assessment

Chapter 1

INTRODUCTION

Data compression is the process of encoding in which fewer bits are used to represent the same amount of data. Compression is necessary and essential to efficiently utilize storage space and speed up data transmission. [Yang et al., 2020] claims that nearly 70-80% of mobile data traffic is produced by video transmission. Increasing the effectiveness of image and video codecs is a regularly used solution. This implies that improved video compression techniques can achieve faster data transmission and can be used to send signals faster with fewer bits and store them in less space. For this reason, video compression is a critical research area nowadays.

1.1 Basics of Image and Video Coding

Video compression is the process of encoding a video file to take up less space than the original file and is easier to transmit over the Internet. Video signals comprise sequential image frames in a time series and optional audio in the background. Frames per second (FPS) is a popular statistic used in video capture to measure frame rate, or the number of images presented sequentially per second. Typically, we can represent a video comprising many RGB or YCbCr consecutive frames without audio in four dimensions - time, color channel, height, and width. That is, videos can be expressed as sequential image frames.

Images, elements that make up the video, are made of pixels. Pixels are typically thought of as the smallest individual element of a digital image, and they can be represented by bits. Since videos are the sequencing of pictures in the time domain, many ideas and methods used in image compression algorithms are also used in video compression systems. Image pixel values located in the same area are usually close

to each other due to the transitions and so contain a lot of spatial redundancies. Additionally, videos comprise many similar frames with a few differences, and there are intraframe and interframe correlations between the adjacent frames so that they can also be compressed in the time domain because of the spatial redundancies between the adjacent frames. By eliminating redundant or unnecessary information, the images and videos are compressed efficiently, and the file size is reduced, making it easier to store and transmit.

The advantages of image compression can be summarized as follows;

1. Reduced File Size; One major benefit of image compression is the reduction in file size. Compressed images take up space when stored, allowing for disk space utilization and faster file transfers.

2. Transmission; Smaller file sizes enable transmission of images over networks, making it easier to share photos and graphics. This is particularly important in today's era of media and online platforms.

3. Bandwidth Optimization; Compressed images consume less bandwidth during transmission, leading to cost savings for internet service providers and improving the viewing experience for users.

There are two types of compression scenarios: lossless and lossy compression. In lossless compression, the data quality remains the same when compressed and reconstructed. Lossless compression algorithms aim to minimize the rate without introducing any distortion. The human stereo-color vision is a complex process that cannot be understood completely. The human eye is more sensitive to high-frequency details than low-frequency details. This creates redundancy in some details. Considering this fact, data space can be saved by sacrificing some less essential and useless details. Unlike the lossless compression scenario, in lossy compression, there will be some information loss in data when it is reconstructed to compress more effectively than in the lossless compression scenario. Thus, the decompressed images will be of lower quality than the original images but will take up less space. In compression,

images compressed to take up a smaller space will be of lower quality. So, there is a trade-off between the quality of images and the file size. This trade-off is called rate-distortion trade-off or rate-distortion theory.

The rate-distortion tradeoff is a fundamental concept in information theory and signal processing. The concept refers to the trade-off between the amount of information that can be transferred or stored (rate) and the quality of the reconstructed signal (distortion). Claude Shannon initially introduced the rate-distortion theory, and it quantifies the relationship between the amount of data that can be transmitted and the level of fidelity in the reconstructed signal [Shannon, 1993]. Rate-distortion tradeoff plays a key role in data compression techniques. Lossy compression algorithms, such as JPEG and JPEG2000, exploit this tradeoff by omitting specific details to reduce the rate while maintaining an acceptable level of perceptual quality. End-to-end rate-distortion optimization of learned image compression is formulated as the following loss equation given by

$$Loss = R + \lambda D = E_{x\sim p_x}[-log_2 p_y^{\wedge}(\lfloor f(x) \rceil)] + \lambda E_{x\sim p_x}[d(x, g(\lfloor f(x) \rceil))] \tag{1.1}$$

where λ is the Lagrange multiplier that determines the desired rate–distortion tradeoff, p_x is the unknown distribution of images, $\lfloor . \rceil$ represents rounding to the nearest integer which is quantization, $y = f(x)$ is the encoder, $\widehat{y} = \lfloor y \rceil$ are the quantized latents, $p_{\widehat{y}}$ is a discrete entropy model, and $\widehat{x} = g(\widehat{y})$ is the decoder with $\widehat{x}$ representing the reconstructed image [Minnen et al., 2018]. Using the loss calculation in Equation 1.1, the sum of rate loss and distortion loss between the original and reconstructed images can be minimized for a fixed tradeoff determined by choice of λ. In this optimization problem, the λ parameter defines the tradeoff between the rate and the distortion. If λ is given a significant value, then this means the effect of the distortion between the compressed and original frames on the overall loss function will be higher. Thus, higher-quality frames with higher rates are obtained. However, if λ is given a small value, the weight of the rate on the loss function will be higher, and better-compressed, lower-quality frames with lower rates are obtained. Hence, λ is a parameter that is determined by the performance request from the model. The

generalized divisive normalization (GDN) layer in Equation 1.2 is used for activation in analysis and synthesis transforms g_a and g_s shown in Figure 2.1. GDN is a parametric nonlinear transformation that is well-suited for making Gaussian-distributed data from images. In the first instance, it was introduced to model nonlinear properties of cortex neurons where linear responses were divided by pooled responses from their rectified neighbors [Ballé et al., 2016].

$$y_i = \frac{x_i}{\sqrt{\beta^2 + \Sigma_j \gamma_j x_j^2}} \tag{1.2}$$

Since the data is discrete after the quantization, it can be losslessly compressed using entropy coding methods. Arithmetic encoder (AE) and arithmetic decoder (AD) are parts of the entropy coding. Entropy coding relies on a prior probability model of the quantized representation, and this is known to both encoder and decoder [Ballé et al., 2018]. In the scale hyperprior model, the latent space (the left side of Figure 2.1) is modeled with Gaussian distribution, and the standard deviation and mean of this distribution are predicted by the hyperprior model (the right side of Figure 2.1) [Ballé et al., 2018]. The hyperprior model can increase the compression performance of the model since it helps to reduce rate loss at latent space by providing parameters of the Gaussian distributed latent space. The amount of side information that is sent is smaller than the reduction of the rate of latent space [Ballé et al., 2018].

1.2 Picture Types in Video Compression

Different types of picture frames are used in video compression to store and transmit video data effectively. These different frames are called I, P, and B frames. These three different types of frames are used for specific purposes.

The first type is I frames, or intra-coded frames, which are also referred to as keyframes. They are complete, independent frames that do not rely on other frames. The decoder uses the I frame as a starting point to begin decoding the video stream. They contain the most detailed information, so I frames have larger sizes than other

frames.

The second type is P frames, or predicted frames, which are compressed frames that rely on previous I or P frames for decoding. They keep track of the variations between the reference frame and the current frame. By using motion estimation algorithms, P frames predict the motion of objects in the video and only encode the changes that occur. For this reason, P frames are considerably smaller than I frames in size. P frames make video compression efficient by exploiting spatial and temporal redundancy in the video sequence.

The third type is B frames, or bidirectional predicted frames, which are video compression's most complex frame types. They use both previous and next frames for decoding. By comparing the changes between the previous and following I or P frames, B frames can be used to forecast the motion of objects in video. Since both the previous and the subsequent frames are used bidirectionally, B frames can be compressed with considerably greater compression ratios than I and even P frames. However, B frames demand higher computing time and power for both encoding and decoding because of the complex motion estimation process.

1.3 Problem Statement

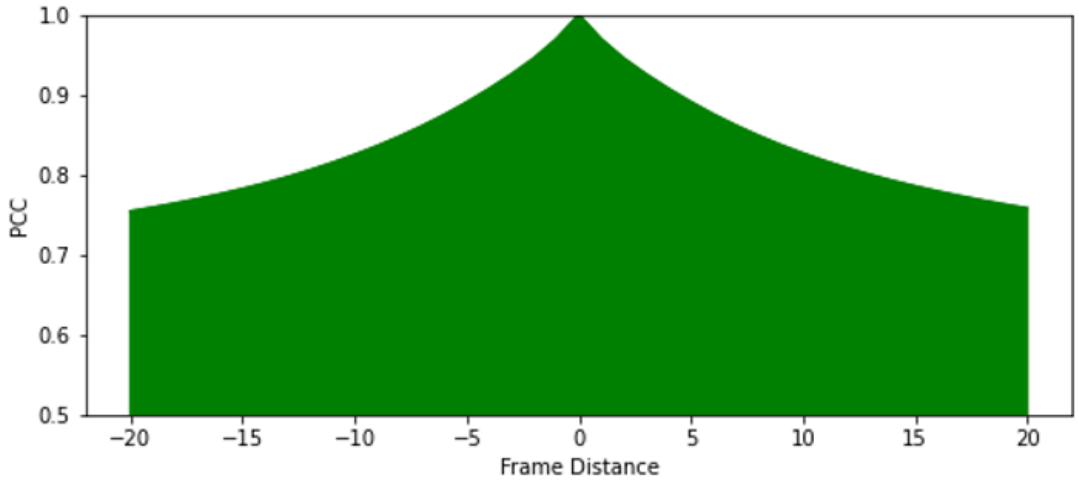


Figure 1.1: Average PCC between frames with various distances.

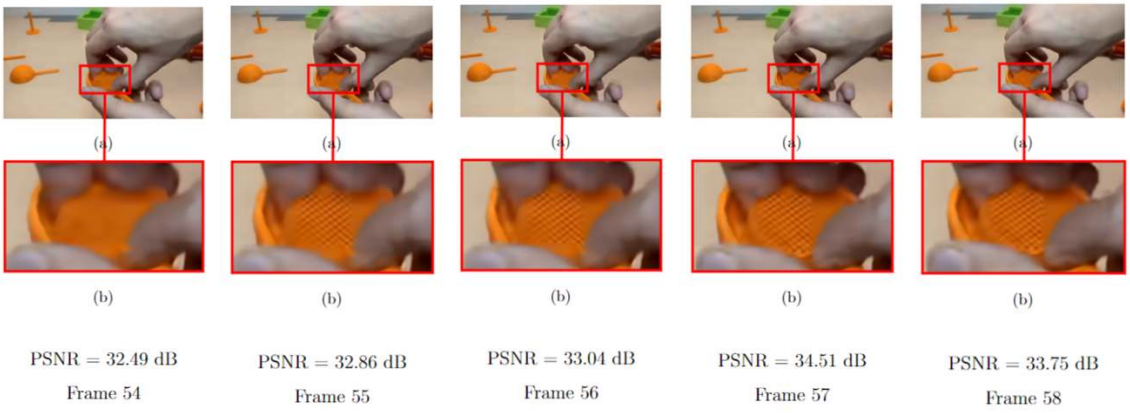


Figure 1.2: Example of the frame-level PSNR and subjective quality for the VVC compressed video sequence in LDV 2.0 dataset

The proposed post-processing network for video compression tries to bring a solution to the following two problems:

- First, inter-frame correlation is essential for the compression efficiency of video. It creates some redundancy. Correlation between the neighboring frames is high, so reconstruction quality can be increased by using this fact. The correlation between two frames can be evaluated using Pearson Correlation Coefficient (PCC) [Yang et al., 2019]. In Figure 1.1, PCC is calculated between each frame and its 40 neighboring (20 previous and 20 subsequent) frames. These verify that the frames in a large range have a strong correlation in contents, and such correlation degrades along frame interval.

- Second, the visual quality of frames remarkably fluctuates after compression [Yang et al., 2018a]. In Figure 1.2, it can be seen that the quality fluctuates between compressed frames in terms of PSNR. Visually, it can also be seen that high-frequency details and artifacts vary from frame to frame; for example, there is some loss in detail in Frame 54 compared to other frames. When the visual quality of the frames is considered, the efficiency of reconstruction can be increased by giving a more significant role to the compressed frames with higher quality and higher rate and a less significant role to the compressed

frames with lower quality and lower rate in the reconstruction process. Thus the distortion between uncompressed frames and reconstructed frames can be decreased at some point. The visual quality of compressed frames can be evaluated using different metrics such as PSNR, MS-SSIM, VMAF, LPIPS, etc.

1.4 Thesis Outline and Contribution

The outline of the thesis is as follows:

Useful information about image and video coding and compression, the importance and motivation of video compression, and the problem statement are given in Chapter 1.

In Chapter 2, first, the standard video codecs such as H.264, H.265, and H.266 are covered. Then, the background, related works, and literature reviews about learned image & video compression, learned quality enhancement for compression, and no-reference quality extraction methods that are used to obtain quality features from images and videos are discussed to be able to give an intuition and background about the topic and task.

Chapter 3 introduces the proposed network for quality enhancement post-processing network. In this chapter, the small structures and networks, such as QG-convLSTM, residuals, and no-reference feature extraction, are covered. By providing figures about the networks with the presentation of the concepts, the impression and effect of the system structure have been tried to be explained.

In Chapter 4, information about datasets prepared and used, training details, evaluation metrics, such as MSE, PSNR, MS-SSIM, VMAF, and experimental results are discussed.

Lastly, in Chapter 5, the conclusion and future direction for this research topic and what can be done to improve this study are discussed.

In this work, the main contributions are:

- In the proposed post-processing network, Quality-Gated Convolutional LSTM cells are used to exploit inter-frame correlations between the video frames.

- Quality features that are fed into the network are obtained using a pre-trained no-reference quality assessment approach that makes use of CNNs and a self-attention mechanism in Transformers to extract both local and non-local quality features.

Chapter 2

BACKGROUND & RELATED WORK

Compression is achieved by applying various algorithms that exploit the natural redundancy in images/videos.

2.1 State-Of-The-Art Video Compression Standards

2.1.1 Advanced Video Coding (AVC / H.264)

AVC (Advanced Video Coding), generally known as h.264, is one of the video compression standards that can provide high-quality video compression at manageable file sizes and has been designed by the Joint Video Team (JVT). The efficiency of h.264 codec in video compression tasks is one of its main benefits. While maintaining video quality, it significantly reduces the space that a file takes. Different methods are used by the h.264 codec to accomplish effective compression. It employs macroblock-based motion compensation, which estimates the movements of objects in a video frame and thus exploits temporal redundancy. It uses variable block-size motion compensation that enables more precise motion prediction of objects. Additionally, variable-length coding is employed to lower the size of a video file. In order to increase compression efficiency, it also uses entropy coding methods like Context-Adaptive Variable-Length Coding (CAVLC) and Context-Adaptive Binary Arithmetic Coding (CABAC). H.264 also supports different video formats, video resolutions, and frame rates [Wiegand et al., 2003].

2.1.2 High Efficiency Video Coding (HEVC / H.265)

High-Efficiency Video Coding (HEVC) [Sullivan et al., 2012] is a video compression standard that performs better than prior ones like H.264/AVC. An HEVC encoder produces a compressed video bitstream by compressing input video com-

posed of sequential video frames. The compressed bitstream is transmitted and then decompressed by the decoder to produce decoded frames sequentially. The fundamental structure of HEVC is the same as that of earlier standards like MPEG-2 and H.264/AVC. However, HEVC includes numerous growing improvements: Greater partitioning flexibility, including a range of partition sizes; enhanced flexibility of prediction modes and change block sizes; more advanced deblocking and interpolation filters; more advanced motion vector prediction; and characteristics that facilitate effective parallel processing.

2.1.3 Versatile Video Coding (VVC / H.266)

Versatile Video Coding (VVC), also known as H.266, is one of the state-of-the-art video compression standards developed by the Joint Video Experts Team (JVET), which is a merger of the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG) [Segall et al., 2018]. By offering higher compression performance and better video quality than earlier video compression standards like H.264, High-Efficiency Video Coding (HEVC/H.265), VVC has taken its place among the state-of-the-art video compression standards. The goal of VVC/H.266 is to increase compression performance significantly while maintaining acceptable video quality. Larger block sizes, improved motion prediction, and better intra-prediction modes are only a few of the sophisticated coding approaches used. By using these methods, VVC is able to encode video content more effectively than competitors, resulting in reduced file sizes for a given level of video quality. The need for high-quality video streaming and video communication services was one of the primary motivations behind the creation of VVC. To enable smooth distribution and playback of material over multiple networks and devices, effective video compression is essential as video resolutions and dynamic range continue to increase. VVC gives compression efficiency benefits over HEVC, which in turn made significant advances over Advanced Video Coding (AVC/H.264), its predecessor. It is important to keep in mind that these improvements frequently come at the expense of greater computing complexity, which might affect the durations for encoding and

decoding.

2.2 Learned Image Compression

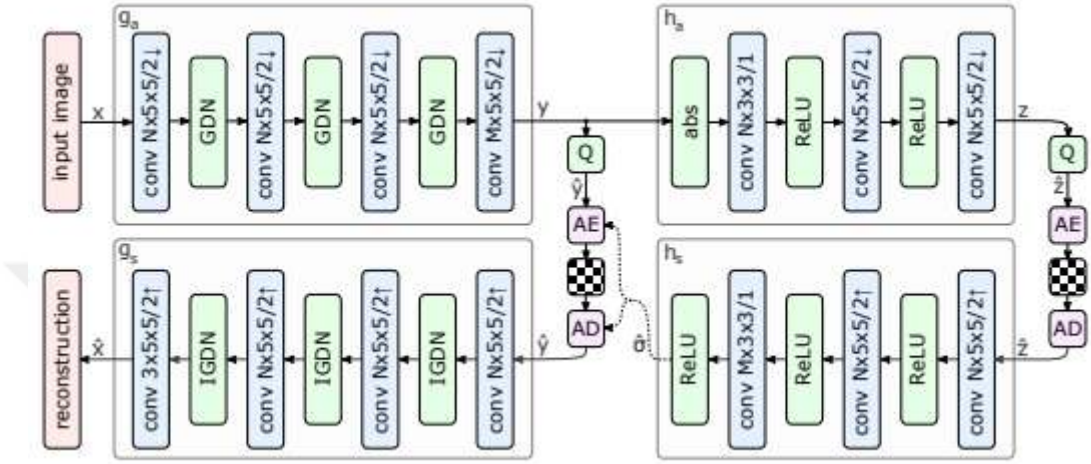


Figure 2.1: The scale hyperprior architecture in [Ballé et al., 2018]. The image autoencoder network is placed on the left side. On the right side, the autoencoder network that employs the proposed scale hyperprior is placed. Here, Q denotes the quantization process, and AE and AD denote the arithmetic encoder and arithmetic decoder, respectively. Parameters of the convolutional layers are shown as the number of filters x height of kernel x width of kernel where ↑ represents upsampling, and ↓ represents downsampling processes [Ballé et al., 2018].

2.2.1 Variational Image Compression With A Scale Hyperprior

Image-adaptive entropy modeling is presented in [Ballé et al., 2018] with a scale hyperprior technique in which a hyperprior network learns the distribution of the image's latent variables, y. With the scale hyperprior technique, it is possible to compress images efficiently by modeling the latent variable y of the image with independent zero-mean Gaussian random variables conditioned on scale hyperprior latent z, which are provided to the decoder as side information. The overall rate-distortion performance of the codec is enhanced with this network because the side

information size is significantly smaller than the latent representation size of the image.

2.3 Learned Video Compression and Quality Enhancement Post-Processing

2.3.1 End-to-End Rate-Distortion Optimized Learned Hierarchical BiDirectional Video Compression

In an end-to-end hierarchical bidirectional compression scenario, I, P, and B frames are compressed hierarchically, and the compression ratio of these frames varies, considering the hierarchical level and the GOP size. In Figure 2.2, the hierarchical levels can be seen. In that network, there are 3 hierarchical levels and 8 frames in a GOP. First, I frames x_0 and x_8 are compressed with an image compression algorithm. Compression rates of the I frames are generally lower compared to the other frames in the GOP because any reference frame is used during the compression. In the first hierarchical level, B frame x_4 is compressed bidirectionally using x_0 and x_8 as references. Since high-quality reference frames are used while compressing, the compression rate of the frame x_4 is higher than the frames x_0 and x_8. In the second hierarchical level, B frame x_2 is compressed bidirectionally using x_0 and x_4 as references, and B frame x_6 is compressed bidirectionally using x_4 and x_8 as references. In the third and last hierarchical level, frames x_1, x_3, x_5, and x_7 are compressed bidirectionally using the other frames of the GOP as references. The same process is repeated for every GOP in a video sequence.

The bidirectional compression network is composed of motion estimation, motion compensation, motion field compression, and residual compression networks [Yılmaz and Tekalp, 2021]. Motion subsampling and temporal motion prediction are included in the motion field compression module to boost the effectiveness of motion compression. It is comparable to delivering one motion vector per 4x4 pixel block when motion field subsampling is used. In order to reduce the dynamic range of the vectors that must be compressed, temporal motion prediction requires the constant velocity motion assumption. Even if occlusion effects are not used, some motion compensation artifacts at motion boundaries are unavoidable. To reduce

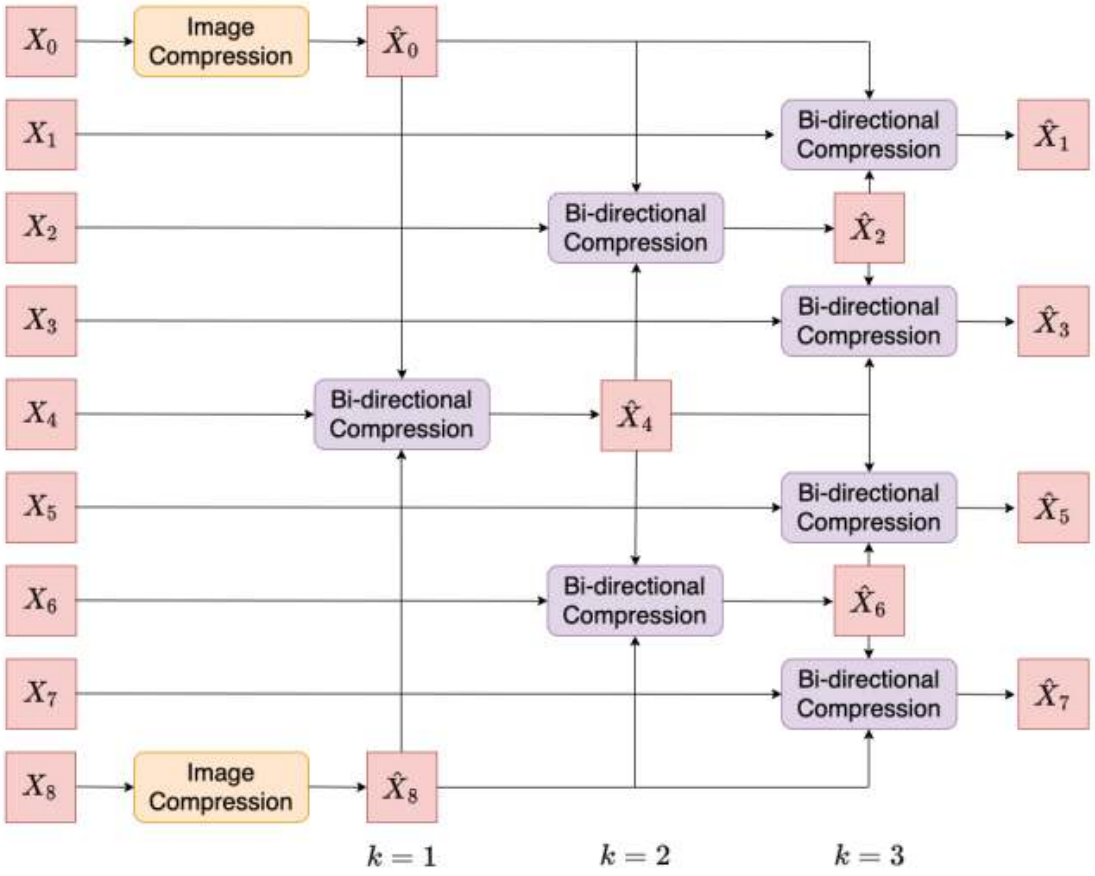


Figure 2.2: 3-level hierarchical bidirectional video compression scheme in [Yılmaz and Tekalp, 2021].

such motion distortions, the use of the learned bidirectional motion compensation network, which is tuned to learn the optimal continuous coefficients to combine the forward and backward warped frames, is suggested in [Yılmaz and Tekalp, 2021].

One of the main contributions in [Yılmaz and Tekalp, 2021] to the literature, an end-to-end optimized hierarchical bidirectional video coding framework is proposed. According to the results, the proposed framework outperforms the SVT-HEVC encoder in terms of both PSNR and MS-SSIM metrics, as well as the HM 16.23 reference codec in MS-SSIM. To make decoder operation simpler, a single bidirectional motion-compensated compression network that functions for all levels of the hierarchy is proposed. Additionally, ablation examinations are conducted to demonstrate how each of the new approaches they suggest, such as motion field subsampling,

temporal bidirectional motion vector prediction, and learned bidirectional motion compensation mask, contributes to the higher performance of the proposed codec.

2.3.2 Quality-Gated Convolutional LSTM for Enhancing Compressed Video

Videos are made sequentially images in a time scale. For this reason, there are usually minor changes or movements between 2 frames. If we use the previous and the next frames while applying compression to a single frame, since there is a small motion, we can efficiently compress the image with a high compression ratio and smaller size. Samely, while enhancing the video quality, using the information in the previous and the next frames and the motion change between these frames, quality distortion can be decreased. For this reason, the bidirectional ConvLSTM architecture provides an enhancement network to exploit the inter-frame correlation between the sequential frames in a timely manner and thus improve the quality.

By combining a bidirectional ConvLSTM cell, which is able to take advantage of the inter-frame correlation between the adjacent frames with quality-related gates to determine input and forget gates of the ConvLSTM cell, the QG-ConvLSTM technique completely uses the inter-frame correlation in the video between the sequential frames bidirectionally, giving it an edge over other approaches for improving compressed video quality. The weights of input and forget gates of the ConvLSTM cells are determined by the quality of compressed images. For example, when the quality of the previous frame is higher, then the forget gate weight of the current ConvLSTM cell will be lower, and this increases the effect of the previous frame in the reconstruction process. Samely, when the quality of the current frame is higher, the input gate weight will be higher, and the effect of the current frame in the reconstruction process will increase. By adjusting the weight considering the quality of frames in the GOP, the reconstruction process can be enhanced. The network may then assign various priorities to frames compressed with different quality levels and learn the trade-off between inter-frame correlation and compressed quality. As a result, the QG-ConvLSTM method improves compressed video quality to a state-of-the-art level.

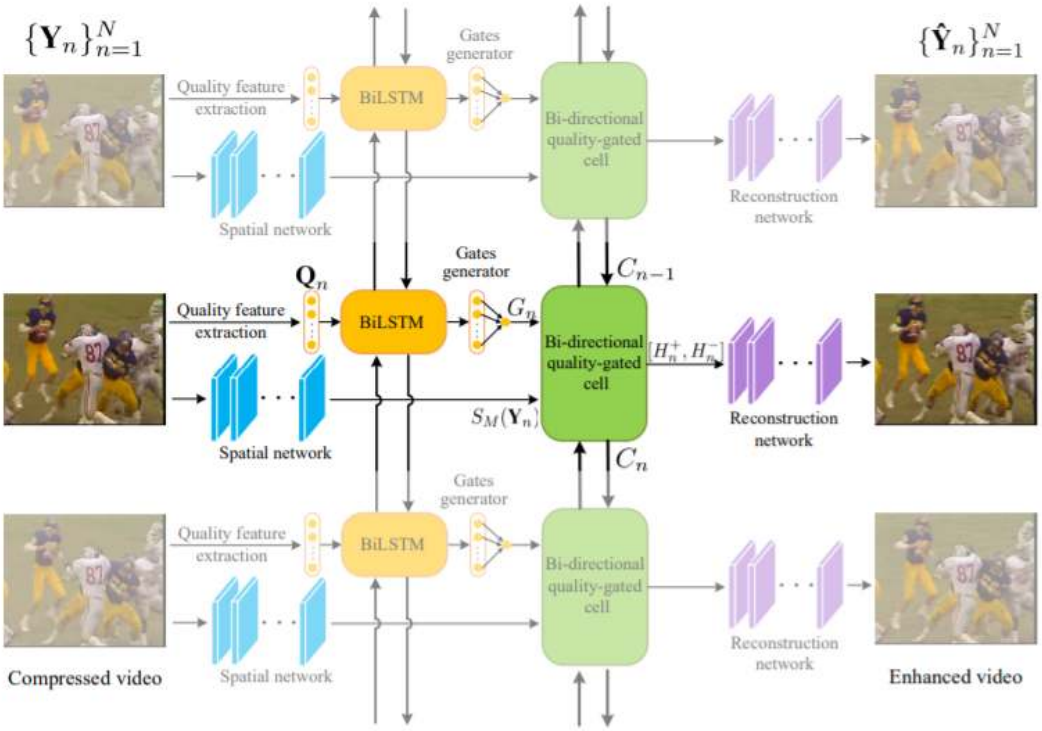


Figure 2.3: The scheme of QG-ConvLSTM based post-processing network presented in [Yang et al., 2019].

The whole network can be seen in Figure 2.3. To obtain the quality features from the images without a reference, the no-reference feature extraction method in [Mittal et al., 2012] is used. By applying this method, 36 spatial features are extracted. Additionally, the quantization parameter (QP) and the bit rate values that can be acquired from the video codec decoder are also used as features. Thus, for a single frame, 38-dimensional quality features are given to the network. Instead of considering a current frame's quality value, the use of relative quality compared to the other frames provides a better enhancement model. In [Yang et al., 2019], for the $n-th$ frame, quality features of current and T surrounding frames are given to the gates generator. Therefore, the feature input for the $n-th$ frame is:

$$Q_n = \{q_{n-T/2}, ...q_n....q_{n+T/2}\} \tag{2.1}$$

The QG-ConvLSTM technique assigns different priorities to frames with different

quality levels by integrating quality-related gates in the ConvLSTM cell. An LSTM network that replaces the "forget" and "input" gates in the original ConvLSTM cell learns the weights of these gates based on quality-related parameters. Due to this, the compressed quality triggers the weights to forget the prior memory and update the current information. As an example, high-quality frames are expected to update their high-quality information to memory while forgetting their lower-quality information, thereby assisting in the improvement of other frames.

The results of experiments demonstrate that the QG-ConvLSTM technique improves compressed video quality to state-of-the-art levels. According to results of [Yang et al., 2019], for QP = 37, the QG-ConvLSTM technique has a PSNR of 0.5871 dB, which is greater than the PSNRs of the second and third-best approaches, MF-CNN (0.5102 dB) and QE-CNN (0.3738 dB), respectively. The QG-ConvLSTM's parameter count is around 36% of the MF-CNN and QE-CNN. As a result, the QG-ConvLSTM technique exceeds all other methods while improving the quality of compressed video with fewer parameters than the most recent quality enhancement approaches.

2.3.3 Learning for Video Compression with Hierarchical Quality and Recurrent Enhancement

Hierarchical learned video compression (HLVC), which is an end-to-end learned bidirectional video compression method, is proposed in [Yang et al., 2020], and the overall architecture can be seen in Figure 2.4. Frames in a GOP are categorized into three levels by HLVC. Keyframes that are individually compressed with the best quality belong to Layer 1. The frame in Layer 2, which is the middle frame of the GOP, is compressed by the Bidirectional Deep Compression (BDDC) network, which receives two Layer 1 frames as input reference frames. Frames in Layer 2 are compressed with higher quality compared to other layers. In Layer 3, frames are compressed with a lower quality compared to the others, Layer 1 and Layer 2 frames, using a Single Motion Deep Compression (SMDC) network. As shown in Figure 2.6, SMDC uses motion vectors produced from a single motion map to compress several

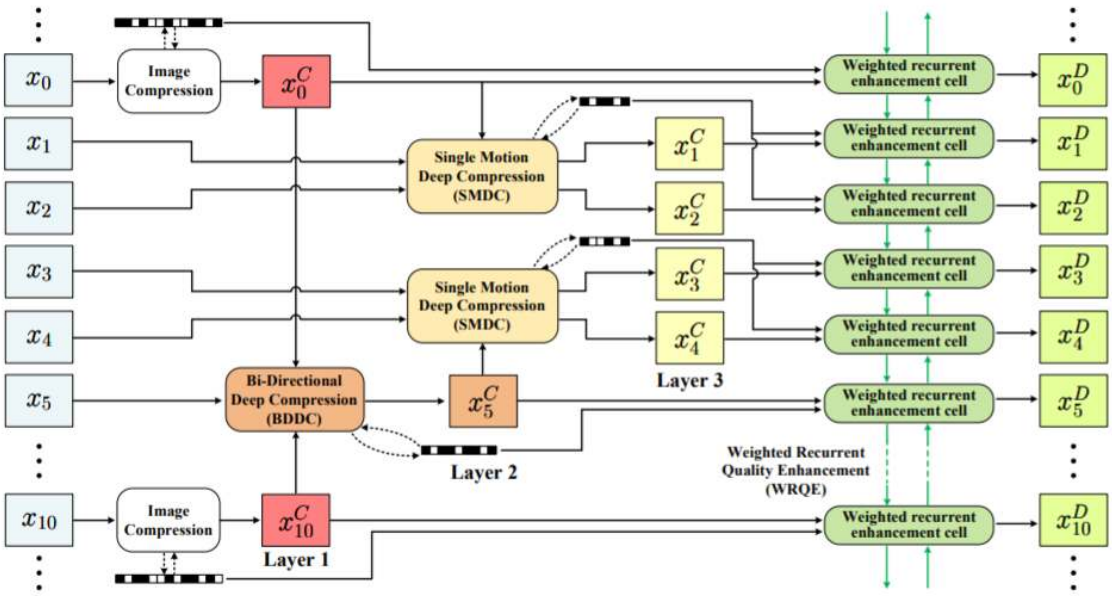


Figure 2.4: The diagram of the HLVC algorithm proposed in [Yang et al., 2020].

frames. After all, a weighted recurrent quality enhancement network (WRQE) is suggested as a post-processing network to improve the quality of compressed frames.

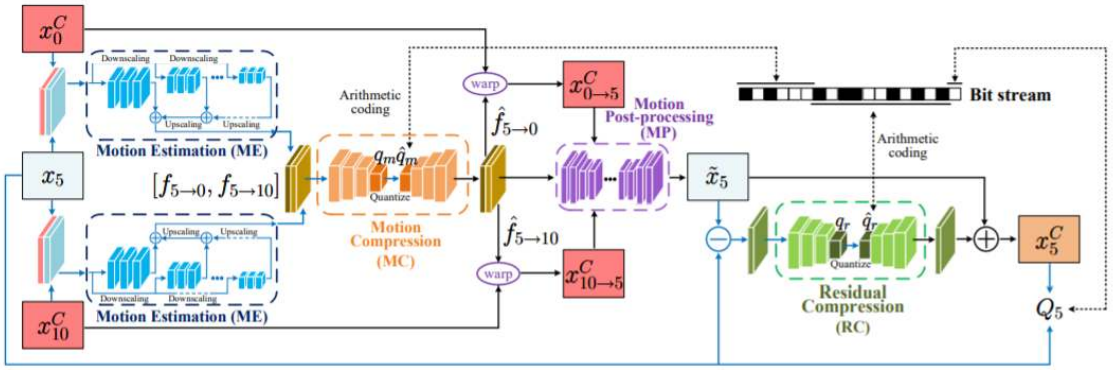


Figure 2.5: The scheme of Bidirectional Deep Compression (BDDC) method proposed in [Yang et al., 2020].

In the HLVC technique, the Bidirectional Deep Compression (BDDC) network is suggested to compress the frames in layer 2. The BDDC network employs the compressed frames in layer 1 as bi-directional references to compress the frames in

layer 2, medium quality layer. BDDC also contributes useful information to compress and improve the frames in layer 3, poor quality frames. The BDDC network uses preceding and subsequent frames from layer 1 and uses them as bidirectional references, as in Figure 2.5.

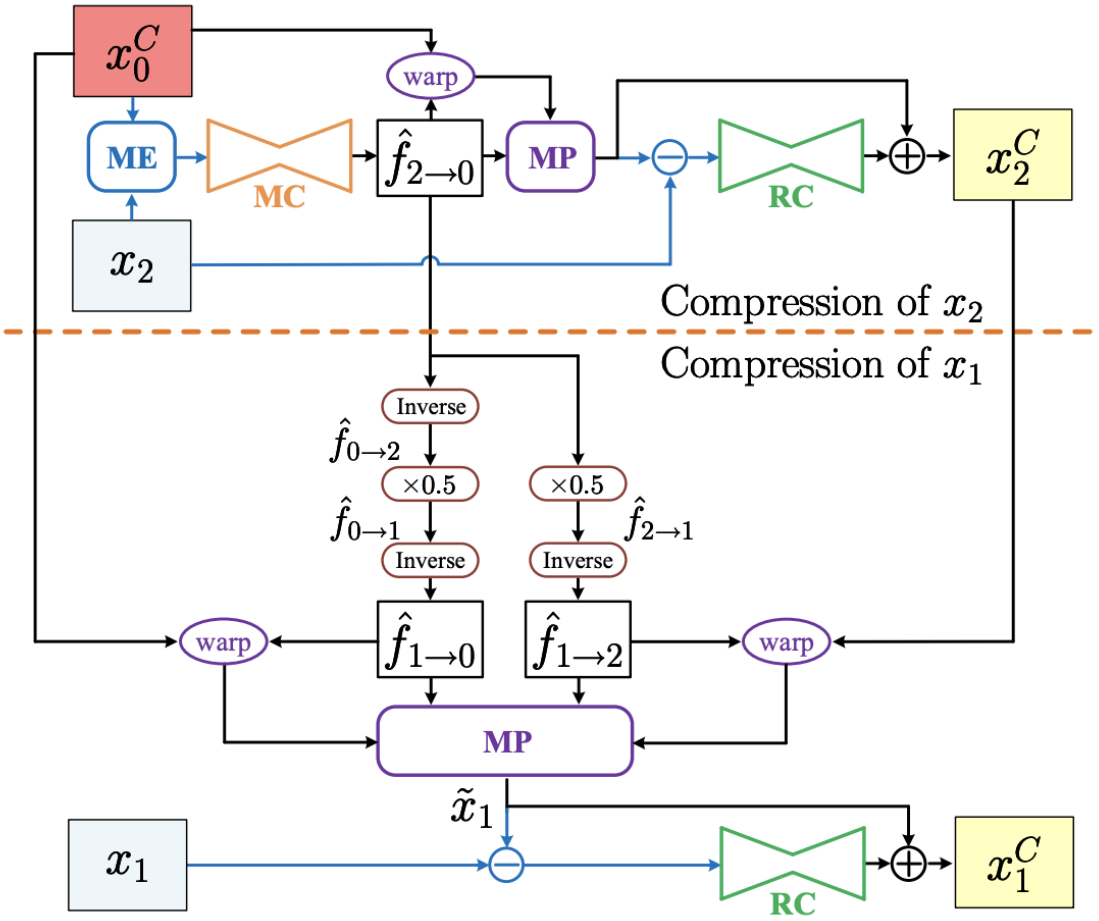


Figure 2.6: The scheme of Single Motion Deep Compression (SMDC) method proposed in [Yang et al., 2020].

In the HLVC technique, the Single Motion Deep Compression (SMDC) network is suggested to compress the frames in layer 3. The bitrate can be decreased since the SMDC network uses a single motion map to represent the motions between multiple frames. Using the single closest compressed frames from layers 1 and 2 as references, the SMDC network compresses two frames into a single motion map. It is shown in Figure 2.6 how the frames x_1 and x_2 are compressed using a single

motion map using ${x_0}^C$ as a reference, whereas x_3 and x_4 utilize ${x_5}^C$ as the reference.

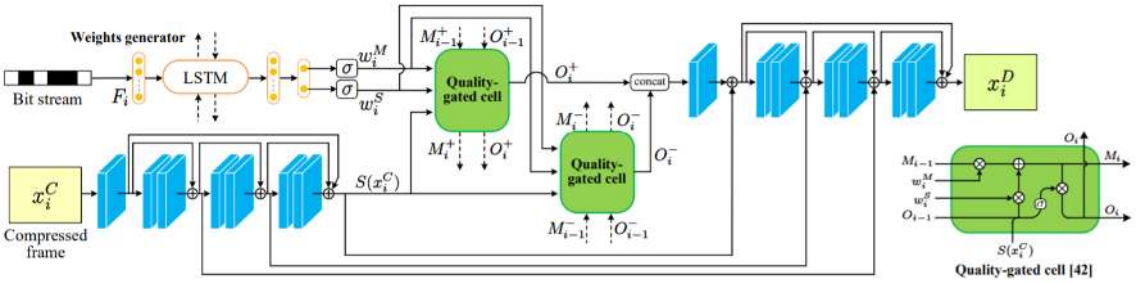


Figure 2.7: The scheme of Weighted Recurrent Quality Enhancement (WRQE) method proposed in [Yang et al., 2020].

The Weighted Recurrent Quality Enhancement (WRQE) network, which employs a quality-gated cell based on [Yang et al., 2019], uses quality attributes to apply multi-frame information for recurrent enhancement fairly and takes advantage of multi-frame correlations. Figure 2.7 illustrates the WRQE network's design. The WRQE network enriches the compressed frames by employing multi-frame information as an input along with quality and bitrate data. The results demonstrate that the HLVC methodology surpasses H.265's(HEVC) "Low-Delay P (LDP) very fast" mode and reaches state-of-the-art performance in learning video compression approaches.

The WRQE network is basically designed by the QG-ConvLSTM network from [Yang et al., 2019] with some substantial changes. Differently, residual blocks and skip connections are used in the spatial feature extraction and reconstruction networks to enhance the architecture.

More crucially, the effect of a frame on the other frames in terms of enhancing the network relies on how higher or lower quality it is compared to the others. The decoder is unable to acquire each frame's precise quality. The compression quality, on the other hand, is encoded in the bit stream, so the compression quality and bitrate can be obtained from the bit stream. Therefore in the network (Figure 2.7), compression quality and bit-rate from the decoder are used as quality features to feed the network. The quality features are processed by the weights generator network

and then are fed into the QG-ConvLSTM cell to determine forget and input gate weights along with the spatial characteristics coming from the spatial network [Yang et al., 2020].

2.4 Other Learned Post-Processing Networks for Quality Enhancement of Video

The earliest works that utilize deep learning solutions mainly employ CNNs for quality enhancement. [Yang et al., 2017] Proposes Decoder-side Scalable (DS) CNN. CNN model learning is used in the DS-CNN approach to decrease HEVC frame distortion in both I and B/P frames. It differs from previously suggested CNN-based quality enhancement methods, which exclusively address intra-coding distortion and are inapplicable to B/P frames. Furthermore, the proposed approach is independent of any type of encoder architecture and thus offers better modularity in terms of encoding.

One of the early works that is closely related to our proposed work is [Yang et al., 2018b]. Authors argue that most methods mostly ignore the similarities between succeeding frames and instead concentrate on improving the quality of a single frame. Furthermore, [Yang et al., 2018b] shows that there is significant quality variation between compressed video frames and that low-quality frames may be improved by employing the nearby high-quality frames. To this end, the authors propose a novel approach, "Multi-Frame Quality Enhancement" (MFQE), to exploit the high-quality frames. To find Peak Quality Frames (PQFs) in compressed video, they first create a Support Vector Machine (SVM) based detector. The non-PQF and its nearest two PQFs are used as the input of a unique Multi-Frame Convolutional Neural Network (MF-CNN), which is aimed at improving the quality of compressed video. Through the Motion Compensation subnet (MC-subnet), the MF-CNN adjusts motion between the non-PQF and PQFs. The Quality Enhancement subnet (QE subnet) then works with the nearby PQFs to lessen the compression artifacts of the non-PQF.

[Guan et al., 2019] improves up on the MFQE [Yang et al., 2018b] by proposing

the MFQE 2.0. The proposed approach utilizes a Bidirectional Long Short-Term Memory (BiLSTM) based detector to identify the presence of PQFs in a compressed video stream and a Multi-Frame Convolutional Neural Network (MF-CNN) to mitigate motion between non-PQF and PQFs. The main contributor is the Bidirectional Long Short-Term Memory (BiLSTM) based detectors which are employed to find Peak Quality Frames (PQFs) in compressed video. The detector is intended to extract PQF spatial and temporal information, and it has a good F1 score, recall, and precision for PQF detection. The frames that are best suited for quality improvement are found using the detector as a pre-processing step. The Multi-Frame Convolutional Neural Network (MF-CNN) is used to improve quality after receiving the output from the detector. Overall, the proposed BiLSTM-based PQF detector outperforms the SVM-based method utilized in the original (MFQE 1.0 [Yang et al., 2018b]).

To complete a framework, the authors propose a Motion Compensation subnet (MC-subnet) used by the Multi-Frame Convolutional Neural Network (MF-CNN) to account for motion between non-PQF and PQFs. The non-PQF and the two PQFs that are closest to it are fed into the MC-subnet, which creates a motion-compensated non-PQF by estimating the motion between them. A block-matching algorithm is used to perform the motion estimation, and the motion vectors are then used to warp the PQFs to the non-PQF frame. The motion-compensated non-PQF is created by averaging the warped PQFs. The Quality Enhancement subnet (QE-subnet) is then fed the motion-compensated non-PQF for additional processing.

Generative approaches are also widely employed for quality enhancement objectives. Recently generative adversarial networks (GaNs) [Goodfellow et al., 2014] are capable of generating higher quality frames. One of the earliest methods proposed by the is called multi-level wavelet-based generative adversarial network (MW-GAN) [Wang et al., 2020]. Authors argue that the currently used techniques primarily aim to enhance the objective quality of compressed images and videos while ignoring the perceptual quality. While using a GaN architecture. The proposed MW-GAN approach focuses on improving the perceptual quality of the compressed video. For recovering high-frequency details, the MW-GAN approach embeds wavelet packet

transform (WPT) in both the generator and the discriminator, which is closely related to the perceptual level. The main argument for using wavelet packet transform (WPT) in the MW-GAN approach is that WPT can turn high-frequency details that are typically lost due to compression into high-frequency sub-bands. The MW-GAN approach uses WPT at various levels to recover high-frequency details. WPT is also taken into account in the MW-GAN discriminator, which encourages the generator even more to recover high-frequency details. Motion compensation and wavelet reconstruction make up the two main parts of the MW-GAN generator. Motion compensation is first developed to compensate the motion between the target frame and its neighbors with a pyramid architecture inspired by the [Guan et al., 2019]. The target frame's sub-bands are then rebuilt using a wavelet reconstruction network made up of a number of wavelet-dense residual blocks (WDRB). Ultimately, authors empirically show that MW-GAN is more capable than DS-CNN [Yang et al., 2017], MFQE [Yang et al., 2018b] and MFQE 2.0 [Guan et al., 2019] in terms of the perceptual quality. Authors later improve upon the work in [Wang et al., 2021] where they propose MW-GAN+. New architecture employs a multi-level wavelet pixel-adaptive (MWP) module for MW-GAN+, and a 3D discriminator is also considered for temporal coherence. Furthermore, the performance of MW-GAN+ is significantly improved compared to the MW-GAN.

With the increasing popularity of transformers and attention mechanisms, more recent works have started to incorporate attention mechanisms into quality enhancement. A Patch-Wise Spatial-Temporal Quality Enhancement (PSTQE) [Ding et al., 2021] network is suggested as a way to improve compressed videos' quality by fusing spatial and temporal data. Three subnets make up the network: one for spatial feature extraction, one for temporal feature extraction, and one for spatial-temporal feature fusion. In contrast to the temporal feature extraction subnet, the spatial feature extraction subnet extracts spatial features. Then, these features are combined by the spatial-temporal feature fusion subnet to enhance each compressed patch in an adaptive manner. The network does this by recalibrating and fusing spatial and temporal features using a channel and spatial-wise attention fusion block. To recalibrate the importance of each channel and each spatial location, the block em-

ploys a channel-wise attention mechanism and a spatial-wise attention mechanism, respectively.

2.5 No-reference image quality assessment

No-reference image quality assessment (NR-IQA) is an important field in image processing where the goal is to automatically and objectively evaluate the perceived quality of an image without relying on a known high-quality reference image.

One of the early prominent works is [Mittal et al., 2012], where authors propose a no-reference image quality assessment model called BRISQUE, which uses natural scene statistics to quantify possible losses of "naturalness" in an image due to the presence of distortions. In contrast to other NR-IQA methods, BRISQUE does not compute distortion-specific features like ringing, blur, or blocking. Instead, it quantifies potential losses of "naturalness" in the image due to the existence of distortions using scene statistics of spatially normalized luminance coefficients, providing a comprehensive evaluation of quality. Under a spatial natural scene statistic model, the underlying features are derived from the empirical distribution of locally normalized luminances and their products. It differs from previous NR IQA methods [Ferzli and Karam, 2009, Varadarajan and Karam, 2008, Sadaka et al., 2008, Narvekar and Karam, 2009] in that it does not require transformation to another coordinate frame (DCT, wavelet, etc.).

With the recent advent of the Visual transformers [Dosovitskiy et al., 2021] and local attention mechanisms [Liu et al., 2021, Liu et al., 2022], various learning quality assessment methods started employing those strategies to extract more meaningful features [Golestaneh et al., 2022, Yang et al., 2022b, Wang et al., 2022, Zhang et al., 2023]. [Ke et al., 2021] propose MUSIQ, MUSIQ processes native resolution images of various sizes and aspect ratios to get around the fixed shape limitation in batch training for CNN-based IQA models. By doing so, MUSIQ is able to prevent the necessity for resizing and cropping, which can lower the quality of the image. Instead, MUSIQ aligns with the human visual system by extracting multi-scale elements from the full-size image. It uses a self-attention mechanism to collect information across

scales, both local and global. A patch encoding module and a hash-based 2D spatial embedding are used in the model to process a multi-scale image representation, which includes the native resolution image and its aspect-ratio-preserving resized variations. The fixed shape constraint in batch training for CNN-based IQA models can be avoided by MUSIQ thanks to the multi-scale representation.

[Yang et al., 2022b] propose Multi-dimension Attention Network for No-Reference Image Quality Assessment (MANIQA). The proposed method uses a combination of Transposed Attention Blocks (TAB), Scale Swin Transformer Blocks (SSTB), and a dual branch structure for patch-weighted quality prediction to comprehensively use information from spatial and channel dimensions. Authors argue that the TAB as the key element for the overall pipeline. Instead of using the spatial dimension to compute cross-covariance across channels, TAB uses self-attention across channels to create an attention map that implicitly encodes the global context. This enables TAB to gather detailed information from various routes that are overlooked by traditional self-attention.

A similar approach to the [Yang et al., 2022b] is proposed in [Wang et al., 2022]. The authors propose a new approach based on the Swin Transformer [Liu et al., 2021] with multi-stage fusion, which combines local and global features to better predict image quality. More formally, Before using an input image, MSTRIQ first uses the Swin Transformer backbone to extract its features. After that, the features are put through a number of stages of fusion, which combines information from various layers to capture both local and global features. The quality of the image is then predicted using a regression model using the fused features. The authors combine relative ranking loss and regression loss to train the model, enabling it to learn from both absolute quality scores and relative quality differences between images. Finally, the authors employ data augmentation techniques to enlarge the training dataset and enhance the model's robustness.

Chapter 3

THE PROPOSED LEARNED POST-PROCESSING NETWORK ARCHITECTURE

Post-processing networks play a crucial role in image and video compression by enhancing the visual quality of frames and decreasing the distortion between compressed and ground truth images. These networks are designed to enhance the overall compression efficiency and the visual quality of the images, which is especially important in applications such as video streaming.

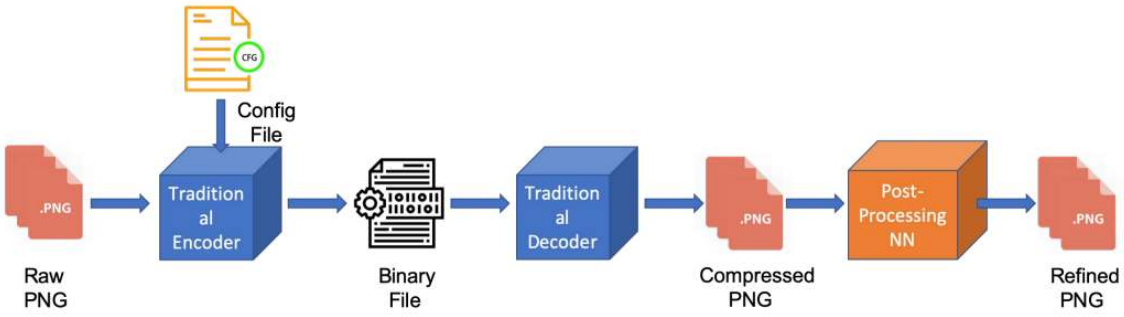


Figure 3.1: The basic architecture for image compression [Xue and Su, 2019].

The whole compression and restoration process can be summarized in four stages as in Figure 3.1:

- Encoding the frames: During compression, traditional encoding algorithms, such as JPEG, JPEG2000, HEVC, etc., or learned encoding algorithms reduce the image or video size by removing irrelevant information and exploiting the redundancy and spatial/temporal correlations.

- Bitstream: An encoder generates a compressed bitstream containing information on how to reconstruct the content. Compared to the original data, this

bitstream is smaller, which is the main goal of compression schemes.

- Decoding the frames: In order to view the content in the beginning, the compressed bitstream is transmitted to a decoder, which reconstructs the frames using the information contained in the bitstream.

- The post-processing network: After decoding, the decompressed frames might not be as high-quality as the original frames because of compression artifacts. Post-processing networks are employed to enhance the visual quality of the reconstructed frames.

Post-processing networks aim at decoded images to reduce compression artifacts, enhance details, and improve overall visual quality. They partially try to remove some of the negative effects of the compression process by applying various techniques, such as:

- Denoising: After the compression and reconstruction process, visual noise in images can be caused by some compression artifacts, such as ringing and blocking.

- Enhancement: Some required details that may have been lost during the compression process can be improved by post-processing networks. They can sharpen edges and textures and refine the details to make the reconstructed images look more similar to the ground truth images.

- Super-resolution: Super-resolution techniques can be employed to upscale the images to a higher resolution from a lower resolution by enhancing the details.

- Artifact reduction: Compression artifacts like ringing can be minimized or even eliminated using particular algorithms that successfully detect the artifacts and make appropriate corrections to remove them.

- Color correction: Some compression techniques could result in color changes or errors. Color distortions can be fixed by post-processing networks to make the images look more realistic.

With the advances in deep learning, learned post-processing networks are widely used to enhance the quality of compressed and restored videos. The network can learn from a large dataset of uncompressed and decompressed videos by using different architectures, such as convolutional neural networks (CNNs), residual networks, etc., by training appropriate weights to improve the quality and reduce the compression artifacts. They are essential for achieving a compromise between the visual quality of videos and compression efficiency. With the use of these networks, higher-quality videos can be obtained without significantly increasing the bitrate.

It is important to note that the specific techniques and algorithms used in post-processing networks can vary depending on the compression standard used, the purpose of the application, and the available resources.

3.1 System Overview

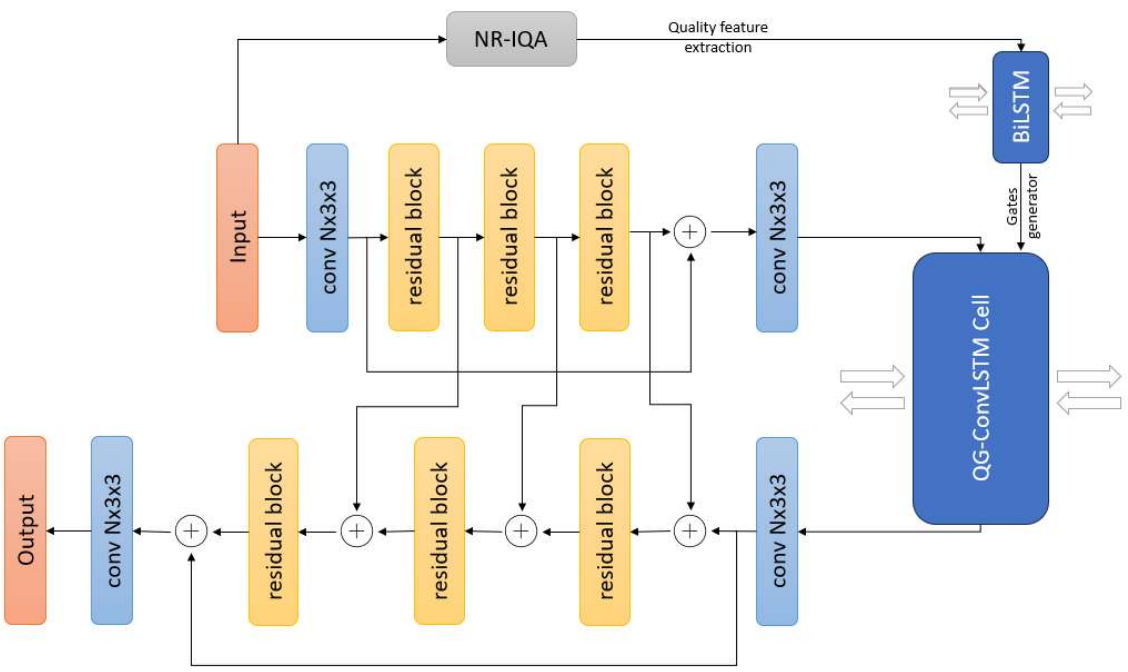


Figure 3.2: The proposed post-processing network.

The overall proposed architecture for a single frame is shown in Figure 3.2. Since videos consist of many consecutive frames and, in this case, it refers to the group of

pictures (GOP), the overall network can be thought of as the network for all frames in the GOP. The overall network can be described in four stages:

- No-reference image quality assessment (NR-IQA) and gates generator
- Spatial network
- Quality-gated ConvLSTM network
- Reconstruction network

3.2 No-Reference Image Quality Assessment (NR-IQA) and Gates Generator

In this part, quality features are extracted from the video frames after passing through a no-reference quality feature extraction process. These features, then, with the gates generator network, are used to determine the weights of the forget and input gates, which determines how much information comes from the previous and the next frames, considering their relative quality and how these adjacent frames affect the current frame in the QG-ConvLSTM cell.

3.2.1 NR-IQA Network with Transformers, Relative Ranking, and Self-Consistency (TReS)

In [Golestaneh et al., 2022], a self-attention and transformer-based network is proposed to obtain no-reference quality features from the images. Firstly, they propose a model that makes use of both local and nonlocal information contained in an image by using CNNs and the self-attention mechanism of transformers. The model basically uses transformers' sequence modeling and self-attention mechanism to learn a non-local representation of the image using the multi-scale data retrieved from different layers of CNNs, in addition to the local features that CNNs generate. The final image quality is predicted by fusing the local and non-local features. Secondly, they suggest a relative ranking loss that clearly preserves each sample's relative rankings. A triplet loss with an adaptive margin based on subjective human scores

is suggested to reduce the distance between the images with the highest (lowest) quality scores and increase the distance between the images with the highest (lowest) scores. Lastly, an equivariant transformation of the input image is suggested as a source of self-supervision in order to increase the robustness of the proposed model.

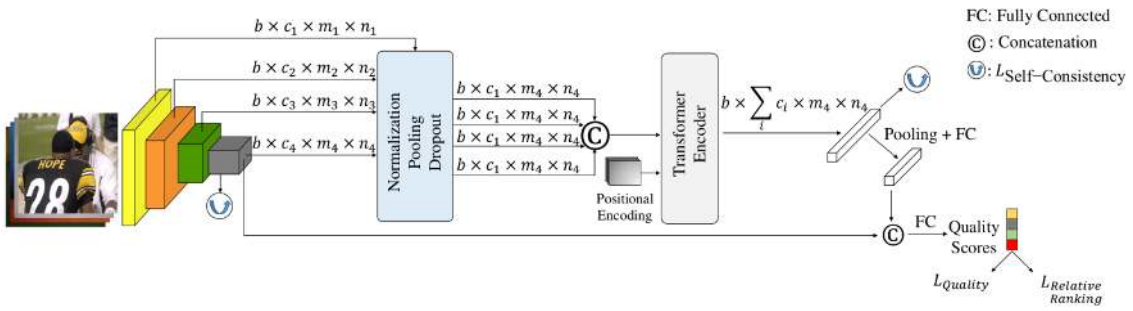


Figure 3.3: No-reference image quality assessment (NR-IQA) network diagram via TReS presented in [Golestaneh et al., 2022].

CNN-Based Feature Extraction

For a given input image with $I \in \mathbb{R}^{3\times m\times n}$ where m is the width and n is the height of the image, the aim of the network is to give a score that measures the perceptual quality. We assume that f_ϕ denotes the CNN network with trainable parameters ϕ and F_i is the feature vector from the i^{th} block of the CNN network. The CNN network consists of 4 blocks, as seen in Figure 3.3. Features, F_is, from each block of the CNN network are used to obtain the multi-scale features and given to the next normalization, pooling, and dropout layers. Because F_is for $i \in \{1, 2, 3, 4\}$ have different ranges and dimensions, normalization, pooling, and dropout processes are needed. The normalization layer of the feature F_is can be shown with the following Euclidean norm equation:

$$F_i = \frac{F_i}{\max(\|F_i\|_2, \epsilon)} \tag{3.1}$$

and the following l_2 pooling layer is:

$$P(x) = \sqrt{g * (x \bigodot x)} \tag{3.2}$$

where $\bigodot$ expresses the point-wise product, and $g(.)$ is the kernel function that uses the Hamming window and applies the Nyquist criterion to blur at the end. After the normalization, pooling, and dropout layers, all features are concatenated, let the output denote $\tilde{F}$, and the output is given to the Transformer Encoder as in Figure 3.3 [Golestaneh et al., 2022].

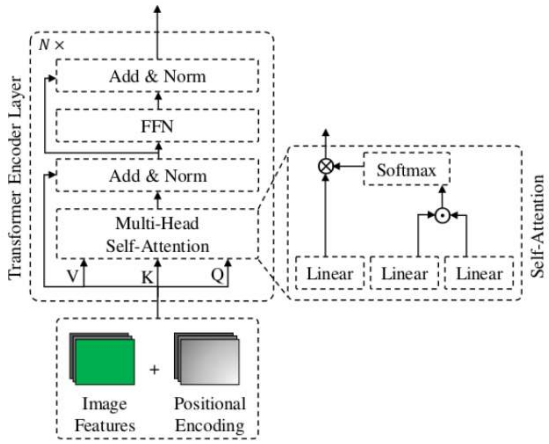


Figure 3.4: An architecture of the Transformer Encoder Layer, which contains a multi-head multi-layer self-attention module. Here, N represents how many encoder layers there are in the Transformer [Golestaneh et al., 2022].

Attention-Based Feature Computation

With previous CNN architecture, we can exploit the structure of images. Since different layers of the CNN network contain different semantic information at the end by concatenating them to a single feature, we obtain the features that represent the image's content more semantically. At each block of CNNs, we can get features at different levels. These processes are needed not to make a mistake about quality because images with less high-frequency details can result in a lower quality score. Local interactions are well-captured by CNNs using convolution with tiny kernel

sizes. However, they can only simulate the relationship between the local characteristics of the picture. By employing the self-attention mechanism, non-local features of the image can be captured with the fact that it is a non-local operation. In order to execute attention operations and collect non-local information, the Transformer encoder, which is made up of a multi-head self-attention layer and a feed-forward neural network, is suggested.

Recently, it has been known that the relation between the sequential data can be represented effectively using transformers. The proposed model can concentrate on the most important areas of the image with the use of the multi-head self-attention layer (Figure 3.4), which computes the attention weights between various positions of the input feature maps. The output of the self-attention layer is then subjected to the feed-forward neural network to further enhance the features. Thus, by combining the advantages of CNNs and Transformers, the network is able to extract both low-level and high-level features from the images.

The Transformer encoder layer takes the concatenated output, $\tilde{F}$, and the number of heads, h, as inputs. The input is initially divided into 3 distinct vector sets: the query group, the key group, and the value group. In the encoder design, a residual connection after normalization is placed in each sub-layer. Then, a feed-forward network (FFN) with two linear transformation layers is placed as in Figure 3.4. Let $\hat{F}$ denote the output of the Transformer encoder layer [Golestaneh et al., 2022].

Feature Fusion and Quality Prediction

Fully connected (FC) layers as fusion layers are employed to forecast the perceptual quality of an image by using features obtained from both local (CNN) and non-local (self-attention) networks. The following regression loss is minimized during the training process of the network.

$$\mathcal{L}_{Quality,B} = \frac{1}{N}\sum_{i}^{N} \|q_i - s_i\| \tag{3.3}$$

where B is the batch size, q_i is the estimated quality score for i^{th} image, s_i is the subjective quality score [Golestaneh et al., 2022].

Relative Ranking

By minimizing the loss described in Equation 3.3, the network can estimate the quality score efficiently. However, the ranking and inter-correlation between the frames are not considered efficiently. Therefore, the use of relative ranking information between the frames can enhance the quality prediction network. Because of the time and computation considerations, the use of the ranking for extreme cases is efficient in terms of both resources and the prediction performance of the network. For this reason, additional triplet relative ranking loss is proposed in [Golestaneh et al., 2022] to optimize the network performance. This triplet loss is calculated by using the highest, second highest, lowest, and second lowest predicted and subjective quality score values of the frames in a batch.

Self-Consistency

Another proposed technique is self-consistency, which is developed to increase the model's resilience by making use of the model's uncertainty for the input frame and its equivariant transformation during training. Therefore, the model is learned to perform consistently and produce results that are comparable for both the input image and its equivariant transformation, such as its horizontally flipped version.

3.2.2 Gates Generator

In the gates generator part, quality features are used to generate the weights of forget and update gates that are used in the next Quality-Gated ConvLSTM network [Yang et al., 2019]. It is known that in the quality enhancement of video compression tasks, only compressed videos with low quality are available as input. Since uncompressed videos are not available, ground-truth referenced quality features cannot be obtained. Therefore, no-reference feature extraction is needed to extract quality features from the images and then use them for the next network. As a no-reference image quality assessment method, the pre-trained TReS (Transformers, Relative Ranking, and Self-Consistency) based assessment algorithm is used. It is described in detail in Subsection 3.2.1.

After we obtain the quality score of each frame in a GOP using the TReS algorithm, for the $n-th$ frame, we fed the quality scores of current and T surrounding frames into the gates generator as described in Equation 2.1. The Bidirectional LSTM (BiLSTM) can learn the temporal characteristic of quality fluctuation. It is used in the gates generator network as in Figure 3.2. To determine the input and forget gate weights of the QG-ConvLSTM cell, the output of the forward and backward LSTM networks is concatenated and fed into a fully connected layer. The output of the fully connected layer is given to the QG-ConvLSTM cell as the learned gate weights.

3.3 Spatial Network

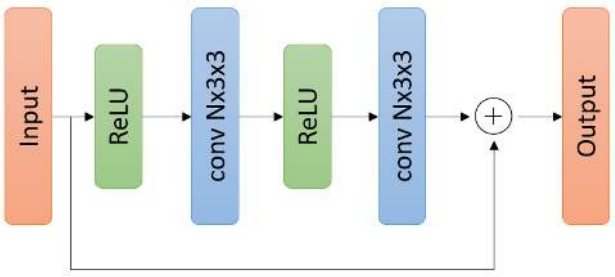


Figure 3.5: A single residual block diagram in Figure 3.2.

The spatial network is part of the whole network that extracts spatial features from the compressed frames and gives these spatial features to the QG-ConvLSTM cell as inputs as in Figure 3.2. CNN layers and residual blocks with ReLU and skip connections are employed in the spatial network. The residual block architecture is shown in Figure 3.5.

3.4 Quality-Gated Convolutional LSTM (QG-ConvLSTM) Network

QG-ConvLSTM cells take both the compressed consecutive frames and output of the gates generator derived from the no-reference quality extraction of the compressed video frames to determine the input and forget gates weights in the ConvLSTM

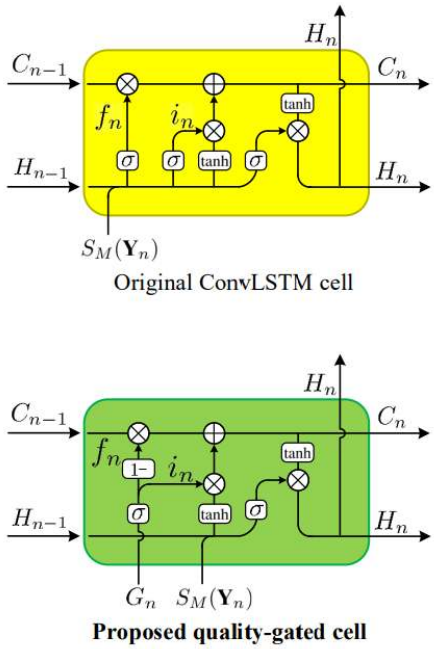


Figure 3.6: An illustration of the quality-gated convolutional LSTM cell [Yang et al., 2019].

cells. The difference is shown in Figure 3.6. The gate weights are determined by the Bidirectional LSTM cell (Figure 3.2), which takes quality features as inputs extracted from the image. The forward and backward LSTM network outputs are concatenated through channels and given as input to a fully connected layer to generate weights of input and forget gates in QG-ConvLSTM cells. According to this framework, if the quality of a compressed frame is higher than the adjacent frames, then its weight should also be higher, and the effect of the other QG-ConvLSTM cells should be lower impact so that the advantage of the information can be fully taken from the compressed frames with higher quality. Proposed QG-ConvLSTM cells are bidirectional to exploit the correlation between neighboring frames in two opposite directions.

3.5 Reconstruction Network

The reconstruction network is part of the whole network that reconstructs the enhanced frames of compressed video using the outputs from QG-ConvLSTM cells and the spatial network. Similar to the spatial network, it employs CNNs, residual blocks, and skip connections. The network is described in Figure 3.2.

Chapter 4

EXPERIMENTAL RESULTS & DATASETS

4.1 Dataset Used

4.1.1 Large-scale Diverse Video (LDV) 2.0 Dataset

The LDV 2.0 dataset is proposed in the NTIRE 2022 Challenge on SuperResolution and Quality Enhancement of Compressed Video [Yang et al., 2022a]. LDV 2.0 is an enlarged version of the original LDV dataset [Yang and Timofte, 2021] with 95 videos added. It consists of 240 videos for training, 15 videos for validation, and 15 videos for the test set. Videos are taken from YouTube and divided into ten different scene categories, including animal, city, close-up, fashion, human, indoor, park, landscape, sports, and vehicle. Frame rates of the ground-truth videos with 4K resolution range from 24 fps to 60 fps. Videos are in the format of YUV 4:2:0, which is widely used, and they are downsampled by 4 using the Lanczos filter [Turkowski, 1990] to contain fewer artifacts. Since the NTIRE 2022 Challenge includes three tracks, there are also three different tracks in the whole dataset. Track 1 targets enhancing the quality of videos compressed by HEVC (H.265) at a fixed QP. Track 2 and Track 3 both intend to super-resolve and enhance the quality of videos compressed with HEVC (H.265). Track 2 requires $\times 2$ super-resolution, while Track 3 requires x4 super-resolution. For this purpose, while Track 1 contains full-resolution raw and compressed videos, Track 2 contains raw and compressed videos that are cropped to the multiples of 16, and Track 3 contains raw and compressed videos that are cropped to the multiples of 64. All compressed videos in the dataset are compressed using HEVC (H.265) at a fixed QP value, so it means at fixed quality. All three track datasets in LDV 2.0 consist of 240 videos for training, 15 videos for validation, and 15 videos for testing. Videos are available with .mkv extension. Because our work contains only quality enhancement of compressed videos task, not a super-resolution task, we used

Video #	**Width**	**Height**	**Frame Number**	**Frame Rate (fps)**
001	960	536	300	30
002	960	536	600	60
003	960	536	600	60
004	960	536	250	25
005	960	536	300	30
006	960	536	250	25
007	960	536	250	25
008	960	536	250	25
009	960	536	250	25
011	960	536	600	60
012	960	536	600	60
013	960	536	600	60
015	960	536	600	60

Table 4.1: LDV 2.0 Track 1, Validation video dataset information

Video Number	**Width**	**Height**	**Frame Number**	**Frame Rate (fps)**
001	960	536	250	25
002	960	536	300	30
003	960	536	300	30
004	960	536	250	25
005	960	536	600	60
006	960	536	300	30
007	960	536	240	24
009	960	536	250	25
011	960	536	600	60
012	960	536	600	60
013	960	536	300	30
014	960	536	300	30
015	960	536	300	30

Table 4.2: LDV 2.0 Track 1, Test video dataset information

only the Track 1 dataset for our research. The width and height of video frames are $960x536$, as seen in Figure 4.1 and Figure 4.2. Frame rate, frame number, and resolution information of the validation and test videos in the dataset can be found in Figure 4.1 and Figure 4.2, respectively. The LDV 2.0 dataset is accessible at `https://github.com/RenYang-home/LDV_dataset`.

4.1.2 Used Dataset Compressed With VVC

Since this work focuses on deep learning-based post-processing algorithms for enhancing the quality of compressed videos, we need both compressed and uncompressed video data. Thus, we can obtain a loss that measures the quality difference or distortion between the compressed and original ground-truth video frames to be able to train the whole network.

For our research, we used LDV 2.0 Track 1 raw videos for uncompressed video data. In the LDV 2.0 dataset, all compressed videos are compressed with the HEVC video compression algorithm at a fixed QP value. However, for compressed video data, instead of using these compressed videos, we created a new compressed video dataset by compressing the raw videos in Track 1 using a VVC codec with a fixed GOP size, not at a fixed QP. There are two main reasons for this. The first one is that recently, it has been shown that the VVC (H.266) compression algorithm can achieve better performance compared to the HEVC (H.265) compression algorithm. Since its performance is better, we used the VVC codec; thus, we can both compare and compete with the state-of-the-art results for this task. The second one is that we want our post-processing network to be able to enhance the quality of bidirectionally compressed videos within the group of pictures (GOPs). In a bidirectional video compression scenario, there are I, P, and B frames, and since the decoding order of frames and the reference types are varying, the compression qualities of these frames are not fixed in a GOP, depending on frame type, reference frame type, and GOP size. When you give a specific QP value and a GOP size to the VVC encoder to apply the VVC compression algorithm, it adjusts the QP value of every frame in the GOP hierarchically, considering the decoding order and reference type.

All compressions are applied in the YUV domain. While making the dataset ready for use, first, we converted MKV videos that were initially available to 4:2:0 YUV format using ffmpeg, which is the widely used tool for multimedia operations and applications of different video compression codecs, with the following command:

ffmpeg -i 001.mkv -pix_fmt yuv420p 001.yuv

Then, we compress all videos hierarchically with VVC codec in the YUV domain with the following command:

EncoderApp -c config.cfg –InputFile=001.yuv –InputBitDepth=8 –InputChromaFormat=420 –FrameRate=30 –SourceWidth=960 –SourceHeight=536 -q 40 –FramesToBeEncoded=300 –OutputBitDepth=8 –OutputBitDepthC=8 -b 001.bin -ip 8 -g 8 -o 001.yuv

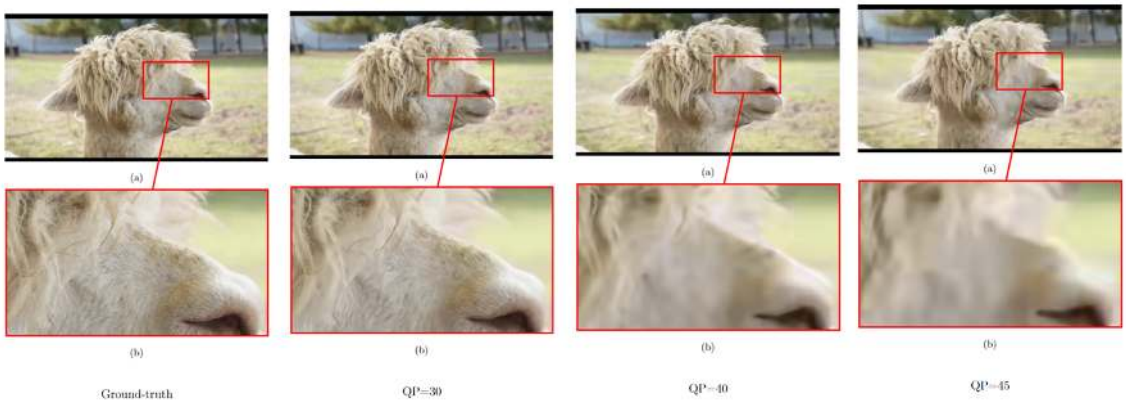


Figure 4.1: Quality comparison of decompressed images with different quantization parameter (QP) values by VVC codec.

Here we set *-g* parameter, which specifies the size of the cyclic GOP structure, as 8 images and *-q*, which specifies the base value of the quantization parameter (QP) controlling rate-distortion trade-off, as 40. The effect of the QP parameter can be seen in Figure 4.1. We can easily see the difference between the ground-truth image and compressed images with different quality. When we focus on the middle of the sheep's face, we can see that the amount of blur is most significant in the last image compressed with $QP = 45$ and least in the first ground-truth image as in Figure 4.1. We can conclude that when the QP value increases, the compression ratio increases and the picture becomes more blurry. Since the YUV videos are

in the 4:2:0 format, *–InputChromaFormat* parameter is set as 420. *–InputBitDepth* and *–OutputBitDepth* parameters, which specify the bit depth of the input and output video, respectively, are set as 8. The other parameters, such as width: *–SourceWidth*, height: *–SourceHeight*, frame number: *–FramesToBeEncoded*, and frame rate: *–FrameRate*, are determined by information about the dataset. Finally, *-ip* parameter, which specifies the intra-frame period, is set as 8.

For the sake of the simplicity of the quality analysis visually, we convert them to RGB frames using ffmpeg with the following command:

ffmpeg -y -pix_fmt yuv420p -s 960×536 -i 001.yuv 001/%3d.png

However, in the end, all metric calculations are made, the model is applied, and the results are evaluated in YUV color space after the RGB-YCbCr conversion by using ITU-R BT.709 coefficients.

Because of its complexity, it takes too much time to compress the videos with the VVC codec. The compression time of a single video can vary between 1-2 days. Since we needed to compress 270 videos in total and compressing them sequentially takes too much time, we used the multiprocessing technique, and by assigning each video that needed to be compressed to a single CPU, we reduced the compression time to 7-8 days efficiently.

In short, we used the original raw video data in the LDV 2.0 dataset as ground truth videos, and we compressed the raw videos in the LDV 2.0 track 1 dataset by using the VVC (H.266) codec to use them as low-quality compressed videos. Thus, we can try to enhance compressed videos by optimizing the loss between compressed and uncompressed video frames.

4.2 Evaluation Metrics

The compressed quality of frames can be evaluated using different metrics such as MSE, PSNR, MS-SSIM, and VMAF.

4.2.1 PSNR

PSNR is a widely used metric in image and video compression tasks. PSNR is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is commonly used to quantify image reconstruction quality subject to lossy compression. Because many signals have a wide dynamic range, PSNR is usually expressed as a logarithmic quantity using the decibel scale. PSNR is calculated by using the mean squared reconstruction error (MSE) after reconstruction and is inversely proportional to MSE. A higher PSNR value is obtained when the reconstructed image is closer to the ground truth image.

PSNR is basically formulated with the mean squared error (MSE). Given a ground truth, noise-free, one-channel image I and its noisy approximation K, MSE is defined as

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{4.1}$$

Signals can have a wide dynamic range, so PSNR, a widely used metric in image and video compression, is usually expressed in decibels, which is a logarithmic scale. The PSNR (in dB) is defined as

$$PSNR = 10log_{10}(\frac{MAX_I^2}{MSE}) \tag{4.2}$$

So, unlike the MSE loss, higher PSNR means more enhanced images in terms of visual quality compared to the noise-free ones.

4.2.2 MS-SSIM

PSNR gives over-smoothed and blurry results since the algorithm removes not only the noise but also a part of the textures needed to stay. SSIM is a quality reconstruction metric that also considers the similarity of the edges and high-frequency content between the denoised image and the original image. To have a good SSIM

measure, an algorithm needs to remove the noise while also preserving the edges of the objects. Hence, SSIM is a better measure in terms of quality. SSIM involves one number per pixel, while PSNR gives you an average value for the whole image. Usually, PSNR is a good metric for RGB images. At the same time, SSIM is a good metric for luminance and chrominance images since SSIM is correlated with the quality and perception of the human visual system.

The SSIM algorithm predicts the perceived quality of digital television and cinematic visuals, as well as other types of digital images and videos. It is also widely used to compare the similarity of two images. The SSIM index is a full-reference metric, which means that image quality is measured or predicted using an initial uncompressed or distortion-free ground truth image as a reference. The SSIM index evolved from previous methods PSNR and MSE, since they are incompatible with human perceptual physiology, so they are unable to represent the human visual comparison fully. The distinction between SSIM with MSE and PSNR is that these methodologies assess absolute errors. However, the concept behind structural information is that pixels have substantial interdependencies, especially when they are spatially adjacent. Since these dependencies provide crucial information about the structure of the objects in the visual scene, SSIM can provide more accurate results in terms of visual quality.

The SSIM metric is computed on several windows of an image. The distance between two windows x and y of size $N \times N$ can be calculated using the following equation [Wang et al., 2003]:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.3}$$

where;

- μ_x the average of x;
- μ_y the average of y;
- σ_x^2 the variance of x;

- σ_y^2 the variance of y;
- σ_{xy} the covariance of x and y;
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ two variables to stabilize the division with weak denominator;
- L the dynamic range of the pixel values (it is generally $2^{\#\text{bits per pixel}}$-1);

and finally, $k_1 = 0.01$, $k_2 = 0.03$ are set by default.

The given formula is only applicable to the brightness of the image, which is utilized to determine the quality of an image. The SSIM score results vary from -1 to $+1$. Remarkably, the total originality of the samples results in a rating of $+1$. The measure is usually determined using an $8x8$ pixel window. Although the window can be shifted by a pixel, scientists advocate clustering windows together to decrease the complexity of the computations.

Multiscale SSIM (MS-SSIM) is an improved version of SSIM that performs at various scales via a multi-step downsampling method, similar to multiscale processing in the primary visual system. It has been demonstrated that it succeeds as well as or more effectively than SSIM using different databases of subjective images and videos.

4.2.3 VMAF

Video Multi-Method Assessment Fusion (VMAF) [Zhi Li, 2016] is a perceptual video quality assessment metric developed by Netflix. VMAF is widely used in video streaming to optimize video encoding and delivery. It is developed to evaluate perceptual video quality consistent with human perception. VMAF contains a combination of different quality assessment algorithms to obtain a single score for video quality assessment.

The standard video quality assessment metrics, such as PSNR and SSIM, failed to represent human perception fully. Differently, VMAF is a video quality metric designed to combine human perception modeling with machine learning. It can

be easily applicable since it is integrated into popular and frequently used video analysis tools such as ffmpeg. VMAF tries to measure video quality and give close and reliable results to how people rate that video by considering different aspects of video quality measurement, such as the spatial and temporal features.

4.3 Training Details & Results

4.3.1 Loss Function

We train our quality enhancement post-processing network by minimizing the loss function

$$\mathcal{L} = \frac{1}{B \times N} \sum_{j=1}^{B} \sum_{i=1}^{N} D(x_{ij}, x_{ij}^{D}) \tag{4.4}$$

where B is the batch size, N is the number of frames in a GOP, i is the frame number, j is the batch number, x_{ij} is the ground-truth image, x_{ij}^{D} is the enhanced image by the proposed network, and the $D(.)$ is the distortion between the ground-truth and enhanced frames. In our case, as a distortion measure, we used the MSE loss.

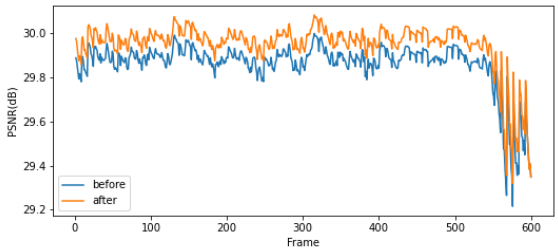
4.3.2 Settings

The proposed network is trained on the LDV 2.0 video dataset, which is described in Section 4.1. The resolution of the videos is 960×536, 240 video is used for training, and 15 videos are used for validation and testing. The QP value of the compressed frames is set to 40, and videos are compressed hierarchically in a GOP with VVC codec. Additionally, the GOP size is set to 9 sequential frames per video, and the batch size is set to 4 videos. The kernel size of the CNN layers in Figure 3.2 is chosen as 3×3, and the number of filters, N, is chosen as 64. The number of surrounding frames in the gates generator, T in Equation 2.1, is set as 4. We chose the parameters that are used in QG-ConvLSTM cells as follows: the number of features is 64, the number of layers is 1, and the kernel size is 3×3. Finally, we used AdamW as an optimizer. Instead of training the whole network, we use

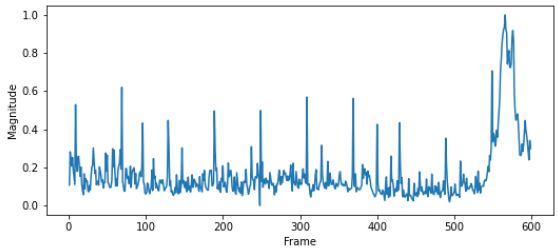
the pre-trained NR-IQA algorithm with TReS for the quality feature prediction process. Pre-trained TReS models trained with different datasets are available at `https://github.com/isalirezag/TReS`.

4.3.3 Results

In this part, we assess the different results and compare the performance of our network in terms of different quality metrics: PSNR, MS-SSIM, and VMAF.



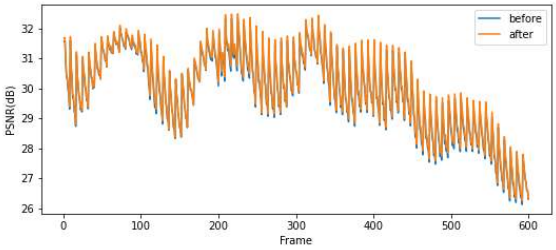
(a)



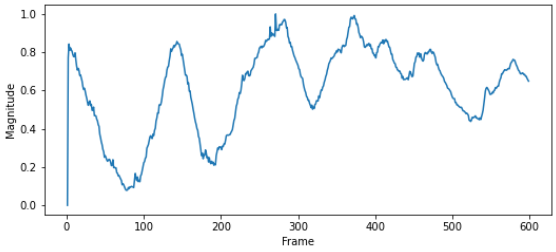
(b)

Figure 4.2: Quality and motion plots of Validation Video #2: (a) The quality change plot (b) The motion change plot.

First of all, as an ablation study, quality changes in terms of PSNR and motion change of validation and test videos are plotted. In motion change plots, to get an intuition about the motion changes of the videos, the sum of the magnitude of optical flow vectors between the video frames is plotted using The Gunnar-Farneback optical flow algorithm. Additionally, in quality change plots, the quality of videos before and after the proposed post-processing network in terms of PSNR is plotted simultaneously, frame by frame. In Figure 4.2, we can observe that when the mag-



(a)

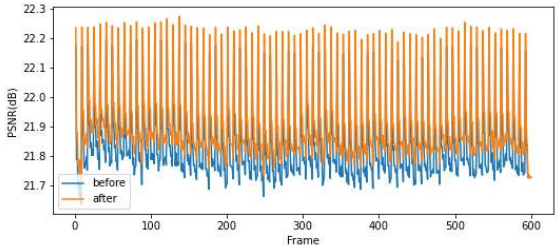


(b)

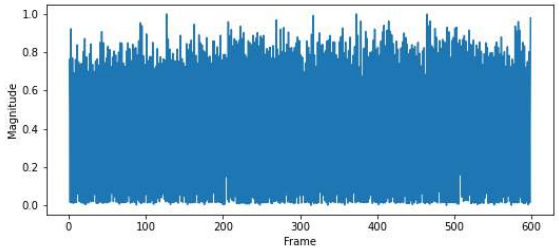
Figure 4.3: Quality and motion plots of Validation Video #3: (a) The quality change plot (b) The motion change plot.

nitude of motion is lower, our model can enhance the quality of video frames better since it can exploit the inter-frame correlation between adjacent frames effectively. Furthermore, when we look at Figure 4.3, we can conclude that when the motion change between the frames is getting higher, then the quality of both compressed and enhanced frames gets lower, and this supports our claim by showing that similar adjacent frames create spatial and temporal redundancy and using these adjacent frames as a reference in both compression and enhancement task, compression quality can be enhanced. Another point is if we look at Figure 4.4, we can observe that when the motion changes fluctuate between the video frames, then the quality of compressed frames also fluctuates. The rest of the motion and quality plots of all validation and test videos are given in Appendix A.

The average quality values of the validation videos in terms of different metrics before and after the proposed post-processing network are shown in Table 4.3. The proposed post-processing model is applied in the YCbCr domain, and all metric calculations are also made in the YCbCr domain. For the sake of simplicity, the change



(a)



(b)

Figure 4.4: Quality and motion plots of Validation Video #13: (a) The quality change plot (b) The motion change plot.

in these metrics after passing through the network is shown in Table 4.4. When the results are analyzed, the changes in Table 4.4 are generally positive, which means that the proposed network is able to enhance the quality of the compressed videos in terms of all metrics we use to measure the quality. While the change in quality between the compressed video and the enhanced video in terms of a specific metric is higher than the other metrics for a specific video, on the other side, the change in quality between the compressed video and the enhanced video in terms of the same metric can be lower than the other metrics for another video. To give an example, in Table 4.3, the PSNR score decreases from the third validation video to the fourth one while the VMAF score increases. This case stems from the fact that the different metrics measure the different sides of quality and give importance to diverse quality components. For instance, the VMAF quality metric is a mixture of more than one metric and usually gives good insights about the perceptual quality of an image. It is designed to represent the human visual system more effectively. However, the PSNR quality metric gives more weight to the distortion between the two frames,

	Before Post-Processing			After Post-Processing		
Video	YUV-PSNR	YUV-SSIM	VMAF	YUV-PSNR	YUV-SSIM	VMAF
1	**35.778914**	**0.942447**	60.610713	35.539688	0.941706	**60.628099**
2	34.649109	0.940329	67.197765	**34.676222**	**0.94065**	**67.325803**
3	34.495246	0.927336	66.680823	**34.541216**	**0.927789**	**66.86387**
4	33.372355	0.917294	72.321273	**33.436808**	**0.91846**	**72.524341**
5	34.33627	0.941378	**67.524766**	**34.34039**	**0.94154**	67.507333
6	34.98791	0.958955	71.084916	**35.073151**	**0.959457**	**71.126365**
7	35.039875	0.911053	59.818836	**35.126251**	**0.912733**	**60.710372**
8	34.786413	0.930605	71.354884	**34.816443**	**0.93159**	**71.594831**
9	33.532782	0.924524	68.435289	**33.547083**	**0.92543**	**68.737852**
11	27.115139	0.820243	52.368061	**27.15483**	**0.822011**	**52.645211**
12	31.595992	0.875001	57.55527	**31.646281**	**0.876152**	**57.959276**
13	26.517817	0.807935	62.275307	**26.560625**	**0.810271**	**62.458148**
15	32.952663	0.936521	65.498495	**32.985972**	**0.936874**	**65.557549**

Table 4.3: Post-processing network results on the validation dataset, trained with VVC compressed dataset at QP=40.

which may not be visible to the human eye, obviously. Additionally, for the sake of a fair comparison of our post-processing model with the proposed model in [Yang et al., 2019], we trained our model with the LDV 2.0 dataset, which is samely compressed with HEVC standard using HM 16.0 with the Low Delay P (LDP) mode at $QP = 42$ as in [Yang et al., 2019]. The results on both validation and test datasets are given in Table 4.5, 4.6, 4.7, 4.8. To evaluate the effects of our proposed model visually and subjectively, examples of the uncompressed, compressed, and enhanced frames with the proposed network are presented in detail in Appendix B.

	Change in Metric After Passing The Network		
Video	YUV-PSNR	YUV-SSIM	VMAF
1	-0.239226	-0.000741	0.017386
2	0.027113	0.000321	0.128038
3	0.04597	0.000453	0.183047
4	0.064453	0.001166	0.203068
5	0.00412	0.000162	-0.017433
6	0.085241	0.000502	0.041449
7	0.086376	0.00168	0.891536
8	0.03003	0.000985	0.239947
9	0.014301	0.000906	0.302563
11	0.039691	0.001768	0.27715
12	0.050289	0.001151	0.404006
13	0.042808	0.002336	0.182841
15	0.033309	0.000353	0.059054

Table 4.4: Post-processing network results on the validation dataset, trained with VVC compressed dataset at QP=40.

	Before Post-Processing			After Post-Processing		
Video	YUV-PSNR	YUV-SSIM	VMAF	YUV-PSNR	YUV-SSIM	VMAF
1	34.969038	0.935956	**54.798094**	**34.977328**	**0.936856**	54.785601
2	33.295074	0.922384	57.999645	**33.388663**	**0.923567**	**58.18432**
3	33.508589	0.919853	62.28197	**33.636053**	**0.921373**	**63.100996**
4	32.160682	0.901955	66.413652	**32.356093**	**0.905051**	**68.248669**
5	33.275252	0.931492	62.642269	**33.344926**	**0.932791**	**63.129033**
6	34.051575	0.95302	67.20061	**34.234778**	**0.954768**	**68.601683**
7	33.865206	0.896763	52.303864	**34.063236**	**0.900365**	**53.569449**
8	33.553491	0.916651	65.152077	**33.68701**	**0.91949**	**66.139175**
9	32.557828	0.911727	62.810869	**32.658082**	**0.914023**	**63.824247**
11	26.602933	0.813967	49.984931	**26.70505**	**0.815779**	**50.085812**
12	31.280144	0.867779	55.565213	**31.427335**	**0.870445**	**55.612656**
13	25.180364	0.751094	**52.99134**	**25.254989**	**0.751197**	52.344464
15	32.175339	0.92495	61.21444	**32.315987**	**0.926416**	**61.595229**

Table 4.5: Post-processing network results on the validation dataset, trained with HM 16.0 compressed dataset at QP=42.

	Change in Metrics After Passing The Network		
Video	YUV-PSNR	YUV-SSIM	VMAF
1	0.00829	0.0009	-0.012493
2	0.093589	0.001183	0.184675
3	0.127464	0.00152	0.819026
4	0.195411	0.003096	1.835017
5	0.069674	0.001299	0.486764
6	0.183203	0.001748	1.401073
7	0.19803	0.003602	1.265585
8	0.133519	0.002839	0.987098
9	0.100254	0.002296	1.013378
11	0.102117	0.001812	0.100881
12	0.147191	0.002666	0.047443
13	0.074625	0.000103	-0.646876
15	0.140648	0.001466	0.380789

Table 4.6: Post-processing network results on the validation dataset, trained with HM 16.0 compressed dataset at QP=42.

	Before Post-Processing			After Post-Processing		
Video	YUV-PSNR	YUV-SSIM	VMAF	YUV-PSNR	YUV-SSIM	VMAF
1	34.738163	0.913768	47.62126	**34.854434**	**0.915969**	**47.650044**
2	31.772576	0.871602	53.749136	**31.866991**	**0.872817**	**54.009797**
3	30.615813	0.889835	69.397716	**30.720613**	**0.891599**	**70.318332**
4	36.649662	0.950816	54.116194	**36.815645**	**0.952987**	**55.344035**
5	32.352053	0.866199	52.042632	**32.469672**	**0.867854**	**52.677534**
6	29.727868	0.885304	52.307746	**29.811471**	**0.887139**	**52.598534**
7	30.400532	0.883261	57.370391	**30.591137**	**0.886961**	**58.059742**
9	31.773416	0.88591	60.118573	**31.958813**	**0.890314**	**61.67429**
11	29.298551	**0.754668**	**45.310916**	**29.350919**	0.754392	44.712166
12	**24.046219**	0.83593	**61.58937**	23.998557	**0.838241**	60.910047
13	30.468497	0.858633	**48.250653**	**30.568091**	**0.859875**	47.914829
14	32.783561	0.898133	53.839276	**32.900049**	**0.899787**	**54.693802**
15	30.307854	0.858812	63.854942	**30.397017**	**0.860073**	**64.538586**

Table 4.7: Post-processing network results on the test dataset, trained with HM 16.0 compressed dataset at QP=42.

	Change in Metrics After Passing The Network		
Video	YUV-PSNR	YUV-SSIM	VMAF
1	0.116271	0.002201	0.028784
2	0.094415	0.001215	0.260661
3	0.1048	0.001764	0.920616
4	0.165983	0.002171	1.227841
5	0.117619	0.001655	0.634902
6	0.083603	0.001835	0.290788
7	0.190605	0.0037	0.689351
9	0.185397	0.004404	1.555717
11	0.052368	-0.000276	-0.59875
12	-0.047662	0.002311	-0.679323
13	0.099594	0.001242	-0.335824
14	0.116488	0.001654	0.854526
15	0.089163	0.001261	0.683644

Table 4.8: Post-processing network results on the test dataset, trained with HM 16.0 compressed dataset at QP=42.

Chapter 5

CONCLUSION

5.1 Discussions

Video sequences consist of consecutive single frames. Since images are represented using pixels and spatially close pixels contain similar information, exploiting the redundancy between the pixels of the images, the size of the stored image can be decreased, and thus, compression efficiency can be enhanced. Therefore, using the advantages of intra-frame correlations, every single image that creates the whole video stream can be compressed efficiently. This brings the fact that advances in image compression algorithms can also be applied to enhance video compression algorithms. In addition to intra-frame correlations, video sequences contain inter-frame correlations between the sequential frames. A little time passes between two consecutive frames, and therefore, a small change in motion occurs. High similarity between adjacent frames enables a video stream to be compressed more efficiently to take up less space.

New studies that focus on utilizing spatial redundancy in videos have continued to be carried out. In this research work, Quality-Gated ConvLSTM cells are used to take advantage of inter-frame correlation between sequential frames. To determine the input and forget gate weights of the ConvLSTM cells, a pre-trained quality feature assessment method that is based on Transformers, relative ranking, and self-consistency concepts is applied. Our proposed scheme is able to increase the quality of compressed videos in terms of some quality measurement metrics; however, it is not sufficient to fully utilize it. There are many issues that need to be improved. With the advances in the applications of this task, various algorithms can be applied to improve the performance of quality enhancement techniques.

5.2 Future Research Directions

This research study focuses on designing an effective architecture for the quality enhancement of compressed videos that are compressed with various video compression algorithms. Mainly, convolutional LSTM cells are used in this work to exploit spatial redundancy between frames; however, with the popularization of applications on Transformers and self-attention that can also be used to extract and reveal the relations between the sequential data, new architectures based on these concepts can be applied for quality enhancement. Additionally, we used a dataset that is compressed by one of the standard video compression codecs, VVC, to train our network. There is a hierarchical compression choice in the VVC codec; however, it compresses the frames in a group-of-pictures (GOP) with close quantization parameter (QP) values. Since the quality of compressed sequential frames in a GOP is closer, our network is not able to fully utilize the quality difference between consecutive frames. For this reason, as a future work, we can apply this quality enhancement post-processing network to end-to-end rate-distortion optimized learned hierarchical bidirectional video compression algorithm proposed in [Yılmaz and Tekalp, 2021] and explained in Subsection 2.3.1. Since, in this network, frames in a GOP are compressed hierarchically using bidirectional references, the quality values of compressed frames decrease at each hierarchical level. Therefore, we can obtain compressed frames with diverse quality, and we can utilize our network more efficiently. Furthermore, we use only a few features coming from the quality assessment process in the gates generator network. Therefore, the network may not utilize the quality features of compressed videos. In the next studies, we can try to increase the number of these features used for specifying the input and forget gate weights. If we apply this network to the learned hierarchical bidirectional video compression network in [Yılmaz and Tekalp, 2021], we can obtain the quality features from the decoder of the compression algorithm without using any no-reference quality assessment algorithm since the decoder has that information. Moreover, since we know the size of the compressed video frames, we can use the bitrates of the compressed frames as features for the gates generator network. The decoder of the VVC (H.266) codec

that we applied to compress videos also gives the bit-rate and QP information of each frame compressed. In future work, we can combine these quality parameters and bit-rates with the features coming from the various no-reference quality assessment techniques thus, we can improve the representation of images in terms of qualities that are used to utilize spatial redundancy between the frames in a video stream. Moreover, we use different metrics to measure quality improvement and distortion; however, we trained our network using MSE loss, which may not fully represent perceptual image quality as humans perceive it. Different loss functions can be developed to train our network to be able to represent the human visual system more effectively. Lastly, in this work, we used pre-trained TReS architecture for no-reference quality assessment. This network also can be trained for our dataset to obtain more reliable results.

BIBLIOGRAPHY

[Ballé et al., 2016] Ballé, J., Laparra, V., and Simoncelli, E. P. (2016). Density modeling of images using a generalized normalization transformation.

[Ballé et al., 2018] Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. (2018). Variational image compression with a scale hyperprior.

[Ding et al., 2021] Ding, Q., Shen, L., Yu, L., Yang, H., and Xu, M. (2021). Patch-wise spatial-temporal quality enhancement for hevc compressed video. *IEEE Transactions on Image Processing*, 30:6459–6472.

[Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[Ferzli and Karam, 2009] Ferzli, R. and Karam, L. J. (2009). A no-reference objective image sharpness metric based on the notion of just noticeable blur (jnb). *IEEE transactions on image processing*, 18(4):717–728.

[Golestaneh et al., 2022] Golestaneh, S. A., Dadsetan, S., and Kitani, K. M. (2022). No-reference image quality assessment via transformers, relative ranking, and self-consistency.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

[Guan et al., 2019] Guan, Z., Xing, Q., Xu, M., Yang, R., Liu, T., and Wang, Z. (2019). Mfqe 2.0: A new approach for multi-frame quality enhancement on compressed video. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):949–963.

[Ke et al., 2021] Ke, J., Wang, Q., Wang, Y., Milanfar, P., and Yang, F. (2021). Musiq: Multi-scale image quality transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5128–5137, Los Alamitos, CA, USA. IEEE Computer Society.

[Liu et al., 2022] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., and Guo, B. (2022). Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Liu et al., 2021] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[Minnen et al., 2018] Minnen, D., Ballé, J., and Toderici, G. (2018). Joint autoregressive and hierarchical priors for learned image compression.

[Mittal et al., 2012] Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708.

[Narvekar and Karam, 2009] Narvekar, N. D. and Karam, L. J. (2009). A no-reference perceptual image sharpness metric based on a cumulative probability of blur detection. In *2009 International Workshop on Quality of Multimedia Experience*, pages 87–91. IEEE.

[Sadaka et al., 2008] Sadaka, N. G., Karam, L. J., Ferzli, R., and Abousleman, G. P. (2008). A no-reference perceptual image sharpness metric based on saliency-

weighted foveal pooling. In *2008 15th IEEE International Conference on Image Processing*, pages 369–372. IEEE.

[Segall et al., 2018] Segall, C., Baroncini, V., Boyce, J., Chen, J., and Suzuki, T. (2018). Jvet-h1002: Joint call for proposals on video compression with capability beyond hevc.

[Shannon, 1993] Shannon, C. E. (1993). *Coding Theorems for a Discrete Source With a Fidelity CriterionInstitute of Radio Engineers, International Convention Record, vol. 7, 1959.*, pages 325–350.

[Sullivan et al., 2012] 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.

[Turkowski, 1990] Turkowski, K. (1990). *Filters for Common Resampling Tasks*, page 147–165. Academic Press Professional, Inc., USA.

[Varadarajan and Karam, 2008] Varadarajan, S. and Karam, L. J. (2008). An improved perception-based no-reference objective image sharpness metric using iterative edge refinement. In *2008 15th IEEE international conference on image processing*, pages 401–404. IEEE.

[Wang et al., 2020] Wang, J., Deng, X., Xu, M., Chen, C., and Song, Y. (2020). Multi-level wavelet-based generative adversarial network for perceptual quality enhancement of compressed video. In *European Conference on Computer Vision*, pages 405–421. Springer.

[Wang et al., 2022] Wang, J., Fan, H., Hou, X., Xu, Y., Li, T., Lu, X., and Fu, L. (2022). Mstriq: No reference image quality assessment based on swin transformer with multi-stage fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1269–1278.

[Wang et al., 2021] Wang, J., Xu, M., Deng, X., Shen, L., and Song, Y. (2021). Mw-gan+ for perceptual quality enhancement on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1.

[Wang et al., 2003] Wang, Z., Simoncelli, E., and Bovik, A. (2003). Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2.

[Wiegand et al., 2003] Wiegand, T., Sullivan, G., Bjøntegaard, G., and Luthra, A. (2003). Overview of the h.264/avc video coding standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13:560 – 576.

[Xue and Su, 2019] Xue, Y. and Su, J. (2019). Attention based image compression post-processing convolutional neural network.

[Yang et al., 2020] Yang, R., Mentzer, F., Gool, L. V., and Timofte, R. (2020). Learning for video compression with hierarchical quality and recurrent enhancement.

[Yang et al., 2019] Yang, R., Sun, X., Xu, M., and Zeng, W. (2019). Quality-gated convolutional lstm for enhancing compressed video. *2019 IEEE International Conference on Multimedia and Expo (ICME)*.

[Yang and Timofte, 2021] Yang, R. and Timofte, R. (2021). NTIRE 2021 challenge on quality enhancement of compressed video: Dataset and study. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

[Yang et al., 2022a] Yang, R., Timofte, R., et al. (2022a). NTIRE 2022 challenge on super-resolution and quality enhancement of compressed video: Dataset, methods and results. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

[Yang et al., 2017] Yang, R., Xu, M., and Wang, Z. (2017). Decoder-side hevc quality enhancement with scalable convolutional neural network. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 817–822. IEEE.

[Yang et al., 2018a] Yang, R., Xu, M., Wang, Z., and Li, T. (2018a). Multi-frame quality enhancement for compressed video. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[Yang et al., 2018b] Yang, R., Xu, M., Wang, Z., and Li, T. (2018b). Multi-frame quality enhancement for compressed video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6664–6673.

[Yang et al., 2022b] Yang, S., Wu, T., Shi, S., Lao, S., Gong, Y., Cao, M., Wang, J., and Yang, Y. (2022b). Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1191–1200.

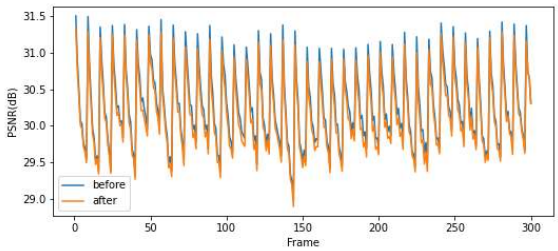
[Yılmaz and Tekalp, 2021] Yılmaz, M. A. and Tekalp, A. M. (2021). End-to-end rate-distortion optimized learned hierarchical bi-directional video compression.

[Zhang et al., 2023] Zhang, W., Zhai, G., Wei, Y., Yang, X., and Ma, K. (2023). Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14071–14081.

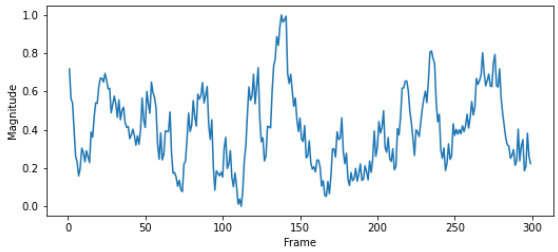
[Zhi Li, 2016] Zhi Li, A. A. (2016). Toward a practical perceptual video quality metric. *Netflix TechBlog*.

Appendix A

ABLATION STUDY

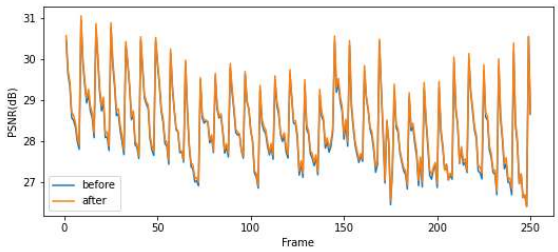


(a)

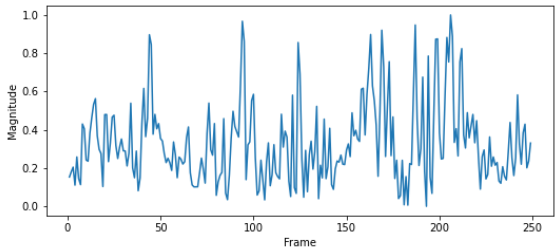


(b)

Figure A.1: Quality and motion plots of Validation Video #1: (a) The quality change plot (b) The motion change plot.

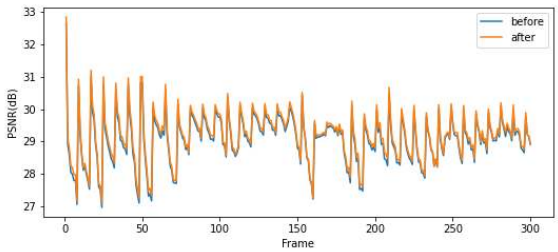


(a)

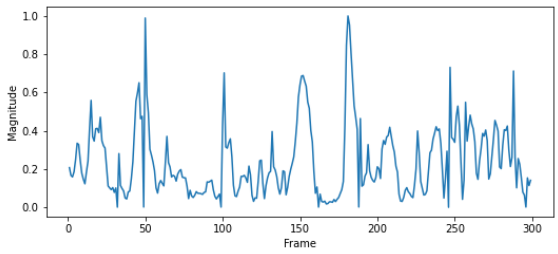


(b)

Figure A.2: Quality and motion plots of Validation Video #4: (a) The quality change plot (b) The motion change plot.

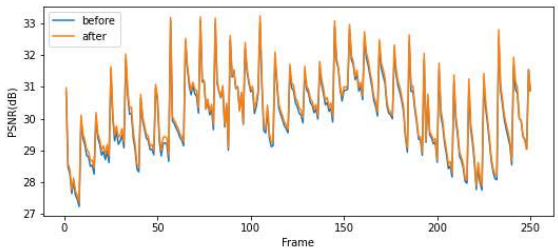


(a)

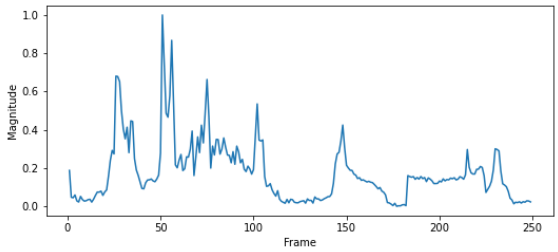


(b)

Figure A.3: Quality and motion plots of Validation Video #5: (a) The quality change plot (b) The motion change plot.

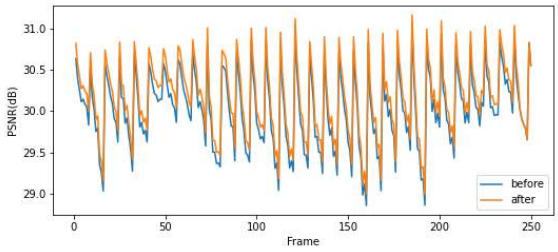


(a)

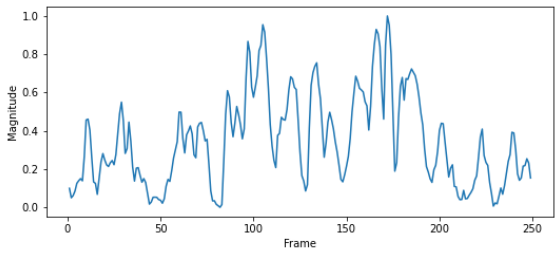


(b)

Figure A.4: Quality and motion plots of Validation Video #6: (a) The quality change plot (b) The motion change plot.

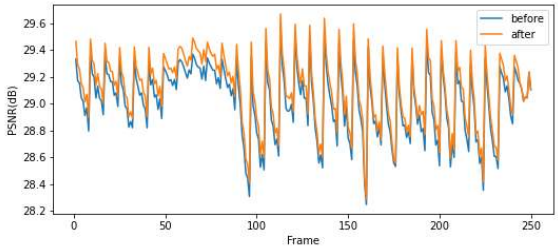


(a)

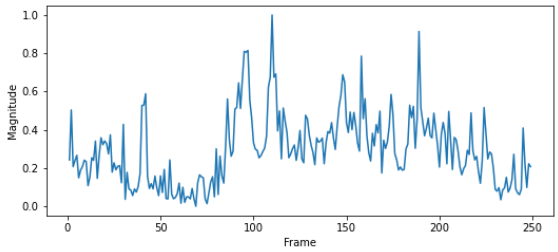


(b)

Figure A.5: Quality and motion plots of Validation Video #7: (a) The quality change plot (b) The motion change plot.

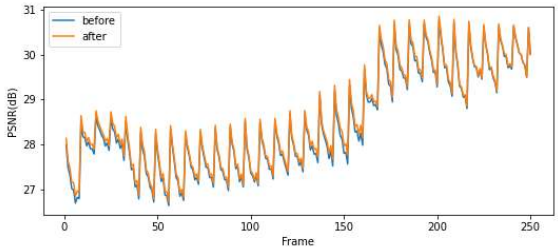


(a)

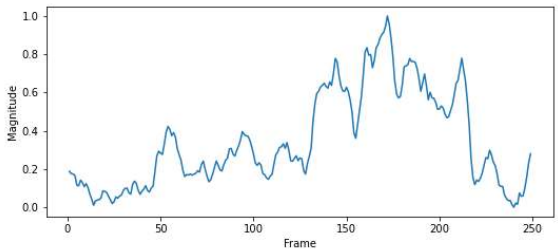


(b)

Figure A.6: Quality and motion plots of Validation Video #8: (a) The quality change plot (b) The motion change plot.

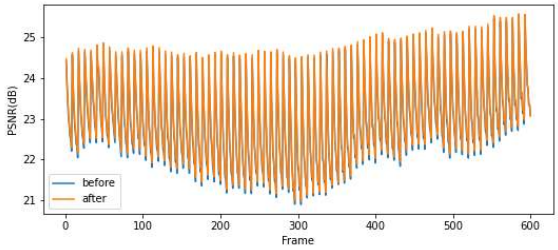


(a)

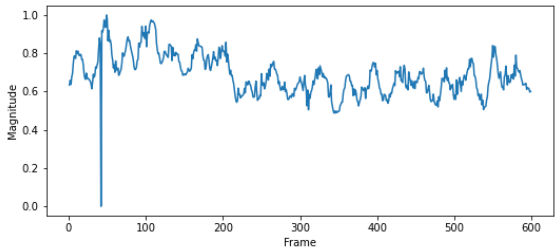


(b)

Figure A.7: Quality and motion plots of Validation Video #9: (a) The quality change plot (b) The motion change plot.

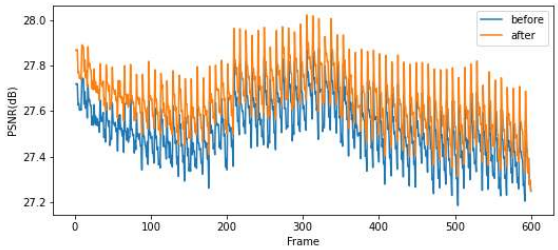


(a)

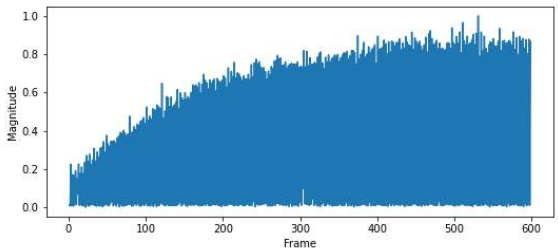


(b)

Figure A.8: Quality and motion plots of Validation Video #11: (a) The quality change plot (b) The motion change plot.

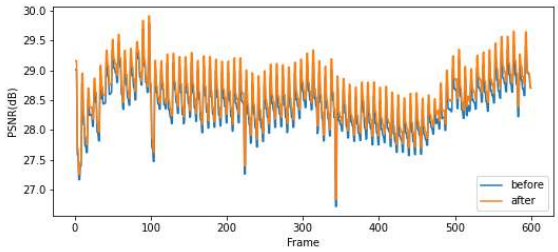


(a)

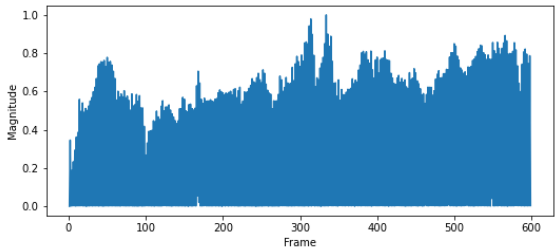


(b)

Figure A.9: Quality and motion plots of Validation Video #12: (a) The quality change plot (b) The motion change plot.

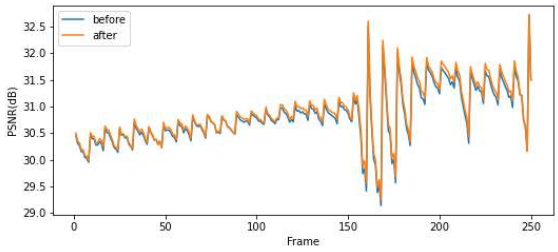


(a)

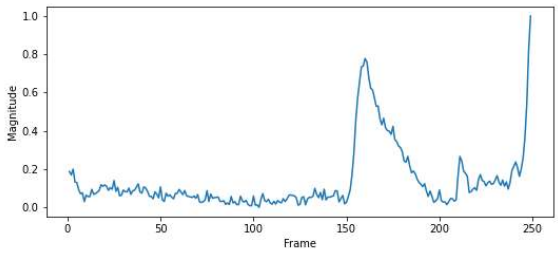


(b)

Figure A.10: Quality and motion plots of Validation Video #15: (a) The quality change plot (b) The motion change plot.

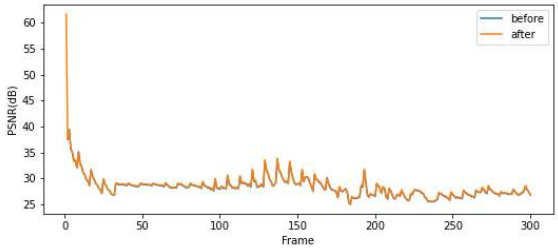


(a)

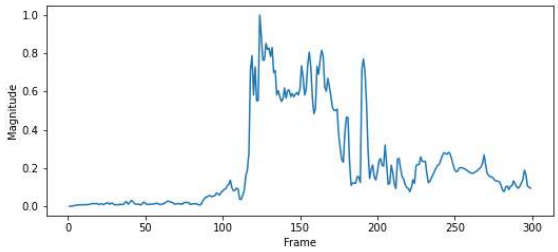


(b)

Figure A.11: Quality and motion plots of Test Video #1: (a) The quality change plot (b) The motion change plot.

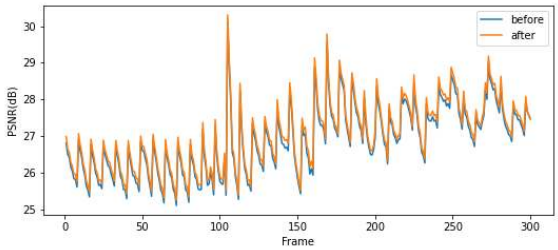


(a)

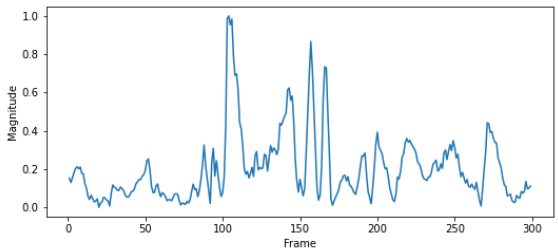


(b)

Figure A.12: Quality and motion plots of Test Video #2: (a) The quality change plot (b) The motion change plot.

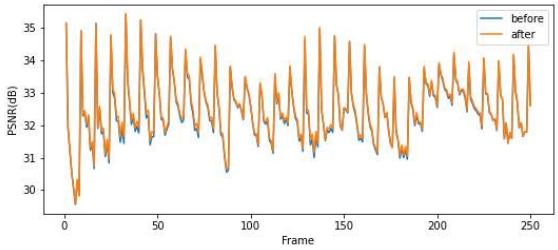


(a)

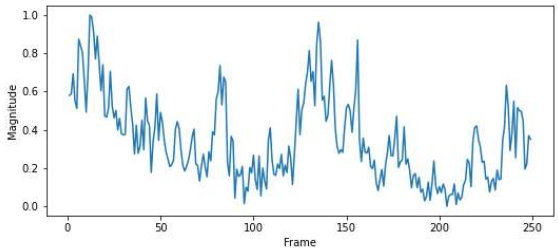


(b)

Figure A.13: Quality and motion plots of Test Video #3: (a) The quality change plot (b) The motion change plot.

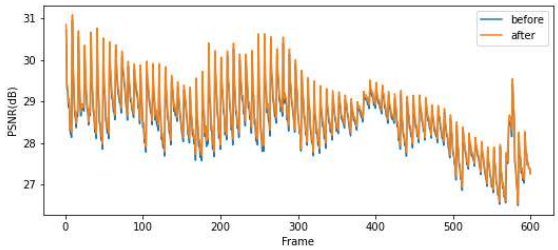


(a)

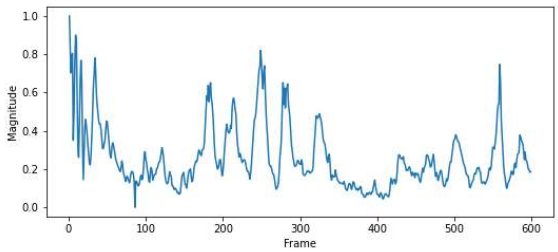


(b)

Figure A.14: Quality and motion plots of Test Video #4: (a) The quality change plot (b) The motion change plot.

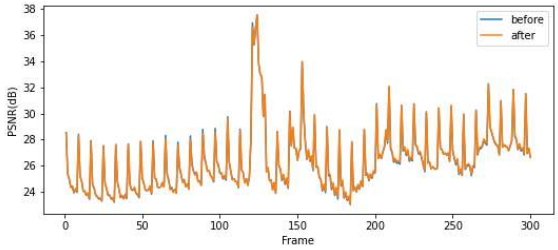


(a)

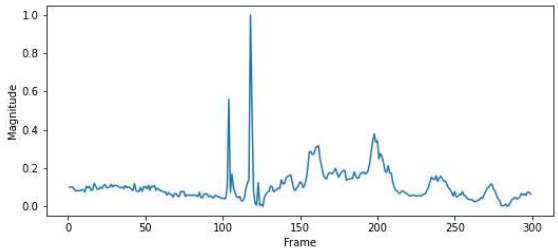


(b)

Figure A.15: Quality and motion plots of Test Video #5: (a) The quality change plot (b) The motion change plot.

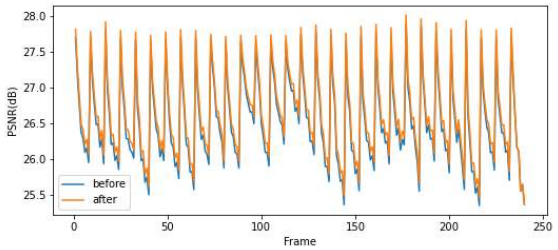


(a)

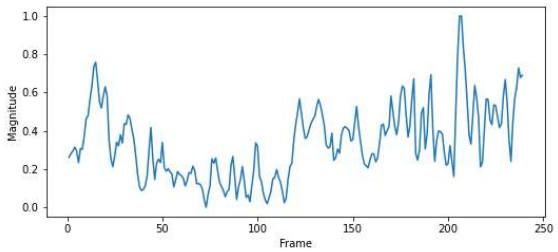


(b)

Figure A.16: Quality and motion plots of Test Video #6: (a) The quality change plot (b) The motion change plot.

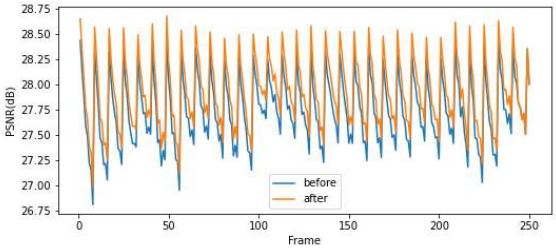


(a)

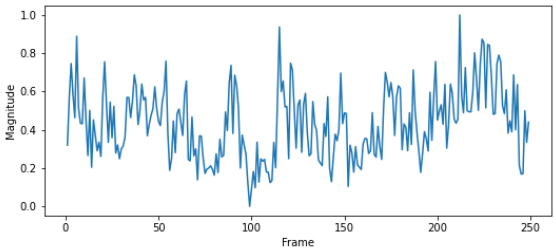


(b)

Figure A.17: Quality and motion plots of Test Video #7: (a) The quality change plot (b) The motion change plot.

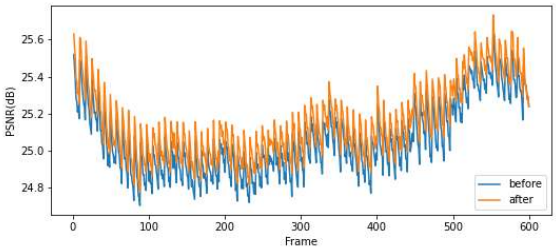


(a)

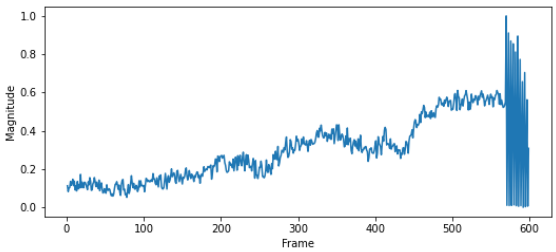


(b)

Figure A.18: Quality and motion plots of Test Video #9: (a) The quality change plot (b) The motion change plot.

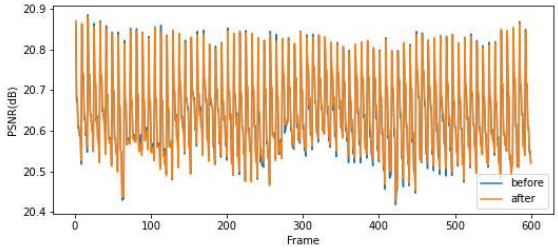


(a)

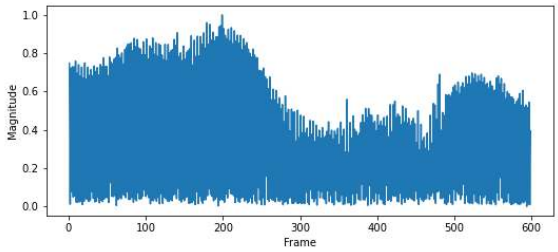


(b)

Figure A.19: Quality and motion plots of Test Video #11: (a) The quality change plot (b) The motion change plot.

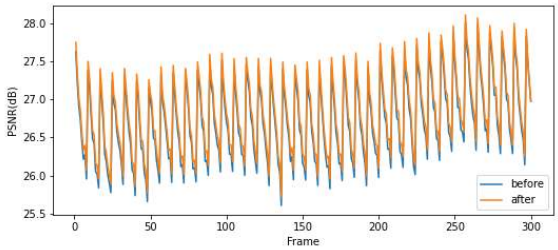


(a)

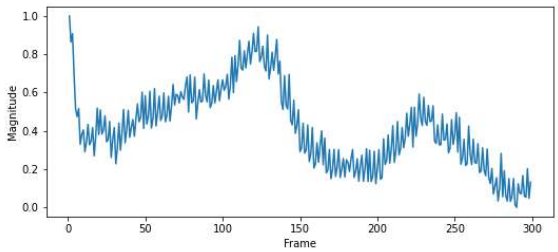


(b)

Figure A.20: Quality and motion plots of Test Video #12: (a) The quality change plot (b) The motion change plot.

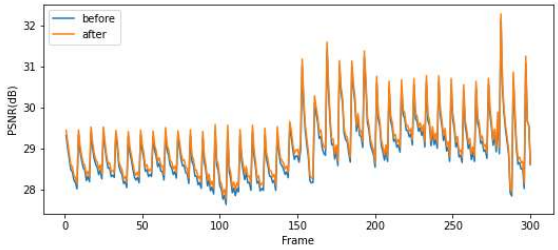


(a)

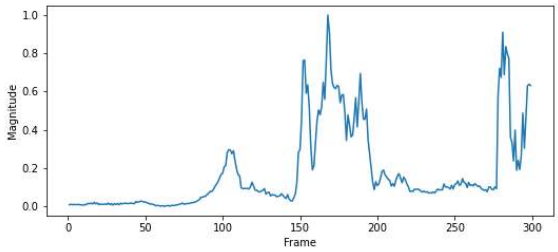


(b)

Figure A.21: Quality and motion plots of Test Video #13: (a) The quality change plot (b) The motion change plot.

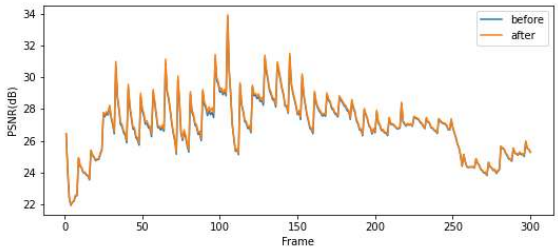


(a)

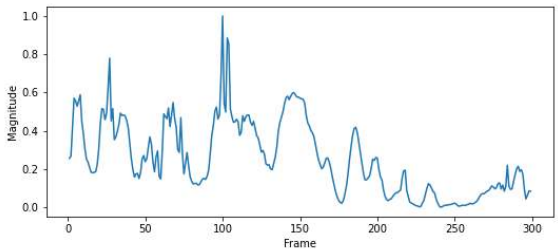


(b)

Figure A.22: Quality and motion plots of Test Video #14: (a) The quality change plot (b) The motion change plot.



(a)



(b)

Figure A.23: Quality and motion plots of Test Video #15: (a) The quality change plot (b) The motion change plot.

Appendix B

QUALITATIVE EXPERIMENTAL RESULTS

(a)

(b)

(c)

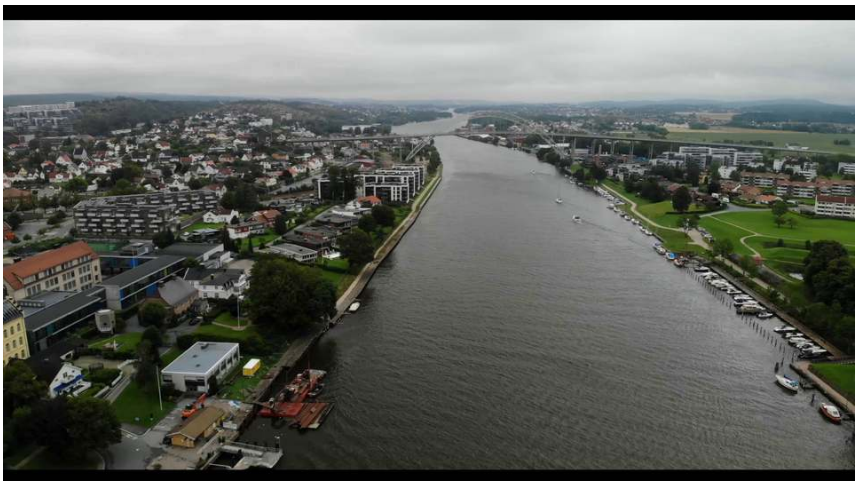
Figure B.1: Quality comparison of decompressed and post-processed images: (a) The ground truth image, (b) the decompressed image by VVC, $qp = 40$, (c) the post-processed image by the network.

(a)

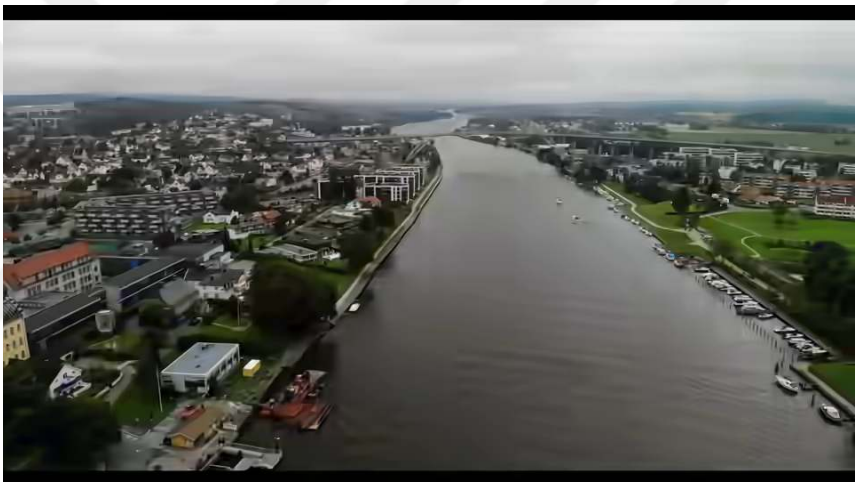
(b)

(c)

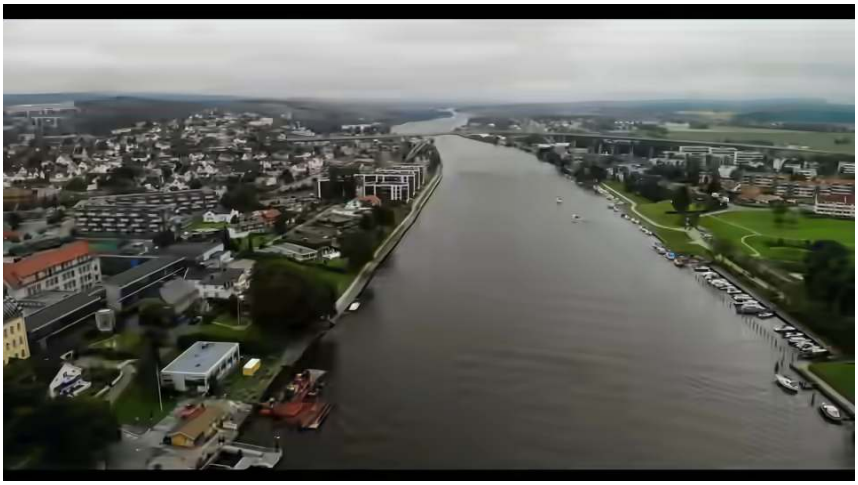
Figure B.2: Quality comparison of decompressed and post-processed images: (a) The ground truth image, (b) the decompressed image by VVC, $qp = 40$, (c) the post-processed image by the network.



(a)



(b)



(c)

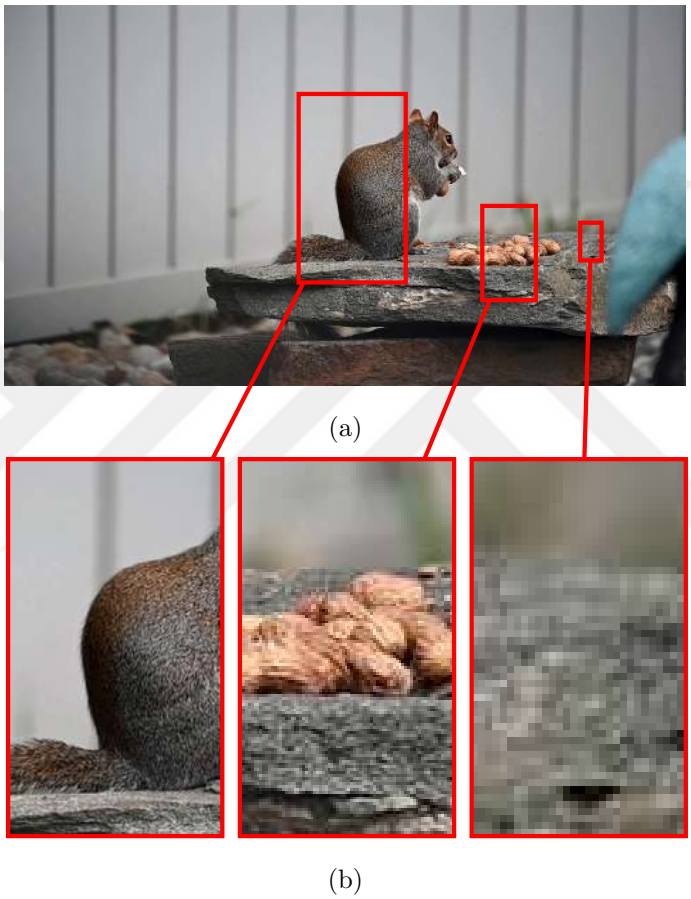
Figure B.3: Quality comparison of decompressed and post-processed images: (a) The ground truth image, (b) the decompressed image by VVC, $qp = 40$, (c) the post-processed image by the network.

(a)

(b)

(c)

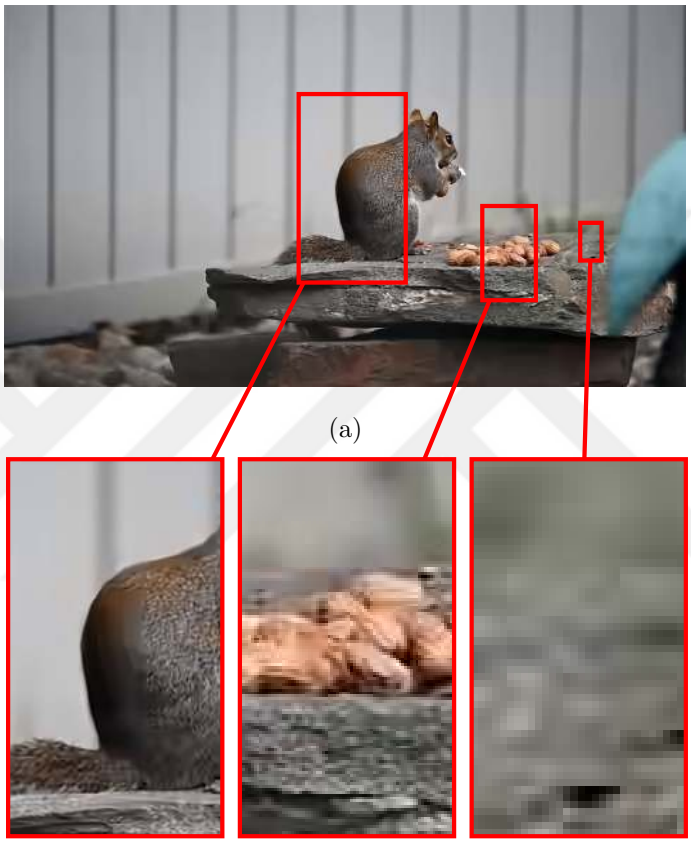
Figure B.4: Quality comparison of decompressed and post-processed images: (a) The ground truth image, (b) the decompressed image by VVC, $qp = 40$, (c) the post-processed image by the network.



(a)

(b)

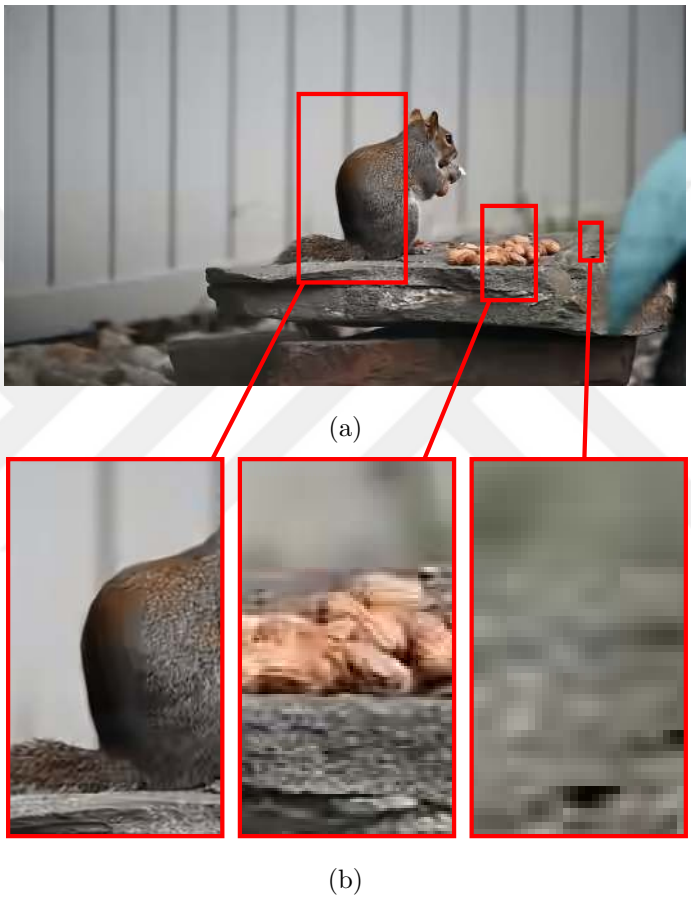
Figure B.5: Quality comparison of decompressed and post-processed images: (a) The ground truth image. (b) Its zoomed version.



(a)

(b)

Figure B.6: Quality comparison of decompressed and post-processed images: (a) The decompressed image by VVC, $qp = 40$. PSNR = 31.48 dB. (b) Its zoomed version.



(a)

(b)

Figure B.7: Quality comparison of decompressed and post-processed images: (a) The post-processed image by the network. PSNR = 31.51 dB. (b) Its zoomed version.