Perceptually Motivated Algorithms for Multimedia

by

Shupei Zhang

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Perceptual factors in vision can facilitate the development of more effective multimedia algorithms. In particular, the wide dynamic range of the human vision system is a motivation for developing image lighting enhancement algorithms. Image lighting enhancement can be achieved by capturing multiple images with different exposure settings and then reconstructing a final image. However, this approach cannot solve the problem of revealing or predicting details in already-captured images. Single-image lighting enhancement is desirable for this scenario, but many challenges remain to be addressed including over-enhancement, noise, and color artifacts due to a lack of understanding of the image content. Another aspect of multimedia algorithms that can benefit from perceptual factors, like the foveation mechanism and perceptual quality, is image and video compression. As the resolution and image quality of modern cameras have increased, the amount of data produced by computational photography has also surged dramatically. This has created a demand for better image/video compression methods that can reduce the data size without compromising the image quality.

In this thesis, four perceptually motivated methods are proposed to address the challenges in single-image lighting enhancement and image/video compression. First, we propose an image lighting enhancement method based on a fusion pyramid, which is a traditional contrast-based fusion approach. Second, we propose a self-attention-based learning strategy to reconstruct a properly exposed image from a single input image. We leverage the self-attention mech-

anism to model the interdependencies between different locations, and design a generative adversarial network (GAN) with a custom HDR loss function to improve the image quality. Third, we propose a novel video compression method that integrates visual saliency information with foveation to reduce perceptual redundancy. This is an innovative approach to subsample and restore the input image using saliency data, which allocates more space for salient regions and less for non-salient ones. Finally, based on the assumption that a group of images can be decomposed into several shared feature matrices, we propose a novel principal component approximation network (PCANet) for image compression. This is the first learning-based method that achieves promising performance while including the size of the network in the bitrate calculation.

Preface

The majority of this thesis has been published or currently under review in peer-reviewed journals or conferences. Chapter 3 of this thesis describes a traditional contrast-based fusion approach for image lighting enhancement, as published in the 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). Chapter 4 describes a high dynamic range (HDR) reconstruction inspired self-attention-based learning strategy to reconstruct a properly exposed image from a single input image, as published in the International Conference on Smart Multimedia (ICSM). Chapter 5 introduces a novel video compression method that integrates visual saliency information with foveation to reduce perceptual redundancy, as published in IEEE Access. Chapter 6 presents a novel principal component approximation network (PCANet) to decompose images for compression, which was published in ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM).

I have written this thesis in first person plural to acknowledge and honor the contribution of my advisors and collaborators.

Related Publications:

- **Shupei Zhang**, Charles Euler, and Anup Basu. "Image dynamic range enhancement based on fusion pyramid," *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1-4. IEEE, 2020.

- **Shupei Zhang**, Kangkang Hu, Zhenkun Zhou, and Anup Basu. "Lighting Enhancement Using Self-attention Guided HDR Reconstruction." *International Conference on Smart Multimedia*, pp. 409-418. Cham:

Springer International Publishing, 2022.

- **Shupei Zhang** and Anup Basu, "Visual Saliency Guided Foveated Video Compression," *IEEE Access*, vol. 11, pp. 62535-62548, 2023.

- **Shupei Zhang**, Chenqiu Zhao, and Anup Basu. "Principal Component Approximation Network for Image Compression," *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 5, Article 121 (May 2024).

- Chenqiu Zhao, Guanfang Dong, **Shupei Zhang**, Zijie Tan, Anup Basu, "Frequency Regularization: Reducing Information Redundancy in Convolutional Neural Networks," *IEEE Access*, vol. 11, pp. 62535-62548, 2023.

Ever tried. Ever failed. No matter. Try Again. Fail again. Fail better.

– Samuel Beckett

Acknowledgements

I would like to thank my supervisor, Professor Anup Basu for his support, motivation and enthusiasm, which enriched my Ph.D. journey. I would also like to thank the rest of my thesis committee, Prof. Irene Cheng, Prof. Bruce Cockburn, Prof. Ehab Elmallah, and Prof. Nadège Thirion-Moreau, for all the great feedback they gave on my research and thesis.

This challenging journey would not have been possible without all my friends and labmates. I am grateful for all their support and friendship. My heartfelt thanks to my parents and family for their unconditional love, continuous encouragement, and endless support.

Finally, special thanks to Logan, for standing by me through thick and thin.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 2
- 1.2 Weakness of Existing Approaches . . . 5
  - 1.2.1 Image Lighting Enhancement . . . 5
  - 1.2.2 Foveated Video Compression . . . 5
  - 1.2.3 Machine-Learning-Based Image Compression . . . 6
- 1.3 Our Contribution . . . 6
  - 1.3.1 Image Dynamic Range Enhancement Based on Fusion Pyramid . . . 6
  - 1.3.2 Lighting Enhancement using Self-Attention Guided HDR Reconstruction . . . 7
  - 1.3.3 Foveated Video Compression . . . 7
  - 1.3.4 Machine Learning Based Image Compression . . . 7
- 1.4 Thesis Organization . . . 8

2 Related Work **9**
- 2.1 Image Dynamic Range Enhancement Based on Fusion Pyramid 9
- 2.2 Lighting Enhancement Using Self-Attention Guided HDR Reconstruction . . . 11
  - 2.2.1 Traditional Methods . . . 11
  - 2.2.2 Deep Learning Approaches . . . 12
  - 2.2.3 Image Quality Metrics . . . 12
- 2.3 Visual Saliency Guided Foveated Video Compression . . . 13
  - 2.3.1 Visual Saliency . . . 13
  - 2.3.2 Foveated Compression . . . 14
- 2.4 Principal Component Approximation Network for Image Compression . . . 15

3 Image Dynamic Range Enhancement Based On Fusion Pyramid **18**
- 3.1 Introduction . . . 18
- 3.2 Proposed Method . . . 19
  - 3.2.1 Image Enhancement . . . 19
- 3.3 Experiments . . . 21
- 3.4 Conclusion . . . 23

4 Lighting Enhancement using Self-Attention Guided HDR Reconstruction **24**
- 4.1 Introduction . . . 24
- 4.2 Proposed Method . . . 25
  - 4.2.1 Generator . . . 25
  - 4.2.2 Self-attention . . . 26
  - 4.2.3 Discriminator . . . 28

4.2.4 Custom Loss Function 29
4.3 Experiments 31
4.3.1 Datasets 31
4.3.2 Evaluation Metrics 32
4.3.3 Ablation Test 33
4.3.4 Objective Image Quality Comparison 34
4.4 Conclusion 38

5 Visual Saliency Guided Foveated Video Compression 40
5.1 Introduction 40
5.2 Proposed Method 42
5.2.1 Overview 42
5.2.2 Saliency Encoding 42
5.2.3 Foveation using Image Warping 43
5.2.4 Salient Area Scaling 48
5.2.5 Effectiveness of Foveation in Reducing Redundancy . . 50
5.3 Experiments and Discussion 51
5.3.1 Subjective Image Quality Assessment 54
5.3.2 Objective Image Quality Assessment 55
5.3.3 Impact of Saliency Prediction Accuracy 57
5.3.4 Applications and Limitations 59
5.4 Conclusion 60

6 Principal Component Approximation Network for Image Compression 67
6.1 Introduction 67
6.2 Principal Component Approximation Network 71
6.3 Experiments 78
6.3.1 Evaluation of the proposed PCANet 80
6.3.2 Discussion 87
6.4 Conclusion 90

7 Conclusion and Future Work 91

References 93

List of Tables

3.1 NIQE score comparison. 23

4.1 Model settings. 34
4.2 Model performance. 34
4.3 Image Quality Comparison. Scores in red indicate the best performance, scores in blue indicate the second-best performance. 36

5.1 Average and total entropy of a frame in the original and warped videos. 52
5.2 Overall average and total information entropy of the original and warped images. 52
5.3 Subjective test results. 55
5.4 BD-EWPSNR, BD-VMAF, and BD-LPIPS scores, as well as the corresponding BD-rate values. For BD-EWPSNR and BD-VMAF, a positive value x indicates that the proposed method can increase the performance by x at the same bitrate. For BD-LPIPS, a negative value $-x$ indicates that the proposed method can increase the performance by $-x$ at the same bitrate. For BD-rate, a negative value $-x$ indicates that the proposed method can achieve the same level of performance with a bitrate saving of $x\%$. 56

6.1 Evaluation of the proposed approach on 20,480 images extracted from the COCO dataset. 83
6.2 Evaluation of the proposed approach on 20,480 images extracted from the CelebA dataset. 85
6.3 Evaluation of the proposed approach on images from video "boats." 87
6.4 BD-PSNR and the corresponding BD-rate values comparing the proposed method and TPAMI2021. For BD-PSNR, a positive value x indicates that the proposed method can increase the performance by x at the same bitrate. For BD-rate, a negative value $-x$ indicates that the proposed method can achieve the same level of performance with a bitrate saving of $x\%$. 88

List of Figures

1.1 Two-dimensional cosine patterns used in JPEG. 4

3.1 Overview of image enhancement process. 19
3.2 Performance comparison. 22

4.1 Generator structure. 26
4.2 Structure of the attention module. 27
4.3 Ablation test image examples. 35
4.4 Reconstruction details of different methods. 35
4.5 Example output of various methods. 37

5.1 Quads and quad vertices. 45
5.2 Salient area scaling. 49
5.3 Transformed meshes under different constraints: a) original mesh, b) transformed mesh without constraints on non-salient quads, c) transformed mesh with smoothing weight scheme, d) transformed mesh with smoothing weight scheme and uniform constraint. 50
5.4 Test pipeline. 52
5.5 Overall rate-distortion curves of the proposed method compared with H.264 and H.265. We conducted two sets of comparisons. The results in the first row shows the comparison between the original H.264 and H.264 incorporating the proposed method. The results in the second row shows the comparison between the original H.265 and H.265 incorporating the proposed method. Three metrics are used for each set of comparison: EWPSNR, VMAF, and LPIPS. For EWPSNR and VMAF, higher is better. For LPIPS, lower is better. 53
5.6 Rate-EWPSNR curves of the proposed method compared with H.264 and H.265 on different categories. The first two rows show the comparison with H.264, and the last two rows show the comparison with H.265. Higher is better. 61
5.7 Rate-LPIPS curves of the proposed method compared with H.264 and H.265 on different categories. The first two rows show the comparison with H.264, and the last two rows show the comparison with H.265. Lower is better. 62
5.8 Rate-VMAF curves of the proposed method compared with H.264 and H.265 on different categories. The first two rows show the comparison with H.264, and the last two rows show the comparison with H.265. Higher is better. 63
5.9 Video quality comparison with H.264. The first and third rows show results produced by the proposed method. The second and fourth rows show results produced by H.264. The results are best viewed in color. 64

5.10 Video quality comparison with H.265. The first and third rows show results produced by the proposed method. The second and fourth rows show results produced by H.265. The results are best viewed in color. 65
5.11 The same 8×8 image block from the original video (left) and the video compressed by the proposed method (right). The pixels in the salient regions are shifted up by about one pixel from their original location. 65
5.12 EWPSNR scores of the reconstructed images with different saliency prediction accuracy and saliency threshold s_t settings. 66
5.13 The warped meshes for different saliency threshold (s_t) settings are shown. The top left image shows the saliency map with a NSS score of 8.3277. The top right image shows the mesh with $s_t = 1$. The bottom left image shows the mesh with $s_t = 50$. The bottom right image shows the mesh with $s_t = 100$. 66

6.1 Illustration of the proposed principal component approximation network. An image is reconstructed by the weighted sum of several shared feature matrices, and the weight vector can be used as the coding vector for image compression. 68
6.2 The main architecture of the proposed PCANet focusing on learning a series of shared feature matrices for image decomposition. 70
6.3 Detailed architecture of the proposed PCANet. This example shows the size of intermediate tensor shapes for four different paths when the input image shape is 178. Results from different paths are up-sampled and summed to produce the final reconstruction result. 76
6.4 The comparisons between the proposed PCANet and state-of-the methods on Kodak dataset. 80
6.5 Illustration of training images extracted from the COCO dataset. 82
6.6 Sample images from the CelebA dataset. 84
6.7 Illustration of sample frames and differences between consecutive frames from the video "boats." 86
6.8 Comparison between the proposed approach and state-of-the-art methods on the boats video. 87
6.9 Demonstration of unseen images reconstructed by the proposed PCANet, with a PSNR of 26.43, and a bpp of 0.158. 89

Glossary

Advanced Video Coding (AVC/H.264) A video compression standard based on block-oriented, motion-compensated coding.

AOMedia Video 1 (AV1) An open, royalty-free video coding format initially designed for video transmissions over the Internet.

Asynchronous Transfer Mode (ATM) A telecommunications standard for digital transmission of multiple types of traffic.

bits per pixel (bpp) The number of bits used to encode the gray-scale value or color value of a pixel. It is used to measure the efficiency of compression algorithms.

Confidence Interval (CI) A range of estimates for an unknown parameter at a certain confidence level.

Convolutional Neural Network (CNN) A type of neural network characterized by shared weights and translation invariance, usually applied in image analysis.

Delentropy A method to measure the entropy of an image.

Discrete Sine Transform (DST) A Fourier-related transform similar to the discrete Fourier transform, but using a purely real matrix.

Discrete Cosine Transform (DCT) A mathematical transform related to the Fourier transform and widely used in digital data compression.

Double-Stimulus Impairment Scale (DSIS) A subjective video quality assessment method. The viewers see an unimpaired reference video, then the same video impaired, and after that they are asked to vote on the second video using a impairment scale.

Full-Reference Image Quality Assessment (FR-IQA) A class of algorithms that aims to automatically assess the quality of images by comparing the original image with the distorted image.

Generative Adversarial Network (GAN) A class of machine learning framework.

HDR Image GRADient-based Evaluator (HIGRADE) A image quality metric for evaluating image quality of tone-mapped HDR pictures.

High Dynamic Range (HDR) Image capturing/processing/display technologies which achieve higher range of luminosity using more than 8 bits per pixel.

High Efficiency Video Coding (HEVC/H.265) A video compression standard designed as part of the MPEG-H project as a successor to the widely used Advanced Video Coding.

Human Visual System (HVS) The human visual system comprises the sensory organ and parts of the central nervous system, which gives humans the sense of vision.

Image Quality Assessment (IQA) A class of algorithms that aims to automatically assess the quality of images.

Joint Photographic Experts Group (JPEG) A lossy still image compression standard developed and maintained by an international committee having the same name.

JPEG2000 An image compression standard and coding system.

Low Dynamic Range (LDR) Traditional image capturing/processing/display technologies which produce lower range of luminosity than HDR using 8 bits per pixel.

Moving Picture Experts Group (MPEG) A digital video standard for compression of full-motion images.

Natural Scene Statistic (NSS) The statistical regularities related to natural scenes.

Natural Image Quality Evaluator (NIQE) A no-reference image quality assessment algorithm based on natural scene statistics.

No-Reference Image Quality Assessment (NR-IQA) A class of algorithms that aims to automatically assess the quality of images without any reference image.

Peak Signal to Noise Ratio (PSNR) The ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.

Quantization Parameter (QP) An index used to derive a scaling matrix to control quantization. It regulates how much spatial detail is saved.

Region of Interest (ROI) A sample within an image identified for a particular purpose.

RGB An additive color model in which the red, green and blue primary colors of light are added together in various ways to reproduce a broad array of colors.

Self-attention A mechanism to capture dependencies and relationships within input data.

Singular Value Decomposition (SVD) A matrix factorization method that decomposes a matrix into three matrices.

Standard Dynamic Range (SDR) An acronym for LDR.

Structural Similarity (SSIM) A method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos.

Variable Resolution (VR) A method for compressing video sequences by encoding different parts of the video at different resolutions.

VP8 An open and royalty-free video compression format released by On2 Technologies in 2008.

VP9 An open and royalty-free video coding format developed by Google.

YUV A color model that describes a color as a Y component (luma) and two chroma components U and V.

Chapter 1

Introduction

Perceptual factors in vision, such as brightness, color, contrast, movement, visual saliency, foveation, and etc., can contribute to the improvement of multimedia algorithms. One category of multimedia algorithms that can benefit from perceptual factors is single-image lighting enhancement. Single-image lighting enhancement is the process of correcting or improving the lighting condition of an image based on information contained in the image itself. Depending on the application area, it can be used to reveal more details or to improve the perceptual quality of the image to the human observer. While single-image lighting enhancement has been explored by researchers for many years, there are still many challenges to be addressed, including over-enhancement, noise, and color artifacts due to a lack of understading of the image content. Chapters 3 and 4 of this thesis explore methods to address these challenges and improve single-image lighting algorithms using computer vision and machine learning techniques.

Another category of multimedia algorithms that can benefit from perceptual factors is image/video compression. Image/video compression aims to reduce the size of data while minimizing the loss of perceptual quality. Although many image/video compression algorithms have been proposed and widely applied in practice, there are still some gaps in utilizing perceptual factors to boost the performance of compression algorithms. Chapters 5 and 6 of this thesis explore methods to improve the efficiency of image/video compression by incorporating visual saliency information and utilizing a decomposition

method that models the data better.

1.1 Motivation

Computational photography refers to the techniques of capturing and processing images with digital methods instead of optical processes. It has been increasingly popular in recent years due to the rapid development of computing capabilities and the limited space on mobile devices for optical improvements. One of the most important applications of computational photography is High Dynamic Range (HDR) imaging, which produces images that are similar to what the human visual system perceives.

The dynamic range of a signal is the ratio between the largest and smallest possible values. In photography, the dynamic range of a scene is the ratio between the brightest and darkest light intensities in the scene. Natural scenes can have a luminance level ranging from $10^7 cd/m^2$ in direct sunlight to $10^{-1} cd/m^2$ at night [84]. This corresponds to more than 8 orders of magnitude. Human eyes are able to adapt to this high dynamic range thanks to the ability to change the pupil size and the presence of two types of photoreceptor cells in the retina: cones and rods. Cones are more sensitive in bright light and are responsible for color vision, while rods are more sensitive in dim light. Therefore humans naturally have a high dynamic range in visual perception and can see details in both bright and dark areas of a scene. Consequently, high dynamic range images are more favorable compared to standard dynamic range (SDR) images because they contain more details and are more similar to what we see [70]. However, very few imaging systems are capable of directly capturing such a high dynamic range [47]. Thus, HDR imaging is usually achieved by bracketing, which is to capture multiple images with different exposure settings and then fuse them into one HDR image. This approach requires extra computation for image alignment and fusion, and is not suitable for enhancing images that have already been taken. Therefore, image lighting enhancement algorithms need to be developed that reveal the details in both bright and dark areas of an existing image. We propose two image

lighting enhancement algorithms in this thesis. The first one, "Image Dynamic Range Enhancement Based on Fusion Pyramid," is a traditional approach that uses a contrast-based fusion pyramid to enhance the image. The second one, "Lighting Enhancement Using Self-Attention Guided HDR Reconstruction," is a self-attention-based learning strategy to reconstruct a properly exposed image from a single input image.

Another important application of computational photography is image/video compression. Computational photography can be used to identify regions of interest in an image or video, which enables the imaging system to prioritize the image quality of the identified regions. Image/video compression benefits from this because of the spatially-varying sensing characteristics of the Human Visual System (HVS) [109].

The two types of photoreceptor cells are distributed differently in the retina. Cones have the highest density in the fovea while rods are almost absent in the fovea and reach their highest density within a 10 to 20 degree periphery of the fovea. Their sensitivity to light also varies. Cones, which are responsible for photopic vision, have a lower sensitivity than rods, which are responsible for scotopic vision. Despite having higher sensitivity, rods have extremely poor visual acuity under low luminance compared to cones under photopic conditions. Since we use our photopic vision most of the time except for in very dark environments, the image quality we perceive is mostly determined by the cones. Cones have the highest density in the center of fovea, so the quality of the part of the image that gets projected to this area has the biggest impact on the perceived image quality. We call the parts of the image in the center of the receptive field the "fixation points." On the other hand, the quality of the image in the periphery has less impact on the perceived image quality. Therefore, compression can be achieved by reducing the image quality in the periphery while maintaining the quality of the fixation points because humans are less sensitive to the peripheral image quality. Foveated compression is a compression technique that takes advantage of this property of the HVS. In this thesis we propose a novel foveated video compression method based on visual saliency and feature preserving image warping.

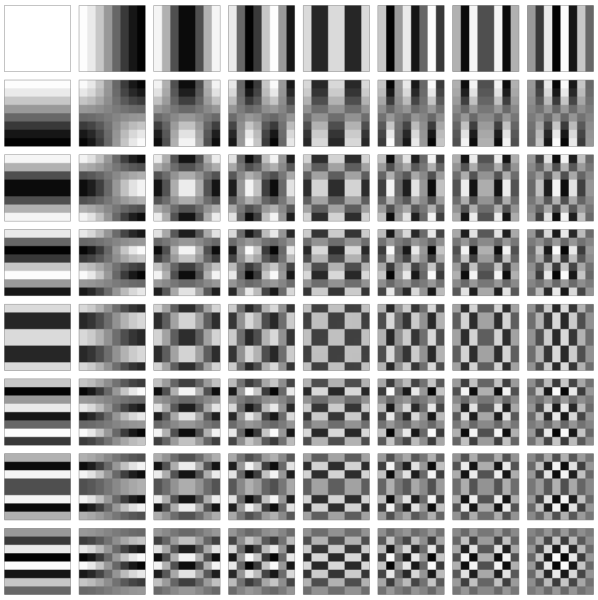

Figure 1.1: Two-dimensional cosine patterns used in JPEG.

Image compression can also be achieved by decomposing images into a set of feature patterns and their corresponding coefficients. This has been widely applied in image compression methods such as JEPG, JPEG2000, and HEIC [49]. The discrete cosine transform (DCT) is the most commonly used transform for decomposing images into 2-dimensional cosine patterns of different frequencies. Figure 1.1 shows the 2-dimensional cosine patterns of a 8×8 block used in JPEG. Images can be decomposed into a set of fixed cosine patterns and their corresponding coefficients using the DCT, and then compressed by scaling and quantizing the coefficients. In this process, more information from the high frequency components are discarded compared to the low frequency components. This is because humans are not equally sensitive to all spatial frequency patterns [109] and the high frequency components are less important for image quality.

Despite being effective in general image compression, the DCT might not be optimal for domain specific images. For example, face images contains many common features such as hair, eyes, nose, and mouth. These features are usually similar among different images and located in similar positions. Therefore, for domain specific images, it might be more effective to decom-

pose images into a set of domain specific features instead of using generic cosine patterns. In this thesis, we adopt this idea and propose a novel principal component approximation network that decomposes images into a set a learnable features.

1.2 Weakness of Existing Approaches

1.2.1 Image Lighting Enhancement

Some common weaknesses of existing image lighting enhancement algorithms are:

- Some methods require multiple images taken at different exposures to enhance the details, such as daytime and nighttime surveillance video frames of the same place [91], [124]. This requirement is usually hard to meet in practice for existing images and videos.
- Some approaches might over enhance the image and produce unnatural images, for example, overly bright nighttime images that look like daytime images, or images with severe color shift [32], [40], [52], [55], [77], [111], [115]. Some noise might also be introduced in the process.
- Some methods might introduce color artifacts in the enhanced image due to a lack of understanding of the image content.

1.2.2 Foveated Video Compression

Some common weaknesses of existing foveated video compression algorithms are:

- Some methods are limited by the existing block-based video encoders they improve upon, and do not fully utilize the HVS's spatially-varying sensing characteristics [33], [48], [98], [117], [128], [129]. This might result in a suboptimal compression performance.

- Existing variable resolution (VR) approaches enlarge the salient areas, which leaves less space for peripheral pixels and decreases the peripheral image quality [15], [20]–[22], [116].

- Multiple fixation points are not handled coherently and effectively.

1.2.3 Machine-Learning-Based Image Compression

Some common weaknesses of existing machine-learning-based image compression algorithms are:

- Existing machine learning based methods usually consider their models as general models that work on all images [2]–[4], [10]–[12], [18], [23], [24], [36], [38], [39], [43], [44], [53], [54], [62], [63], [65], [68], [79], [80], [82], [86], [95], [96], [103], [106], [107], [110], [114], [120], [121], [127], [130], [132]. Thus, they do not include the model size when calculating the bitrate. However, some evidence implies that memorization still plays an important role in the performance of deep neural networks, which reduces their ability to generalize.

- Existing machine learning models are usually large, which suggests significant redundancy in the models themselves.

1.3 Our Contribution

1.3.1 Image Dynamic Range Enhancement Based on Fusion Pyramid

- We propose a novel segmentation-fusion approach, based on an image pyramid, to produce natural images from single exposures.

- An enhancement method is designed, based on the Retinex model, to improve the lighting condition individually for different parts of the image [40].

- The enhancement is guided by HDR inspired quality metrics, which ensures the enhanced image quality.

1.3.2 Lighting Enhancement using Self-Attention Guided HDR Reconstruction

- This is the first work to utilize the self-attention mechanism to model long-distance dependencies across different regions in images for lighting enhancement. This mechanism helps reduce the artifacts and boosts the output image quality.

- We design a new HDR loss function inspired by the characteristics of HDR images and the HDR reconstruction process. We show that this loss function can alleviate the color shift/artifacts in the output images.

- We compare our work with several state-of-the-art methods utilizing objective tests to show that our proposed method outperforms all other existing methods.

1.3.3 Foveated Video Compression

- A foveation process, based on per-quad image warping, is used to preserve the image quality of salient regions, achieving non-uniform subsampling based on saliency level.

- The saliency data is incorporated at a lower granularity, providing more precise quality control of salient regions.

- Our method is independent of traditional encoding processes, making it applicable to improve most existing compression methods.

1.3.4 Machine Learning Based Image Compression

1. We propose the principal component approximation network to learn shared feature matrices for image compression. The network parameters are used to approximate these shared matrices, thereby reducing information redundancy inside the proposed network. Therefore, the proposed approach achieves promising compression results even after taking the size of network parameters into account.

2. The size of the proposed network is relatively small, containing only around 4 million trainable parameters. The architecture is very straightforward and explainable.

3. Comprehensive experiments based on several standard datasets demonstrate the effectiveness of the proposed approach.

4. A new metric is proposed to evaluate the information redundancy inside the models.

1.4 Thesis Organization

The rest of the thesis is organized as follows: In Chapter 2 we discuss existing methods for image lighting enhancement, foveated video compression, and machine-learning-based image compression in detail. In Chapter 3, we present a traditional image lighting enhancement method based on a fusion pyramid. In Chapter 4, we introduce a lighting enhancement method using self-attention guided HDR reconstruction. In Chapter 5, we propose a foveated video compression method based on visual saliency and feature preserving image warping. In Chapter 6, we propose a principal component approximation network for image decomposition and compression. Finally, we conclude the thesis in Chapter 7.

Chapter 2

Related Work

This chapter reviews related publications to this thesis. Section 2.1 and Section 2.2 both discuss image lighting enhancement methods. Section 2.1 has a broader scope and covers general image lighting enhancement methods while Section 2.2 focuses on single image lighting enhancement methods. Section 2.3 introduces visual saliency and reviews foveated video compression methods based on visual saliency. Finally, Section 2.4 reviews image compression methods based on deep learning.

2.1 Image Dynamic Range Enhancement Based on Fusion Pyramid

Rao et al. overview various video enhancement processing and analysis algorithms. The authors categorize video enhancement methods into two types: self-enhancement and context-based fusion enhancement. Self-enhancement methods, including contrast enhancement, HDR-based enhancement, wavelet-based enhancement and compression-based enhancement, are all based on the information in the current image itself; while context-based enhancement utilizes illumination information in multiple frames [90].

Contrast enhancement techniques include histogram equalization and tone mapping. There are many publications related to the former one, which can be divided into global and local methods [1], [45]. Zhang et al. use per-frame multi-exposure and best exposed region detection to produce frames for later fusion and achieve better contrast [124]. Didyk et al. classify regions in a

frame into three categories, then enhance the contrast of the regions based on their categories [28]. Rafael et al. present an enhancement method for reverse tone mapping based on a bilateral filter [56]. Francesco et al. also take the reverse tone mapping approach to enhance LDR videos. Lu and Jian adopt a zone-based exposure analysis and use high-level features of the image to set priority of the zones. Then, a best non-linear curve mapping, which is the result of a global optimization, is applied to the whole image.

The compression-based video enhancement process happens in the decompression phase of a video. The basic idea of this method is to enhance the image by manipulating the DCT coefficients [64]. It has several advantages including low computational complexity, less severe block artifacts compared with post-decompression methods and it is applicable to any DCT-based image compression method [105]. Wavelet-based methods are mostly used in de-noising and image feature preservation [30].

The second type of video enhancement, context-based enhancement, is accomplished by extracting and fusing meaningful information in a video sequence. One of the approaches is enhancing videos by exploiting the context in both night-time and day-time videos [91]. But, this approach is generally limited to full day surveillance videos captured by fixed cameras [124]. Frames in a video may contain images of the same scene with different exposures, making it possible to enhance the dynamic range [131]. Yu et al. apply a zone system to input videos for exposure evaluation and then remap each region using several different tone mapping curves to achieve better contrast. They also use information in frames to maintain temporal consistency [126]. Henrik et al. apply adaptive spatio-temporal smoothing to low illumination videos to reduce noise and enhance contrast [75].

Several researchers have used deep learning to enhance images. Marnerides et al. use three CNN branches to process local, large pixel neighborhood and global information, respectively [77]. Gabriel et al. propose a UNet-like CNN for HDR reconstruction [32]. They train their CNN on a simulated HDR dataset created from a subset of MIT Places dataset to make the model more generalizable. Yifan et al. propose EnlightenGAN, which does not require

paired HDR-LDR images for training [52]. EnlightenGAN also utilizes adversarial learning. However, a downside of deep learning approaches is that they require significant computational resources, and can be slow if executed on a CPU.

2.2 Lighting Enhancement Using Self-Attention Guided HDR Reconstruction

Producing images without losing details in scenes with extreme contrast can be achieved by taking a series of images at different exposure settings and then fusing them [42]. Another approach is to capture an image using HDR cameras to preserve the dynamic range in natural scenes [13]. However, these methods cannot be applied to existing images. Thus, some single image lighting enhancement algorithms are developed to address this problem.

2.2.1 Traditional Methods

Adaptive Histogram Equalization (AHE) is a contrast enhancement method that can be used to enhance images [87]. However, AHE usually produces artifacts around high contrast edges. Some other classical image processing techniques based on the Retinex theory are also used in solving this problem [60]. However, these methods often create unnatural or over-enhanced images. Wang et al. propose a method based on the bi-log transformation to enhance the image while preserving its naturalness [111]. This approach solves the issue of over-enhancing, but it cannot produce images with high visual quality. Fu et al. develop a weighted variational model to improve the prior representation and noise suppression of earlier logarithmic transformation based methods [34]. Dong et al. notice that the inverted low-light images are similar to images with haze [29]. They then apply an optimized de-hazing algorithm on inverted low-light images to enhance them. Guo et al. combine this observation and the Retinex theory and proposed LIME, a simple yet effective low-light image enhancement algorithm [40]. Ren et al. propose a Retinex-model-based decomposition method, which sequentially estimates a smoothed

illumination map and a noise-suppressed reflectance map [94]. Li et al. introduce a robust Retinex model with an additional noise map [67]. They also propose an optimization function with regularization terms for illumination and reflectance.

2.2.2 Deep Learning Approaches

Currently, many researchers are using deep learning to enhance images. An and Lee use a deep convolutional neural network (CNN) to reconstruct the radiance map from raw Bayer images [8]. Marnerides et al. use three CNN branches to process local, large pixel neighborhood and global information [77]. Gabriel et al. propose a UNet-like CNN for HDR reconstruction [32]. They train their CNN on a simulated HDR dataset created from a subset of the MIT Places dataset to make the model more generalizable. RetinexNet is a deep learning model with a decomposition net and an enhancement net that is intended for low-light enhancement [115]. Deep SR-ITM is a joint super-resolution and inverse tone-mapping framework that boosts the contrast and details of images [55]. Yifan et al. propose EnlightenGAN, which does not require paired HDR-LDR images for training [52]. EnlightenGAN also utilizes adversarial learning.

There are several HDR reconstruction/image enhancement methods that use the attention mechanism, but our approach is very different from those. Yan et al. [122] and Niu et al. [85] use the attention mechanism for multi-exposure HDR fusion while our method focuses on reconstruction using a single exposure. Li et al. [66] use this mechanism for single exposure reconstruction. Their approach models the feature interdependencies between convolutional kernels, while we model the interdependencies between different locations in the entire image.

2.2.3 Image Quality Metrics

Image quality assessment (IQA) can be performed by either subjective tests or objective tests. Subjective tests involve human evaluators, and the results are based on the ratings from those evaluators. Objective tests automate the

process of image quality assessment and can be classified into two categories: full-reference assessment (FR-IQA) and no-reference assessment (NR-IQA).

HDR-VDP is a FR-IQA developed for use under all luminance conditions. It is a comprehensive model for HDR IQA, but it only takes luminance into account and does not consider color [76].

When evaluating HDR images reconstructed by various algorithms, it is difficult to assume a reference image since camera settings and characteristics might vary for different images. Thus, no-reference IQA is more suitable for evaluating HDR reconstruction algorithms. Many attempts have been made to develop no-reference IQA models to match the results from subjective IQAs. Natural Image Quality Evaluator (NIQE) is a no-reference IQA model based on natural scene statistics. It measures the image quality only by calculating deviations from statistical regularities observed in natural images. NIQE outperforms popular FR-IQA like peak signal-to-noise ratio (PSNR) and structural similarity (SSIM)[81], [113]. Debarati et al. propose a model, HDR Image GRADient-based Evaluator (HIGRADE), for evaluating image quality of tone-mapped HDR pictures. They combine the natural scene statistic (NSS) model and HDR-specific gradient-based features into this model and validate the model on HDR datasets [59].

2.3 Visual Saliency Guided Foveated Video Compression

2.3.1 Visual Saliency

Visual saliency data gives us a description of visual fixation points and relative saliency levels in image and video frames. Saliency information can be obtained by using eye-trackers to track eye movements when viewing images and videos. However, gathering such data requires specific hardware, proper experimental setup, and many participants for subjective evaluations. Thus, researchers have proposed many visual saliency models using biological/psychological knowledge and machine learning methods.

Visual saliency detection methods can be categorized into bottom-up and

top-down models [26]. Before deep learning was widely applied in this field, most of the early methods were bottom-up models. The early methods usually involve biological and psychological research about the visual attention mechanism. Furthermore, these two approaches match common believes about the biological process of human vision. In general, these models try to establish links between visual saliency and low-level image features, such as color, contrast, and brightness [26], [50].

Differing from the above approaches, top-down models try to find factors that have the most impact on visual saliency. These models use visual saliency datasets, which contain images and their saliency annotations, for a data-driven analysis. In recent years, deep learning has been introduced into this area and has boosted the performance of saliency prediction [17], [27], [31], [51], [57], [58], [72], [112].

2.3.2 Foveated Compression

Basu *et al.* propose a variable resolution (VR) model for video conferencing and demonstrate that it can achieve higher compression rates than JPEG [15]. Wiebe *et al.* improve the performance of video transmission under the asynchronous transfer mode (ATM) protocol by introducing foveal priority dithering [116]. VR was later extend for improvement of the MPEG algorithm based on the available network bandwidth [20], transmission of 3D mesh and texture [21], and improvement of the HEVC algorithm [22]. The distinct advantage of VR-based methods is that the quality of an image changes smoothly and continuously. This prevents creating hard edges or artifacts around region boundaries.

Other research approaches usually make improvements based on existing video compression methods like JPEG2000, AVC, and HEVC. Sanchez *et al.* use a Gaussian distribution to assign different priority levels to data packets according to their distance to the region of interest (ROI) [98]. Pohl *et al.* use an eye-tracker to get real-time fixation information. They divide the video into several fixed tiles, then compress different tiles at different resolutions based on the fixation information. Another approach for foveated compression is to set

different quantization parameters (QPs) for different regions in a video frame [33], [48], [117], [128], [129]. QP controls the step length in the quantization process of coefficients. A higher QP results in larger quantization steps, which causes the decoded image quality to decrease and the compression ratio to increase. These foveated compression methods also use eye-trackers to acquire real-time saliency information and assign higher QPs to regions with higher visual saliency. Polakovič *et al*. blur the blocks in the visual periphery to remove details in those areas and consequently remove high frequency components in the transformed coefficients [88].

In conclusion, existing foveated compression methods can be classified into two main categories: VR-based methods and ROI-optimized methods based on existing video encoders. VR-based methods use pixel relocation to achieve foveation based on the distance from the fixation point. However, this approach enlarges salient areas and is less effective in handling multiple fixation points. In contrast, the method proposed in this thesis addresses these limitations by using a per-quad image warping process for foveation. In addition, ROI-optimized methods based on AVC/HEVC are limited by block-based compression, which necessitates the encoding and transmission of all pixels regardless of their saliency. However, our approach overcomes this constraint through a novel saliency-based image warping process, enabling the removal of unimportant pixels before encoding and transmission. This property also makes the proposed method compatible with most existing video compression methods.

2.4 Principal Component Approximation Network for Image Compression

Typically, image compression methods [5], [7], [9], [25], [46], [74], [89], [101], [108], [119] utilize entropy coding to reduce the statistical redundancy in image data. However, humans can better perceive low-frequency components than high-frequency components in images, and entropy coding cannot take advantage of this. Thus, since the 1960s, some transform techniques have been

proposed to address this problem, e.g., the Fourier Transform [9], Hadamard Transform [89], Discrete Cosine Transform (DCT) [5], and Wavelet Transform [25]. These transforms are effective because compression can be achieved by removing some high frequency components in the frequency domain. For example, JPEG [108], a well-known image compression method, divides an image into coding blocks. Then, the coding blocks are transformed using DCT and quantized for entropy coding [74]. JPEG2000 [25] is similar to JPEG except that it uses a wavelet transform to achieve higher compression rates. These traditional transform-based methods are widely used in image compression, but they use fixed, hand-crafted transforms to convert the images into different frequency components. This limits their performance since the hand-crafted transforms might not be optimal for all images. Our method tries to overcome this limitation by learning the shared feature matrices for image decomposition.

With the rise of deep learning networks in the computer vision field, many sophisticated methods based on deep learning have been proposed for image and video compression [2]–[4], [10]–[12], [18], [23], [24], [36]–[39], [43], [44], [53], [54], [62], [63], [65], [68], [79], [80], [82], [86], [95], [96], [102], [103], [106], [107], [110], [114], [120], [121], [127], [130], [132]. These methods typically rely on entropy encoding with an encoder-decoder architecture, where the encoder generates a compressed representation of an image, and the decoder reverses the encoding process. For example, Balle et al. [11] propose a nonlinear transform coding framework to map an image to a latent code space via a parametric analysis transform. Toderici et al. [107] propose a variable-rate learned compression method based on recurrent models, and Rippel et al. [96] propose a generative adversarial network (GAN) using an auto-encoder structure with pyramidal analysis for image compression. Recently, Hu et al. [44] propose a coarse-to-fine hyper-prior model for entropy estimation. Zhu et al. [130] propose a multivariate Gaussian mixture model for compression where a novel probabilistic vector quantization is utilized to effectively approximate the parameters of Gaussians. Rhee et al. [95] propose processing low- and high-frequency regions separately. Wodlinger et al. [120] utilize a stereo attention

module during decoding to improve image quality. Zou et al. [132] propose a window-based local attention block to learn local features.

In summary, almost all these methods incorporate deep learning networks and information entropy into existing encoder-decoder image compression pipelines to reduce redundancy inside images. These methods optimize the models to output the minimum possible number of bits to represent images, assuming that the models are general and can be applied to compress any image. Under this assumption, the bitrate calculation need not include the size of the model parameters since the models only need to be transmitted once. However, we observe that even for the same architecture, bigger models usually achieve better performance, which indicates that memorization still plays an important role in the performance of these models. Thus, the assumption of generalization might be only partially true, and the size of the model parameters might still be important for image compression. Our method takes the size of the parameters into account and tries to reduce redundancy in the encoded results as well as in the model itself.

In contrast to the above-mentioned approaches, our approach considers image compression as a matrix factorization problem. We focus on finding a number of shared feature matrices for a group of images, and the images are reconstructed by the weighted sum of these feature matrices. Specifically, weight vectors are used for encoding images and these shared feature matrices are approximated by the proposed network. In addition, when using a relatively large number of images, such as the 20,480 we used in our main experiment, promising compression results are achieved even after considering the parameters of the network. By contrast, other methods do not consider the size of the network parameters.

Chapter 3

Image Dynamic Range Enhancement Based On Fusion Pyramid

3.1 Introduction

The dynamic range in an image or video is the ratio between the largest and smallest values in luminance. A high dynamic range (HDR) image can reveal more details especially in the darkest and brightest areas than a low dynamic range (LDR) one. But software improvements or better image sensors are required to capture HDR contents. Also, re-capturing existing LDR videos using HDR techniques is probably impossible. Thus, it is important to develop algorithms to enhance dynamic range in existing videos.

Humans have a relatively high dynamic range because of the ability to control pupil size and having two types of photoreceptor cells — rods and cones. Rods are sensitive enough to respond to a single photon, which enables us to see objects in dark environments. On the other hand, cones are less sensitive, making it possible for us to see in bright environments. In recent years, advances in display technologies have enabled many devices, such as mobile phones, laptops and monitors to display HDR content. HDR content on these screens have better contrast and more details in relatively bright and dark areas [90]. They provide a better viewing experience because images on them are closer to human perception of the scene. Unfortunately, imaging systems are not able to keep pace with the advances in display devices. HDR

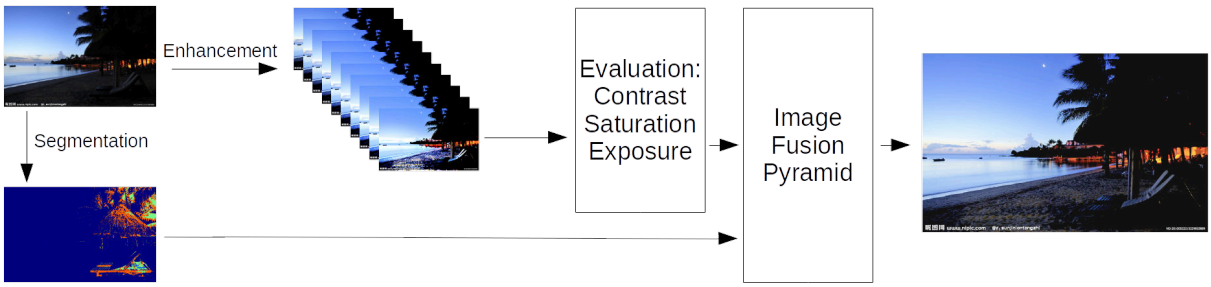

Figure 3.1: Overview of image enhancement process.

video capture is still rare, not to mention existing videos are unlikely to be shot again. Therefore, development of image and video enhancement algorithms to make them more suitable for display on HDR devices has become a necessity.

3.2 Proposed Method

3.2.1 Image Enhancement

As shown in Figure 3.1, our proposed method adopts a contrast-based fusion approach. The input image undergoes a series of processes to be enhanced. First, the image is segmented into several regions according to its luminance distribution. Then, we generate multiple enhanced images based on different enhancing parameters. The enhanced images are then evaluated for their image quality. For each region, the enhanced image with the best quality in that specific region is chosen for later fusion. Finally, the chosen images are fused together using a image pyramid.

Segmentation

The image is segmented based on the illuminance of the pixels. Every image is divided into 10 regions using the following method:

$$r = floor(\frac{x_{i,j}}{255/10}) \tag{3.1}$$

$x_{i,j}$ denotes the luminance of the pixel at location (i, j). r denotes the index of the region this pixel belongs to. Then, a morphological transformation, closing, is applied to the segmentation result to eliminate small holes inside each region. This transformation ensures the quality of the image fusion, later.

Enhancement

For the enhancement component, an illumination map estimation based approach is adopted [40]. The Retinex model regards a captured image as the result of the following formula:

$$L = R \circ T \tag{3.2}$$

where L and R are the captured image and the desired recovery, respectively. T represents the illumination map, and the operator $\circ$ denotes element-wise multiplication. Therefore, by estimating the illumination map, the desired image output can be recovered. In this paper, the illumination map is estimated using the following formula:

$$illumination = max(r, g, b)^{\gamma} \tag{3.3}$$

where r, g and b denote the red, green and blue color channels in the image, respectively. And γ is the parameter for gamma correction. The illumination is then gaussian-blurred and the pixel values in the map are clipped to fit in the range $(0, 1)$. Then, the desired image is recovered by:

$$image_{rec} = 1 - \frac{(1 - image_{ori}) - \lambda * (1 - illumination)}{illumination} \tag{3.4}$$

By manipulating λ we can control the exposure enhancement applied to the image. In this step we generate several images using different values of λ as candidate images for quality evaluation and fusion.

Quality Evaluation

The multiple candidate images generated for fusion are evaluated by three metrics: contrast, saturation and exposure. For each metric, a score map having the same size as the original image is generated. The element-wise multiplication of the three score maps is the overall quality score map of the image. HDR images usually have better contrast, more saturated colors and correct exposure. Therefore, it is reasonable to use these three metrics. For contrast evaluation, a laplacian filter is applied to the gray-scale version of the

image to generate the relative contrast map. For saturation evaluation, the image is converted to the HSV space, and the saturation plane is used as the saturation score map. For exposure evaluation, the image is converted to the Lab space. Exposure score map is generated using the following equation:

$$E = e^{-(l-0.5)^2} \tag{3.5}$$

where l is the luminance channel of the Lab image. This equation is based on the assumption that a well-exposed image should have an overall luminance close to 0.5. The three score maps are then normalized to the range $(0, 1)$ and then multiplied together.

Fusion

Fusion is done using an image pyramid to eliminate the hard edges and artifacts. For each region in the image, one of the generated candidate images with the best quality in that region is chosen. We evaluate the quality of the image in that region by calculating the sum of the values of the score map in that region:

$$Q_r = \sum_{i=1}^{n} s_i \tag{3.6}$$

where Q_r is the region quality score, s_i is the score value of pixel i and n is the total number of pixels in that region. Then, the chosen regions are concatenated together using a 5-level pyramid to get the final result.

3.3 Experiments

We compare our algorithm with three state-of-the-art methods, including HDR-CNN, HDR-ExpandNet and EnlightenGAN. The original test images and results are shown in Figure 4.5.

As can be seen from Figure 4.5, our proposed algorithm can effectively enhance images with non-optimal exposures. Our algorithm outperforms the other two deep learning based models, HDR-CNN and HDR-ExpandNet. Those two models do not work so well with dark images, while our algorithm can successfully reveal details in under-exposed areas.

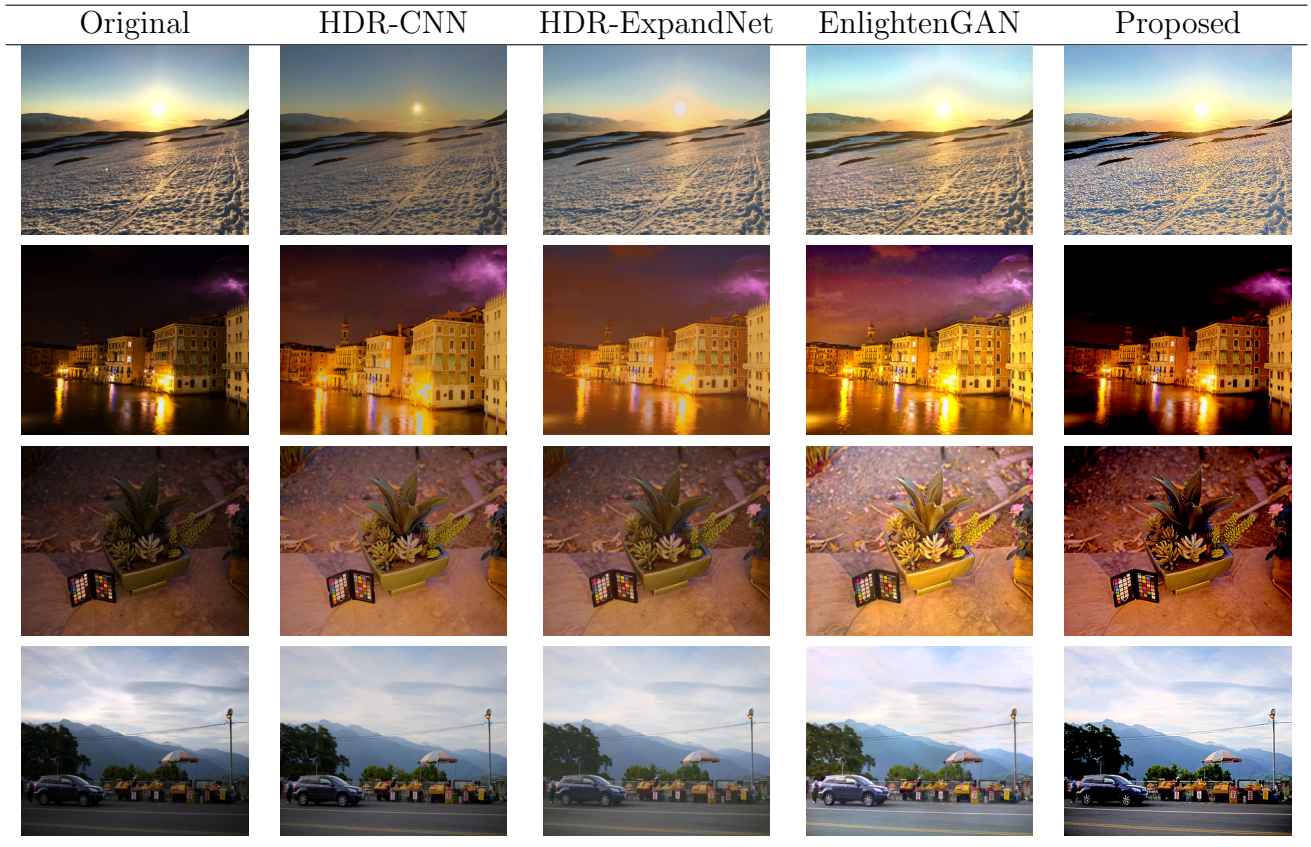

Figure 3.2: Performance comparison.

Note that EnlightenGAN tends to create images that are too bright compared to the original ones, making the night scenes appear more like a daylight scene. However, our algorithm can produce results that are more natural while revealing details. Moreover, our algorithm does not require a GPU to run and it requires less resources and time to generate the results.

We adopt Natural Image Quality Evaluator (NIQE) [81], a well-known no-reference image quality assessment for evaluating real image restoration without ground-truth, to provide quantitative comparisons. We gather the total NIQE score for all test images using the compared algorithms. The results are shown in Table 3.1. Lower scores are assigned to images that look more natural, while higher scores indicate worse quality.

We observed an improvement of 3.9% with respect to the state-of-the-art, HDR-CNN. The proposed algorithm runs significantly faster than the deep-

Algorithm	Original	HDR-CNN	HDR-ExpandNet	EnlightenGAN	Proposed
NIQE score	405.19	384.95	439.96	400.96	369.98

Table 3.1: NIQE score comparison.

learning-based methods.

3.4 Conclusion

In this chapter, we proposed a novel algorithm for enhancing the dynamic range in images. The proposed algorithm uses an image pyramid to fuse potential enhanced images together. We adopted an approach that does not require any training, and can produce more natural results than deep learning approaches. Experiments show that our algorithm achieved a 3.9% performance improvement over the state-of-the-art.

Chapter 4

Lighting Enhancement using Self-Attention Guided HDR Reconstruction

4.1 Introduction

The dynamic range in an image or video is the ratio between the largest and smallest values of luminance. High Dynamic Range (HDR) can reveal more details, especially in the darkest and brightest areas than Low Dynamic Range (LDR). HDR content can offer a better viewing experience because they are closer to human perception of the scene. Human eyes can adapt to a wide range of luminance levels by controlling the pupil and having two types of photoreceptors that work in both bright and dark environments. This is a result of adaptation to the large range of illumination values exhibited by natural scenes [84]. However, a large dynamic range causes most normal imaging systems to get either overexposed or underexposed. Unfortunately, devices that are capable of HDR image and video capture are still rare, not to mention that it is impossible to capture existing images and videos again. Thus, developing lighting enhancement algorithms for images and videos has become a necessity.

The majority of recent image lighting enhancement methods utilize CNNs. A problem with this approach is that the sizes of the convolutional kernels are relatively small and not enough to leverage long-distance or global dependencies in the images. Thus, artifacts usually appear in their reconstructions

when there is high contrast or tinted light source in the scene.

In this paper, we propose a new neural network model combining the self-attention mechanism, adversarial training, and customized loss function inspired by the HDR reconstruction process to enhance over- or under-exposed images and address the problems mentioned above. We also conduct several ablation tests to demonstrate the effectiveness of our proposed method. Our contributions are listed below:

- This is the first work to utilize the self-attention mechanism to model long-distance dependencies across different regions in images for lighting enhancement. This mechanism helps reduce the artifacts and boosts the output image quality.
- We design a new HDR loss function inspired by the characteristics of HDR images and the HDR reconstruction process. We show that this loss function can alleviate the color shift/artifacts in the output images.
- We compare our work with several state-of-the-art methods utilizing objective tests to show that our proposed method outperforms all other existing methods.

4.2 Proposed Method

As shown in Figure 4.1, we adapt a UNet-like CNN with self-attention mechanism as the generator, and a pre-trained ResNet as the discriminator. Several custom loss functions are used together with the discriminator to guide the training process of the generator.

4.2.1 Generator

The generator has an encoder-decoder structure similar to UNet [97]. Between the second and third up-sampling modules we introduce a self-attention module to eliminate local color artifacts.

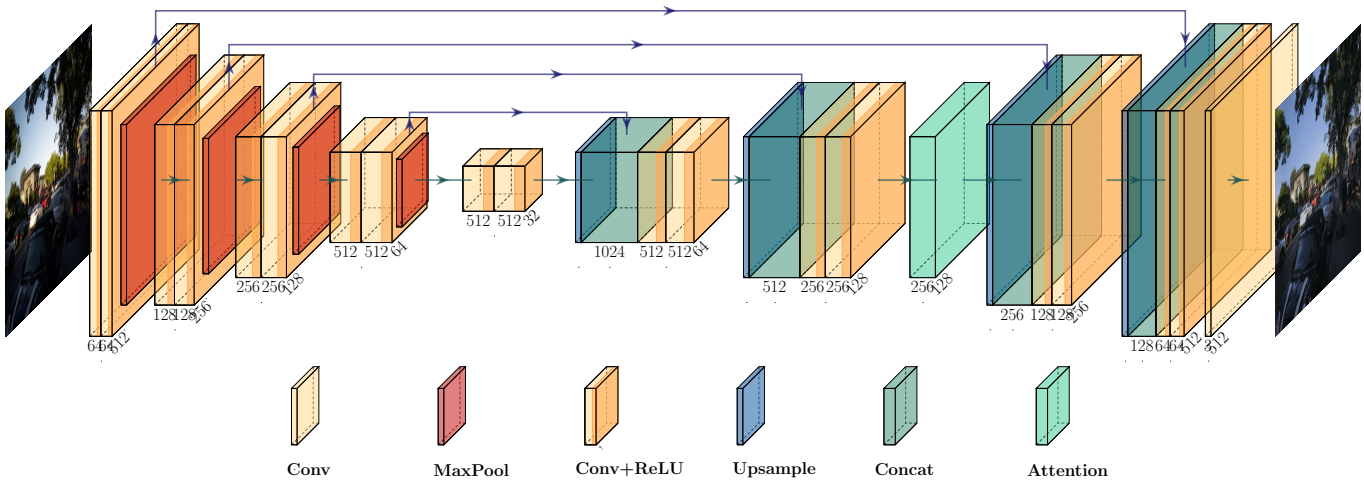

Figure 4.1: Generator structure.

4.2.2 Self-attention

Traditional CNNs can only model relatively local features due to their limited field of reception. This prevents CNNs from capturing dependencies across the entire image and creates some local color artifacts in the experiments. The self-attention mechanism is one approach to solve this problem [123].

The self-attention module in our model is complementary to the convolution layers and helps capture cues from all positions in the image to reduce artifacts introduced by pure convolution. Different from the attention module in [66], which models the interdependencies between different convolutional kernels (image feature extractors), our method focuses on modeling the spatial interdependencies across different locations in the image.

This module takes the output features of one intermediate convolutional module as input. The structure of this module is shown in Figure 4.2. This attention module works differently from convolutional kernels. The parameters in convolutional kernels cannot change after the training process, which means the features they can extract are fixed. However, the attention module computes the attention map based on the input. There are no fixed rules to decide which features are related to another one, so the attention module is more flexible than convolutional layers.

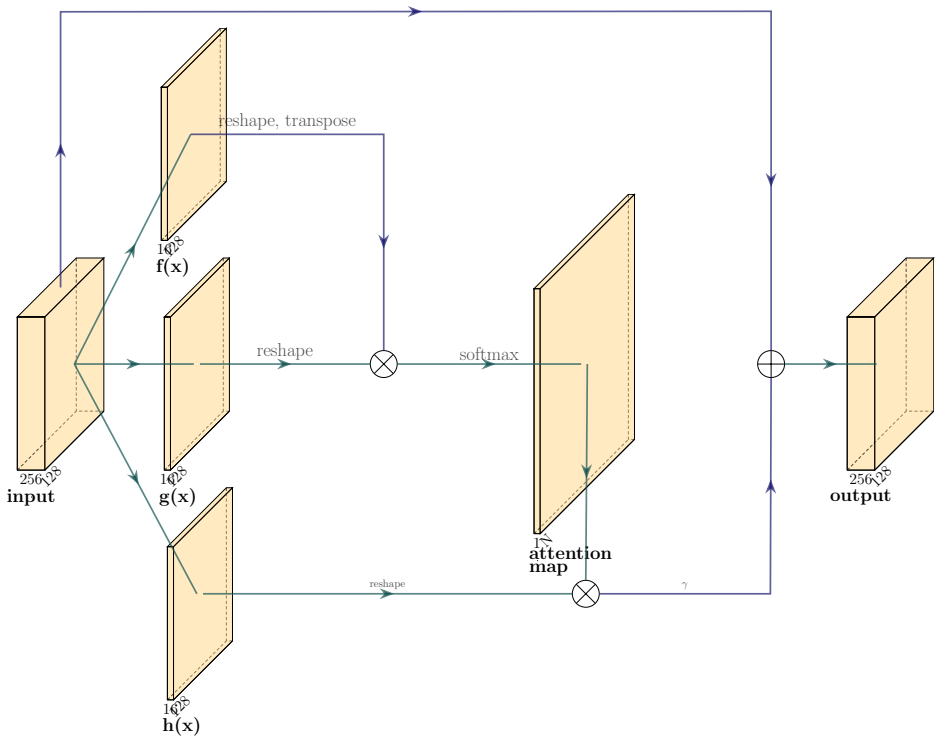

Figure 4.2: Structure of the attention module.

The input image features x of shape (C, H, W) is first transformed into two feature matrices, $\boldsymbol{f(x)}$ and $\boldsymbol{g(x)}$, for calculating the attention map, where C represents the image feature channels, H and W are feature height and width, respectively, and $\boldsymbol{f(x)} = \boldsymbol{W_f x}$, $\boldsymbol{g(x)} = \boldsymbol{W_g x}$. Here, we use a 1×1 convolutional kernel with an output channel of $C/8$ to reduce memory utilization. Thus, $\boldsymbol{f(x)}$, $\boldsymbol{g(x)}$ and $\boldsymbol{h(x)}$ are matrices of size $(C/8, H, W)$. They are then reshaped to size $(C/8, N)$ to convert the 2D features into 1D features, where $N = H \times W$.

The attention map is obtained by multiplying the transpose of $\boldsymbol{f(x)}$ with $\boldsymbol{g(x)}$ and then applying the softmax function to the result:

$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N} \exp(s_{ij})}, \text{ where } s_{ij} = \boldsymbol{f(x_i)}^T \boldsymbol{g(x_j)}. \tag{4.1}$$

$\beta_{j,i}$ denotes the extent to which the model attends to the i^{th} location when synthesizing the j^{th} region. This is the key to the self-attention mechanism as it enables the model to find out the relationship between any two locations in

the entire image. The final output is calculated using the following formula:

$$\begin{aligned} \boldsymbol{o} &= \boldsymbol{v}(\boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta}^{\boldsymbol{T}}), \text{ where} \\ \boldsymbol{h}(\boldsymbol{x}) &= \boldsymbol{W_h}\boldsymbol{x} \text{ and} \\ \boldsymbol{v}(\boldsymbol{x}) &= \gamma\boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{x} \end{aligned} \tag{4.2}$$

As shown in Figure 4.2, h represents a convolution layer using an 1×1 kernel. v is a linear transformation with a learnable parameter γ. The input features are added back to the final output.

The self-attention layer is applied after the second upsampling block of the generator.

4.2.3 Discriminator

We use a pre-trained ResNet-18 as our discriminator for its simplicity and relatively good performance. It was originally trained for classification, but in the training process it is also trained to classify good and bad HDR reconstructions. The classifier layer in the original structure is replaced with a linear transformation layer: $y = xA^T + b$, where x and y are the input and output of this layer, A is the weight matrix of size $(1, N)$ and b is a scalar bias. N is the channel size of the output from the last average pooling layer. Thus, the final result is a scalar indicating the quality of the HDR reconstruction from the generator.

We use the simple minimax GAN loss function:

$$E_x \left[\log\left(D(x)\right)\right] + E_z \left[\log\left(1 - D(G(z))\right)\right], \tag{4.3}$$

where $D(x)$ is the discriminator's estimate of the probability that the HDR reconstruction x (x is from the dataset) is good, E_x is the expected value over all good reconstructions, $G(z)$ is the generator's output when the given image is z, $D(G(z))$ is the discriminator's estimate of the probability that a reconstruction from the generator is good, E_z is the expected value over all generated reconstructions. The generator tries to minimize this loss while the discriminator tries to maximize it.

4.2.4 Custom Loss Function

We also propose two custom loss functions to help train the network.

Feature Preserving Loss

The image features are similar in the original image and its HDR reconstruction, except that the HDR version has better contrast and details. Thus, if both images are processed by a classifier network, the image features extracted should be similar. In order to preserve image features in our enhancement process, we adopt ResNet to extract image features from the input and output of the generator and build a loss function based on the mean square error between these features.

Only the first six ResNet basic blocks are used in our implementation. Two image feature tensors are obtained by feeding the original image and its HDR counterpart to the partial ResNet. The feature preserving loss is the mean square error between these two tensors:

$$L_{FP}(\boldsymbol{I}, \boldsymbol{I}^{'}) = \frac{1}{N} \sum_{i=1}^{N} \left(\varphi_i(\boldsymbol{I}) - \varphi_i(\boldsymbol{I}^{'}) \right)^2, \tag{4.4}$$

where I and $I^{'}$ are the original image and the corresponding generator output, N is the number of elements in the image features and φ denotes the partial ResNet feature extractor.

HDR Loss

HDR images usually have better contrast, more saturated colors and correct exposure. Thus, we combine these three metrics to develop a HDR loss function. This loss function helps the generator produce images that have more HDR image characteristics.

Contrast The contrast of an image is calculated by applying a Laplacian filter to the input. This operation highlights regions with rapid intensity change, which are regions with high contrast levels. By using this metric, we ensure the details in images are revealed.

The Laplacian $L(x, y)$ of an image with pixel intensity $I(x, y)$ is given by:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{4.5}$$

For simplicity we implement this using convolution. The convolutional kernel is a discrete approximation of the Laplacian function:

$$k_L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{4.6}$$

The contrast score map S_c is obtained by convolution:

$$S_c = I * k_L, \tag{4.7}$$

where I is the input image and $*$ denotes the convolution operation.

Saturation Saturation at position (i, j), $S_s(i, j)$, is calculated by converting the image from RGB to HSV color space using the following formulae:

$$\begin{aligned} C_{max}(i, j) &= \max\left(R_{i,j}, G_{i,j}, B_{i,j}\right) \\ C_{min}(i, j) &= \min\left(R_{i,j}, G_{i,j}, B_{i,j}\right) \\ S_s(i, j) &= C_{max}(i, j) - C_{min}(i, j). \end{aligned} \tag{4.8}$$

Exposure HDR images have correct exposure, and the image will not be too dark or too bright. We use the maximum value among the three color channels as the exposure value:

$$S_e(i, j) = \max\left(R_{i,j}, G_{i,j}, B_{i,j}\right) \tag{4.9}$$

Finally, the HDR loss is calculated using the following formulae:

$$\begin{aligned} L_{HDR}(\boldsymbol{I}, \boldsymbol{I}') &= \sum \left[S(\boldsymbol{I}) - S(\boldsymbol{I}')\right]^2, \text{ where} \\ S(\boldsymbol{I}) &= S_c(\boldsymbol{I}) S_s(\boldsymbol{I}) S_e(\boldsymbol{I}) \text{ and} \\ S(\boldsymbol{I}') &= S_c(\boldsymbol{I}') S_s(\boldsymbol{I}') S_e(\boldsymbol{I}') \end{aligned} \tag{4.10}$$

The training procedure is described in Algorithm 2. z and y refer to the original images and ground truth HDR reconstructions in the dataset.

Algorithm 1: Training procedure using GAN and custom loss functions

1. Initialize the parameters in the generator G and the discriminator D;

while *maximum training steps not reached* **do**

2. Load one batch of training data (z, y) from the dataset;

Stage 1: Train discriminator

3. Generate the output $G(z)$ by passing z to the generator;
4. Get the response for the ground truth and the output from the discriminator, $D(y)$ and $D(G(z))$;
5. Use gradient descent to maximize $D(y) - D(G(z))$;

Stage 2: Train generator

6. Calculate the total loss $L_{total} = L_{FP} + L_{HDR}$;
7. Use gradient descent to maximize $D(G(z))$ and minimize L_{total};

end

4.3 Experiments

4.3.1 Datasets

Samule et al. developed a computational photography pipeline including capturing, aligning, and merging a burst of frames to reconstruct HDR images. They captured images using a variety of Android mobile cameras and processed them through this pipeline. The results are published as the HDR+ Burst Photography Dataset. The dataset consists of 3640 bursts with 28461 images in total [42]. Every burst in this dataset is made up of several raw images and one final resulting image. Even though the exposure time and gain are the same across all images in a burst, the images are generally different from one another.

We retrieve paired images from the dataset by pairing a random raw image with the corresponding final result image. In this way, we augment our data to further enlarge the datasets. In the training process we use 3340 image pairs from the HDR+ Burst Photography Dataset. We use the remaining 280 non-synthetic images pairs as the test data.

4.3.2 Evaluation Metrics

NIQE is a no reference, opinion unaware, distortion unaware IQA model [81]. "Opinion unaware" means that this model does not require training on images that have quality scores, and "distortion unaware" means that this model does not use specific distortion patterns to evaluate images. NIQE calculates the image features using a NSS model and then fits these features to a multivariate Gaussian (MVG) model. The NIQE score is the distance between this fit and the MVG model derived from a set of natural images. Hence a lower NIQE score indicates that the input image is more natural or has better quality.

HIGRADE is a metric for evaluating HDR images. Similar to NIQE, it uses an established NSS model to extract image features [59]. Other than this, some new image features are designed to consider artifacts during HDR processing, including log-derivatives and gradient domain scene statistics. HIGRADE-1 and HIGRADE-2 are two variants of this model which incorporate different gradient domain scene statistics. HIGRADE-1 utilizes gradient magnitude features, which is widely used in IQAs, while HIGRADE-2 explores the advantage of a less researched feature, gradient orientation information. Higher HIGRADE scores indicate higher quality.

All other image quality metrics, including PSNR, SSIM, MS-SSIM, and HDR-VDP-3, are full reference IQAs. We use the ground truth HDR images from the datasets as references.

PSNR can be calculated using the following formula:

$$PSNR = 10 \log_{10}(\frac{m^2}{\boldsymbol{MSE}}), \tag{4.11}$$

where $\boldsymbol{MSE}$ is the mean square error between the test and reference images, and m is the maximum value possible in the image (255 for 8-bit images). Higher scores are better.

The SSIM index is calculated on various windows of an image. The measurement for two $N \times N$ windows around (x, y) is defined as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{4.12}$$

In this equation, μ_x and μ_y are the local means. σ_x and σ_y are the local standard deviations, and σ_{xy} is the cross-covariance. C_1 and C_2 are the regularization constants for the luminance and contrast. $C_1 = (0.01L)^2$ and $C_2 = (0.03L)^2$, where L is the dynamic range of the input image. Higher SSIM scores are better with a maximum of 1.

Multiscale SSIM (MS-SSIM) measures the image quality at several scales, and it is more robust than SSIM.

HDR-VDP is a visual metric that compares a pair of images (a reference and a test image) and predicts visibility (probability that the difference at a certain location is noticed by a person) and quality of the test image (expressed as a mean-opinion-score). HDR-VDP is based on a new visual model which is derived from contrast sensitivity measurements. Thus, this model can work with images having arbitrary illuminance ranges. We use the quality evaluation part of this metric; higher scores are better with a maximum of 100.

4.3.3 Ablation Test

We train our model in several different settings using various techniques mentioned in the last section. This can help us understand the function of each part in the model. The settings we use is listed in Table 4.1. Note that Model 6 is the one we use to report our final performance. Model 1 is used as a performance baseline since only mean square error (MSE) loss is used as the loss function. Model 2 is trained using only the HDR loss, and Model 3 is trained using GAN. For Model 4, both GAN and HDR loss are used. We introduce feature preserving loss and self-attention module in Models 5 and 6.

These six models were tested using NIQE and two HIGRADE metrics, with the results listed in Table 4.2. From the scores of Models 1-3, it is clear that when only HDR loss or GAN is used, the model performs worse than the baseline model. But the scores of Model 4 indicates that those techniques work better when combined together. Model 5 improves the NIQE score and HIGRADE1 score to 4.5651 and 0.1154, respectively, with the help of the feature preserving loss. Model 6 increases the performance considerably, indicating

Table 4.1: Model settings.

Model	GAN	Self-attention	MSE Loss	Feature Preserving Loss	HDR Loss
1			*		
2					*
3	*				
4	*				*
5	*			*	*
6	*	*		*	*

that the self-attention module is a key part of the model.

Table 4.2: Model performance.

Model	NIQE	HIGRADE1	HIGRADE2
1	4.83	-0.26	-0.22
2	5.13	-0.26	-0.29
3	5.07	0.04	0.02
4	4.82	0.06	-0.02
5	4.57	0.12	0.07
6	3.14	0.27	0.33

Figure 4.3 includes output images from the six models when the same input image is used. The input image and corresponding ground truth are also shown in Figure 4.3. All the models reveal the details in the dark areas, but Model 1 does not perform as well as the others. The output of Model 3 has a brown tinge. Adding HDR loss helps alleviate this problem, and makes the image look more natural. From the output of Model 5, we can see that feature-preserving loss makes the color of the image closer to the original one. The output of Model 6 has the best overall image quality.

4.3.4 Objective Image Quality Comparison

Comparisons are made between LIME [40], HDR-CNN [32], EnlightenGAN [52], HDR-ExpandNet [77], RetinexNet [115], Deep-SR-ITM [55] and our proposed method. HDR-CNN and HDR-ExpandNet originally produce HDR images with linear luminance levels, so the output of those two methods is tone-mapped using the Reinhard curve [93].

Some example output images are shown in Figure 4.5. The original images

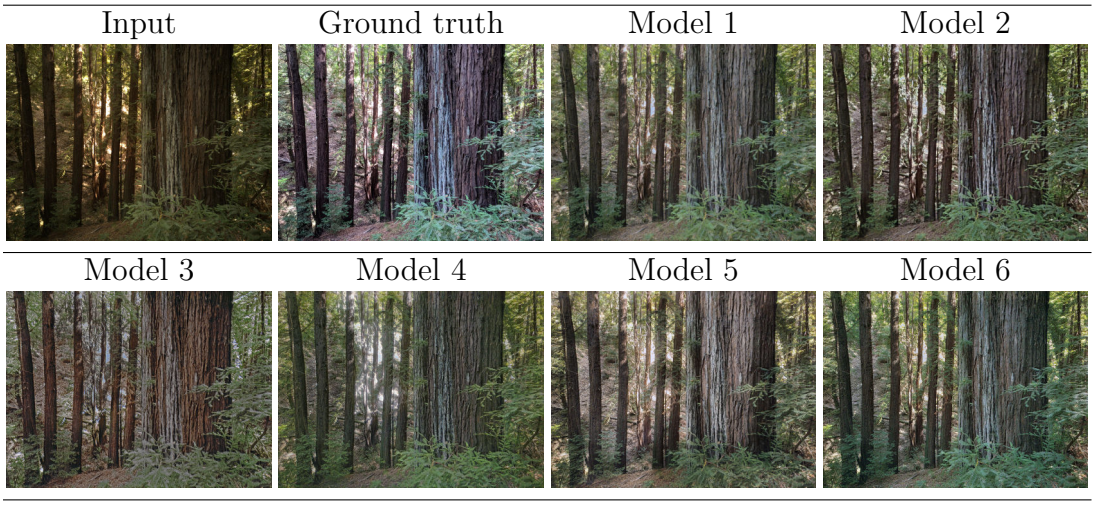

Figure 4.3: Ablation test image examples.

Figure 4.4: Reconstruction details of different methods.

and output images of all compared methods are shown in various columns. RetinexNet's output contain many unnatural colors and noise. The output of Deep-SR-ITM has a strong blue tinge, and HDR-ExpandNet produces some images that look washed out. Compared to LIME, HDR-CNN, EnlightenGAN, and our method, the other methods perform worse in terms of image quality.

For the first two test images, EnlightenGAN seems to be the best, followed by our proposed method. Both methods are able to reveal details in dark areas, while LIME is not able to recover dark areas in the second image. For yellowish images, like Images 3-6 and 13, LIME, HDR-CNN, and EnlightenGAN appear to ignore this problem and produce more yellow-orangish images, but our method is able to identify the problem and produce more reasonable colors. For images that include large parts of sky in them, for example, Images 2, 6, 8, and 11, the proposed method appears to be better at recovering the colors of the sky and details of the clouds. Furthermore, for images with large contrast or relatively high brightness, such as Images 10-12, LIME introduces some noise into the output, and HDR-CNN's and EnlightenGAN's output contain color artifacts. In comparison, our method can successfully recover details without introducing noticeable noise and artifacts. Figure 4.4 gives an example of some details in the reconstructed images of LIME, HDR-CNN, EnlightenGAN, and the proposed method. LIME fails to reconstruct the details in those two areas. HDR-CNN and EnlightenGAN introduce noise in both regions, while the proposed method is able to recover details without producing noise.

Table 4.3: Image Quality Comparison. Scores in red indicate the best performance, scores in blue indicate the second-best performance.

IQAs	PSNR	SSIM	MS -SSIM	NIQE	HDR-VDP-3	HI-GRADE1	HI-GRADE2
Input	14.94	0.48	0.68	3.73	6.53	-1.16	-1.19
LIME	14.23	0.40	0.68	4.32	6.95	-0.55	-0.46
HDR-CNN	14.88	0.45	0.62	2.73	6.67	-0.29	-0.00
EnlightenGAN	12.56	0.43	0.68	3.08	6.24	0.12	0.29
HDR-ExpandNet	15.47	0.46	0.68	3.30	6.86	-1.10	-0.75
RetinexNet	12.20	0.31	0.52	7.85	5.23	-0.06	0.25
Deep-SR-ITM	14.39	0.49	0.67	4.24	6.75	-0.09	-0.07
Proposed	17.01	0.48	0.70	3.14	7.26	0.27	0.33

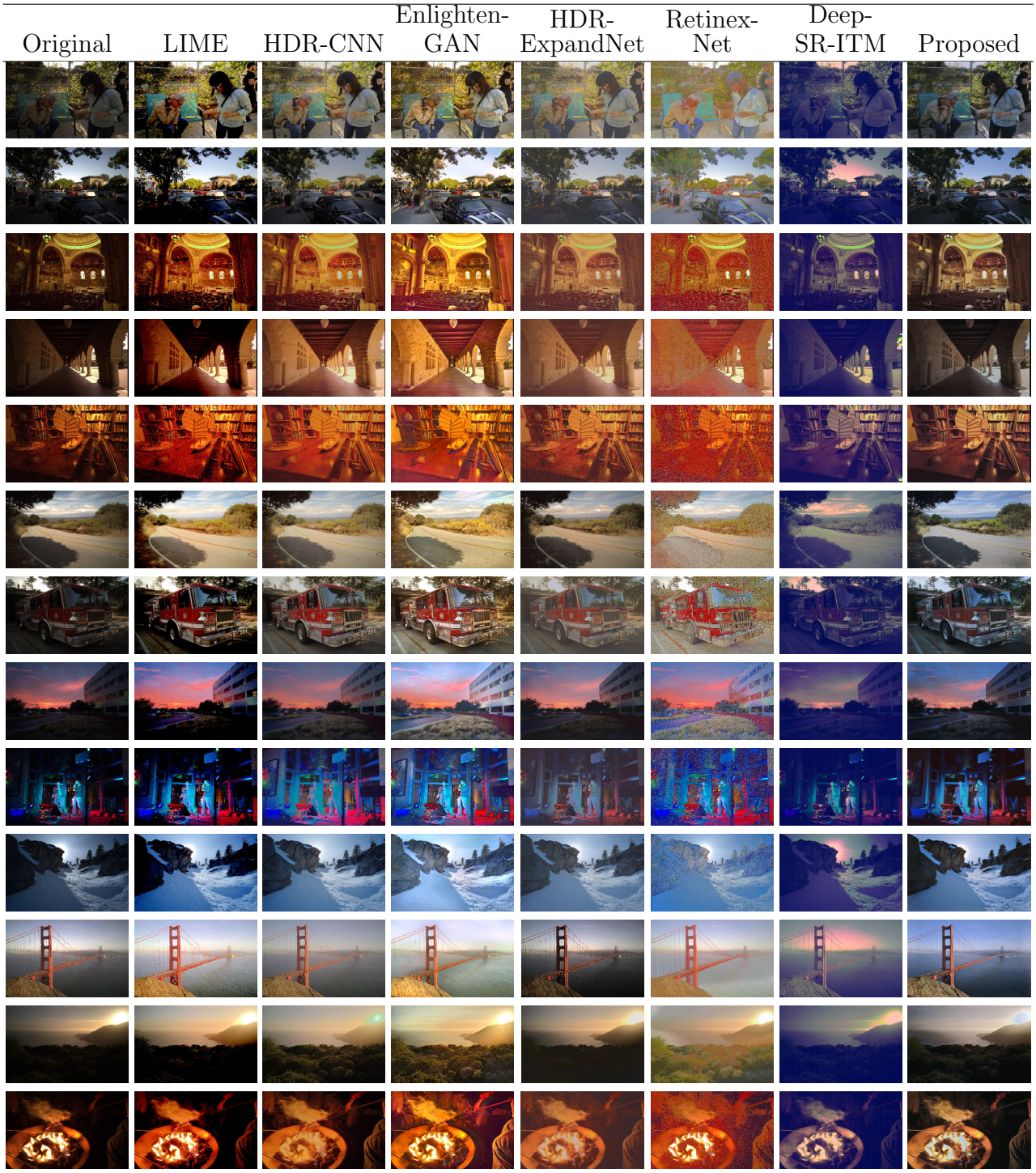

Figure 4.5: Example output of various methods.

We use both FR-IQAs and NR-IQAs to measure the image quality of the above-mentioned algorithms. Note that for NIQE, a lower score is better,

while for all other metrics, a higher score is better. The results are listed in Table 4.3.

For the four FR-IQAs, all the models generate relatively low scores. For example, an image with good quality usually gets a PSNR higher than 40 dB when compared to its reference, but all models score lower than 20 dB. The same situation occurs in evaluations using SSIM, MS-SSIM, and HDR-VDP-3. We believe the reason for this is that the output images are very different from the reference images from the dataset. The reference images are synthesized using multiple images with different exposure settings, but all compared models produce their output based on only one input image. Thus, the reconstructions are very different since less information is available.

However, the results still show that our model can produce the output with the highest quality. Our PSNR score shows that the output image of our model has a 1.6 times higher signal-to-noise ratio than the output of the model in the second place. The proposed method comes in second place in the SSIM test, but only falls short by a small margin. As for MS-SSIM and HDR-VDP-3, the proposed method ranks first and leads by a large margin.

For the NIQE test, the proposed method performs worse than HDR-CNN and EnlightenGAN, but is much better than the other methods. For HIGRADE-1 and HIGRADE-2, our model scores the highest.

In summary, the proposed model performs better than all other methods for 5 of the 7 IQAs we used in the comparisons. This shows that our model can effectively enhance SDR images and attain good image quality.

4.4 Conclusion

We proposed an encoder-decoder model with adversarial training, a self-attention mechanism and customized loss functions to enhance incorrectly exposed images. To the best of our knowledge, we are the first to combine adversarial training with a self-attention mechanism to improve reconstruction quality. This helps our network exploit the location interdependency and reduce artifacts. Objective comparisons between our proposed model and several other

state-of-the-art methods were conducted. These comparisons demonstrated that our model performs better in terms of image naturalness and the ability to adapt to different lighting conditions.

Chapter 5

Visual Saliency Guided Foveated Video Compression

5.1 Introduction

Widely applied video compression methods, *e.g.* AVC [118], HEVC [104], VP8 [14], VP9 [83] and AV1 [41], use block-based algorithms to reduce spatial and temporal redundancy. A video frame is first divided into several blocks, then the encoder performs intra-frame (for spatial redundancy) or inter-frame (for temporal redundancy) predictions according to the frame type. Blocks might be partitioned into smaller ones in this process. The encoder calculates the errors and transmission costs of different prediction modes and partitioning patterns, and records the best performing combination for transmission. Next, a block-wise transform is applied to the prediction errors, resulting in coefficients in another domain. The discrete cosine transform (DCT) and discrete sine transform (DST) are commonly used as block-wise transforms. The coefficients are then quantized and encoded into a bitstream.

Various compression methods can effectively eliminate spatial and temporal redundancies. However, the spatially-varying sensing characteristics of the Human Visual System (HVS) are often not considered. Humans have two types of photoreceptors in the eye, namely rods and cones [109]. Rods and cones are unevenly distributed across the human retina. Cones have the highest density in the fovea, the center of the retina, while rods are almost absent in the fovea and reach their highest density in a 10 to 20 degree periphery of the

fovea. Rods and cones also have different sensitivity to light. Rods support vision under low illumination levels, while cones support vision under normal and higher brightness. Even though rods have higher sensitivity, their visual acuity under low illumination is extremely poor compared to visual acuity under photopic conditions. The reason for this is that signals from many rods converge onto a single neuron within the retina. This improves sensitivity in exchange for spatial resolution. On the other hand, every cone is connected to multiple neurons, and they have a high density in the fovea. This means that the fovea has a higher spatial resolution than the periphery. As a result, the Human Visual System encodes more information from the center of the receptive field, and less information from the periphery.

Existing compression methods treat all parts of a video frame equally, encoding all blocks with the same resolution. Resolution scales evenly as the target video resolution changes. This introduces perceptual redundancy since information in the periphery is sampled at a lower spatial resolution due to the characteristics of the HVS. To eliminate this redundancy, different blocks in the video needs to be encoded with different resolutions depending on their locations. The blocks in the periphery should be encoded with a lower resolution, while the blocks in the fovea should be encoded with a higher resolution. In this paper, we propose a novel video compression method which incorporates the non-uniform spatial resolution of the HVS to reduce perceptual redundancy. The proposed method has the following novel features:

- A foveation process based on per-quad image warping is used to preserve image quality of salient regions, achieving non-uniform subsampling based on saliency level.
- The saliency data is incorporated at a lower granularity, providing more precise quality control of salient regions.
- Our method is independent of traditional encoding processes, making it applicable to improve most existing compression methods.

5.2 Proposed Method

5.2.1 Overview

Our method aims to reduce perceptual redundancy, which is usually not handled by widely used video compression methods. As mentioned in the introduction, HVS encodes more information from the center than the periphery of the receptive field. Thus, humans are more sensitive to quality degradation around the fixation point. Furthermore, the salient area, in a single video frame, i.e., the area that needs to have relatively high quality after compression, is usually a very small part of the frame given the limited viewing time for each frame. Based on these two factors, we design an algorithm to subsample different regions of a video frame at different sampling rates according to their saliency. Pixels in salient areas are sampled at a higher sampling rate to preserve the image quality in those areas, while other pixels are sampled at a much lower sampling rate to effectively reduce perceptual redundancy. The overall pipeline of our approach is illustrated in Figure 5.4.

5.2.2 Saliency Encoding

The saliency map is a grayscale image. This map needs to be transmitted to the decoder to provide the necessary information for image reconstruction. But transmitting it as an image significantly increases the data size. As an alternative, we use a few parameters to describe this saliency map and only transmit the parameters for reconstruction on the decoder end. Such a simplification is possible because the saliency maps are formed by a combination of several gaussian distributions.

The saliency datasets we use are collected using eye-trackers or similar technologies. The direct output of these devices are fixation points instead of saliency maps. For every frame, a collection of fixation points is generated: $C = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n) \,|x_i, y_i \in \mathbb{R}^2\}$, where (x_i, y_i) are the coordinates of a single fixation point i, and n is the total number of fixation points.

Then, the saliency map is generated using Eq. 5.1.

$$\begin{aligned} f(x_j, y_j) = \sum_{i=1}^{n} A_i \exp -(a_i(x_j - x_{0i})^2 \\ +2b_i(x_j - x_{0i})(y_j - y_{0i}) \\ +c_i(y_j - y_{0i})^2) \end{aligned} \tag{5.1}$$

Function $f(x_j, y_j)$ gives the saliency value at location (x_j, y_j). A is the amplitude of the distribution and (x_0, y_0) is the center of the distribution. a, b, c are the other three parameters that are used to define the distribution in Eq. 5.2.

$$\begin{aligned} a_i &= \frac{\cos^2\theta_i}{2\sigma_{Xi}^2} + \frac{\sin^2\theta_i}{2\sigma_{Yi}^2} \\ b_i &= -\frac{\sin 2\theta_i}{4\sigma_{Xi}^2} + \frac{\sin 2\theta_i}{4\sigma_{Yi}^2} \\ c_i &= \frac{\sin^2\theta_i}{2\sigma_{Xi}^2} + \frac{\cos^2\theta_i}{2\sigma_{Yi}^2} \end{aligned} \tag{5.2}$$

θ is the angle of the long axis of the distribution blob. σ_X and σ_Y are the standard deviations along the X and Y axes, respectively.

Let $\mathcal{D}$ be the set of parameters: $\{(A_i, a_i, b_i, c_i, x_{0i}, y_{0i}) \,|i = 1, 2, ..., n\}$. Then, the parameterization of the saliency map can be formulated as an optimization problem; namely, finding the $\mathcal{D}$ that minimizes the difference between the actual saliency values and the fitted values generated using $\mathcal{D}$ in Eq 5.1:

$$\mathcal{D}_{min} = \underset{\mathcal{D}}{\operatorname{argmin}} \sum_{j=0}^{m} \left(f_{\mathcal{D}}\left(x_j, y_j\right) - S_{x_j, y_j} \right)^2 \ , \tag{5.3}$$

where $\mathcal{D}_{min}$ is the optimal parameter set, m is the total number of pixels, S_{x_j,y_j} is the saliency value in the ground truth saliency map at location (x_j, y_j), and $f_{\mathcal{D}}\left(x_j, y_j\right)$ is the fitted saliency value at (x_j, y_j) calculated using f with the parameter set $\mathcal{D}$. This equation can be solved using a non-linear least-squares solver.

5.2.3 Foveation using Image Warping

In foveated compression, the quality of the salient areas needs to be preserved and this restriction gradually relaxes as the distance from the salient areas

increases. The variable resolution transformation [15] introduces one way to approach this problem, by placing pixels to new locations based on their distances to the fixation point. After VR transformation, the area around the fixation point is enlarged and the other areas are squeezed. However, this is not optimal since the salient area takes up more space than in the original image. It also reduces the space available for other areas and impacts the overall image quality. Furthermore, when dealing with multiple fixation points, two methods can be used with VR, namely, collaborative foveae and competing foveae. However, which method is better for compression cannot be determined before applying them to assess the resulting image quality. To address these issues, we propose a subsampling strategy inspired by feature-aware texturing [35].

Problem formulation

Our goal is to find an image warping function that can reduce the total number of pixels in an image by sub-sampling, while maintaining the image quality of salient regions. Specifically, the function W: $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ defines a mapping of pixel locations from the original image to the warped image.

$$W(x_i, y_i) = (x_i', y_i') \tag{5.4}$$

In Eq. 5.4, (x_i, y_i) is the coordinate of a pixel in the original image, and $x_i \in (1, h), y_i \in (1, w)$, where h and w are the height and width of the original image, respectively. Similarly, (x_i', y_i') is the coordinate of a pixel in the warped image, with $x_i' \in (1, h'), y_i \in (1, w')$, where h' and w' are the height and width of the warped image, respectively. Since we are reducing the total number of pixels, we have $hw > h'w'$. To preserve image quality in salient areas, W needs to sample the salient regions at a higher sampling rate, and other regions at a lower sampling rate. We divide the original image into rectangular grids, and denote the resulting mesh as $G = (V, E, F)$, where $V = v_1, v_2, ..., v_n$ is the set of vertices of the mesh, E is the set of edges between adjacent vertices, and F is the set of faces formed by vertices and edges. We denote the set of quads formed by four adjacent vertices as $Q = \{Q_{ij} : (v_{ij1}, v_{ij2}, v_{ij3}, v_{ij4})\}$, where Q_{ij} is the quad located at the i$^{\text{th}}$ row (r rows in total) and j$^{\text{th}}$ column (c columns

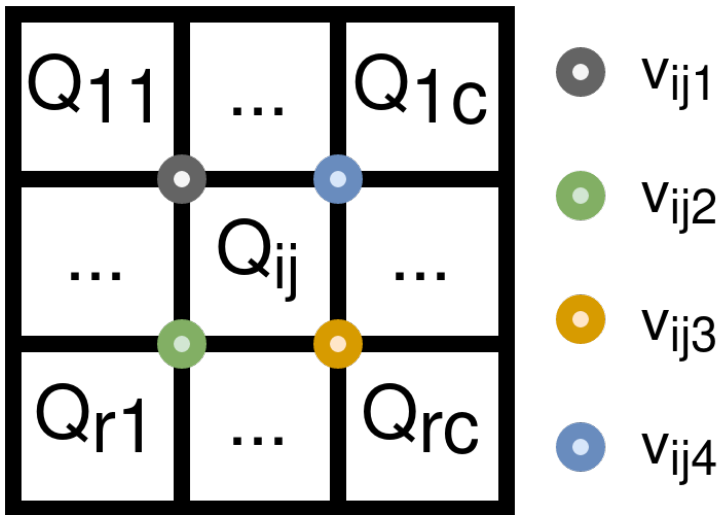

Figure 5.1: Quads and quad vertices.

in total), and $v_{ij1}, v_{ij2}, v_{ij3}, v_{ij4}$ are the four vertices. The quads are illustrated in Figure 5.1.

The saliency map associated with the input image specifies visual saliency at the pixel level. We divide this saliency map using the same mesh and obtain the set of faces $F_s = S_{ij}$, where S_{ij} is the face corresponding to Q_{ij}. We define the salient areas as the set of quads Q_s, whose average saliency level exceeds the threshold s_t. In general, a smaller saliency threshold will result in a larger area being categorized as salient, and vice versa. In cases where saliency predictions may not be precise, a smaller threshold value is recommended as it increases the probability of capturing the actual salient regions by expanding the labeled salient regions.

$$Q_s = \left\{ Q_{ij} | \frac{\sum_{p=1}^{k} m_p}{k} > s_t, m_p \in S_{ij} \right\} \tag{5.5}$$

In Eq. 5.5, m_p is the pixel value in the saliency map. Now, we can formulate the problem as finding the warping function to transform all quads in Q to reduce the image size while maintaining the size of any $Q_{ij} \in Q_s$.

Feature Preserving Mesh Transformation

To preserve the quality of salient areas, a bigger portion of pixels are sampled in any $Q_{ij} \in Q_s$ than in other quads. The result of this is that any $Q_{ij} \in Q_s$

contains more pixels than other quads and thus has a larger area. The variation in size makes it hard to describe the whole transform as a single warping function. Thus, we carry out the transformation on a per quad basis. Similar to the VR transformation, our method results in salient regions being enlarged and other regions being squeezed. As a consequence, the relative offsets of quads to the origin might change after the transformation and it is not trivial to calculate the new offsets. We decided to make the translation of each quad a free parameter if other restrictions are met.

The transformation of a quad can be expressed as the transformation of its four edges. We denote the four edges as vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3, \boldsymbol{e}_4$, where:

$$\begin{aligned} \boldsymbol{e}_{ij1} &= v_{ij1} - v_{ij2} \\ \boldsymbol{e}_{ij2} &= v_{ij2} - v_{ij3} \\ \boldsymbol{e}_{ij3} &= v_{ij3} - v_{ij4} \\ \boldsymbol{e}_{ij4} &= v_{ij4} - v_{ij1}. \end{aligned} \tag{5.6}$$

The transformation can be performed as a matrix multiplication:

$$t\tilde{\boldsymbol{e}}' = \boldsymbol{H}\tilde{\boldsymbol{e}}, \tag{5.7}$$

where $\tilde{\boldsymbol{e}}'$ and $\tilde{\boldsymbol{e}}$ are the transformed and original homogeneous coordinates of the edge in the form: $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, t is a multiplier to turn $\tilde{\boldsymbol{e}}'$ into homogeneous coordinates, and $H = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ c_0 & c_1 & c_3 \end{bmatrix}$ is the transformation matrix.

Then, the target transformed edge can be calculated as:

$$\begin{aligned} \tilde{\boldsymbol{e}}'_k &= \tilde{\boldsymbol{v}}'_k - \tilde{\boldsymbol{v}}'_{k+1} \\ &= \boldsymbol{H}_k\tilde{\boldsymbol{v}}_k - \boldsymbol{H}_k\tilde{\boldsymbol{v}}_{k+1},\ k = 1, ..., 4 \text{ cyclically}, \end{aligned} \tag{5.8}$$

where $\tilde{\boldsymbol{v}}_k$ represents the homogeneous coordinates of the vertex v_k. This equation defines the relationship between the transformed vertices and the original ones.

For any $Q_{ij} \in Q_s$, only translation is allowed since we want to maintain the original size. Thus, their transformation matrices have the form: $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$,

where t_x and t_y are translation parameters. Using Eq. 5.8 enables t_x and t_y to be free. A total of $4N_s$ linear equations can be obtained from Eq. 5.8, where N_s is the number of $Q_{ij} \in Q_s$.

For all other quads, we want them to scale with the entire image using the same scaling ratio. Thus, their transformation matrices have the form: $\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$, where s_x and s_y are the scaling ratios for the height and width, respectively. A total of $4N_{ns}$ linear equations can be obtained from Eq. 5.8, where N_{ns} is the number of $Q_{ij} \notin Q_s$.

Furthermore, vertices on the boundaries of the original image should stay on the boundaries after the transformation. Thus, for these vertices, Eq. 5.9 is used.

$$\tilde{\boldsymbol{v}}'_k = \boldsymbol{H}_k \tilde{\boldsymbol{v}}_k \tag{5.9}$$

In total we have $4(N_s + N_{ns})$ equations and they form a system of linear equations. This system is overdetermined, so we can solve it using least squares. Solving this system for $\tilde{\boldsymbol{v}}'$ gives us the transformed homogeneous coordinates with their squared errors to the desired coordinates minimized.

Saliency Guided Weighting

In the system of linear equations, for any single equation, multiplying both the left and right-hand sides with the same weight parameter w does not break the equality. However, when solving this system in the least squares sense, adding the weight w causes the squared residual to be multiplied by w^2. Consequently, the transformed locations of vertices with bigger weights will be closer to their location estimated by the given transformation. This means the shapes and sizes of quads containing those vertices are better preserved. Thus, we apply the average saliency level of a quad as the weight w to all four edges in the quad. The corresponding equations are changed to:

$$\begin{aligned} w\left(\tilde{\boldsymbol{v}}'_k - \tilde{\boldsymbol{v}}'_{k+1}\right) &= w\left(\boldsymbol{H}_k \tilde{\boldsymbol{v}}_k - \boldsymbol{H}_k \tilde{\boldsymbol{v}}_{k+1}\right) \\ k &= 1, ..., 4 \text{ cyclically, where,} \\ w &= max(\overline{m_p}, 1) \ , \ m_p \in S_{ij} \ . \end{aligned} \tag{5.10}$$

We use the value 1 as the minimum of the saliency level.

5.2.4 Salient Area Scaling

It is not always possible to keep the size of the salient areas unchanged. If the target compression scale is too small, the salient areas will also need to be scaled to make sure they do not fall outside the compressed image. Thus, we calculate the maximum possible scale for the salient area using Eq. 5.11. (x_{ij}, y_{ij}) is the coordinate of a point in S_{ij}, and s_{margin} is a parameter to control the space reserved for peripheral regions, as shown in Figure 5.2. The required salient area scales, s_{rsx} and s_{rsy}, are defined as the maximum ratio of the area occupied in the x and y axes plus the two margins. Then, the maximum possible scales are calculated by dividing the target scales by the required salient area scales. If the result is larger than 1, 1 is used as the scale.

$$
\begin{aligned}
s_{sx} &= min\left(\frac{s_x}{s_{rsx}}, 1\right) \\
s_{sy} &= min\left(\frac{s_y}{s_{rsy}}, 1\right) \\
\text{where,} & \\
s_{rsx} &= \frac{max(x_{ij}) - min(x_{ij})}{h} + 2 \times s_{margin} \\
s_{rsy} &= \frac{max(y_{ij}) - min(y_{ij})}{w} + 2 \times s_{margin}
\end{aligned}
\tag{5.11}
$$

Peripheral Image Quality Constraints

The non-salient parts of the image might suffer from a loss of quality because of the sudden change in the transformation method at the boundaries of salient regions. The salient quads are only allowed to translate, while the non-salient quads are allowed to translate and transform perspectively. This results in a relatively big deformation at the boundaries, as shown in Figure 5.3. To address this problem, we develop a smoothing weighting method and introduce a uniform constraint on non-salient quads.

The weights of non-salient quads are defined in Eq. 5.12.

$$
w = max(\overline{m_p}, 1) \ , \ m_p \in S'_{ij} \ , \tag{5.12}
$$

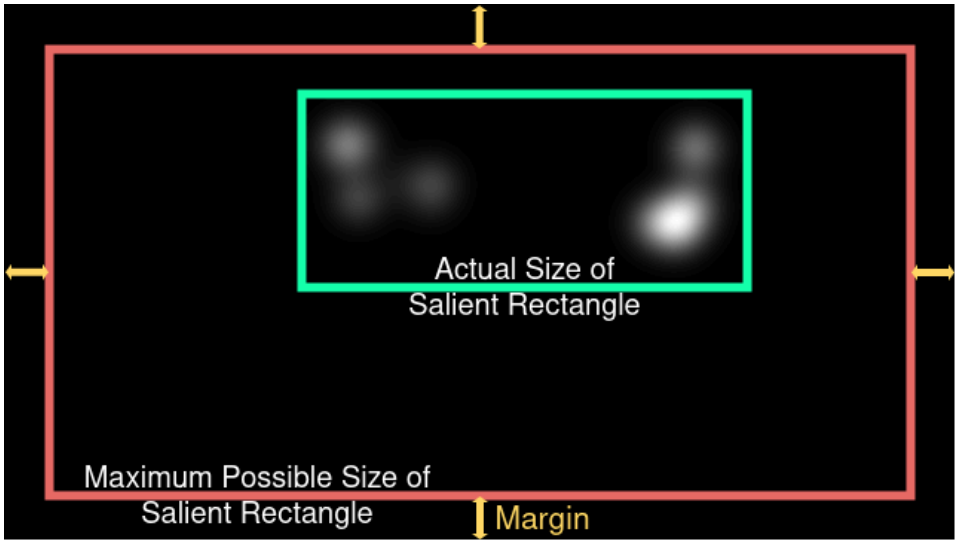

Figure 5.2: Salient area scaling.

where S' is the new saliency map generated using the parameters discussed in Section 5.2.2. Specifically, $1.5\sigma_X$ and $1.5\sigma_Y$ are used to increase the saliency level on the boundaries, consequently increasing the weight of quads in that region. This also ensures that the saliency level changes smoothly from the fixation centers to peripheral regions, which prevents generating artifacts due to sudden changes in weight.

Furthermore, we introduce the uniform constraint as a set of additional linear equations to further alleviate this problem. To make sure pixels are uniformly sampled in the non-salient quads, one intuitive approach is to make sure quads on the same row (column) have the same width (height).

$$w\left(\tilde{\boldsymbol{v}}'_k - \tilde{\boldsymbol{v}}'_{k+1}\right) = w\left(\tilde{\boldsymbol{v}_0}'_k - \tilde{\boldsymbol{v}_0}'_{k+1}\right) \\ , k = 1, ..., 4 \text{ cyclically} \tag{5.13}$$

Thus, the linear equations can be formulated as Eq. 5.13, where $\tilde{\boldsymbol{v_0}}$ denotes the vertices in the first quad of this row (column), and w is calculated using Eq. 5.12.

As illustrated in Figure 5.3, when transforming the mesh without any constraints, the salient areas near the center overlap with each other, and the quads near the edge of the image are squeezed into a very small area. Because of this, for some quads in those areas, no pixel is sampled, and the information

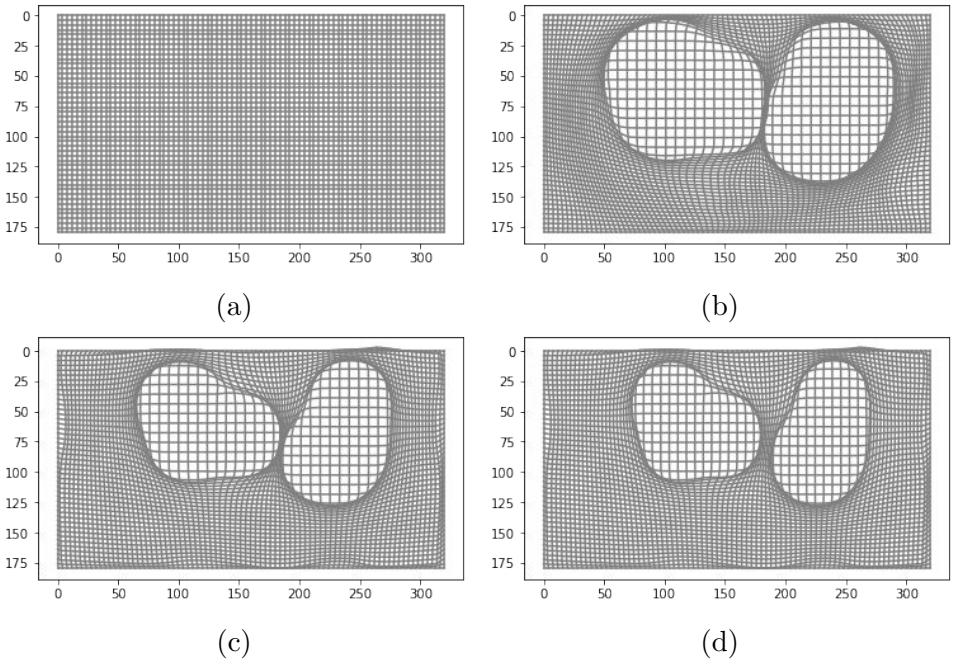

Figure 5.3: Transformed meshes under different constraints: a) original mesh, b) transformed mesh without constraints on non-salient quads, c) transformed mesh with smoothing weight scheme, d) transformed mesh with smoothing weight scheme and uniform constraint.

in those quads is totally lost. After applying the smoothing weight method, the extent of deformation on the boundaries of salient areas is reduced, as shown in Figure 5.3 (c). Finally, after applying the uniform constraint, the salient areas do not overlap anymore, and the deformation is further reduced, allowing the pixels in non-salient areas to be sampled more uniformly.

5.2.5 Effectiveness of Foveation in Reducing Redundancy

The foveation process reduces the total number of pixels that need to be encoded. We assume that this can reduce redundancy in the video frames. To verify this assumption, we calculate the average information entropy and total information entropy of all video frames in the dataset before and after applying foveation.

Information entropy measures the average level of information that a random variable contains [100]. For a discrete random variable X with n possible

values $x_1, x_2, ..., x_n$, the information entropy $H(f)$ is defined as Eq. 5.14, where p_i is the probability of x_i.

$$H(f) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{5.14}$$

The entropy can be seen as a lower bound on the average number of bits needed to encode a random variable. However, it is not trivial to extend Shannon's original information entropy to higher dimensions, such as images. We use delentropy to measure the average information entropy of an image in our experiment, as it compares favorably with the conventional intensity-based histogram entropy and the compressed data rates of a lossless image encoder [61]. The results are shown in Tables 5.1 and 5.2.

The results confirm our assumption. As shown in Table 5.1, the average entropy increases from 4.0907 bits per pixel (bpp) to 4.4182 bpp after the foveation process. According to [61], images with simple patterns like a pure black image has a lower average entropy than images with complex patterns, like a natural scene. Therefore, the increase in the average entropy indicates that the foveation process has removed some redundancy in the video frames. It also means that each pixel is carrying more meaningful information than before.

The total entropy for a frame decreases from $1.29E6$ bits to $6.95E5$ bits, which means that the total amount of information in the video frames has been reduced. Since this is the lower bound on the average number of bits needed to encode a frame without loss, it shows that our method can effectively reduce redundancy in the video frames, if the perceived quality remains similar. Thus, adding the foveation process can help increase the compression ratio.

Table 5.2 shows the results of all categories, and we can draw similar conclusions.

5.3 Experiments and Discussion

We test our compression algorithm on the UCF Sports dataset [78]. The UCF Sports dataset contains 150 videos in 12 categories. The resolution of these

Table 5.1: Average and total entropy of a frame in the original and warped videos.

Original Image Avg. Entropy	4.09 bpp
Warped Image Avg. Entropy	4.42 bpp
Original Image Total Entropy	1.29E6 bits
Warped Image Total Entropy	6.95E5 bits

Table 5.2: Overall average and total information entropy of the original and warped images.

Category	Original Image Avg. Entropy	Warped Image Avg. Entropy	Original Image Total Entropy	Warped Image Total Entropy
Diving	3.96	4.31	1.15E6	6.25E5
Golf	4.13	4.44	1.53E6	8.07E5
Kicking	3.30	3.70	1.23E6	6.81E5
Lifting	3.48	3.92	1.01E6	5.70E5
Riding	3.82	4.23	1.37E6	7.55E5
Run	3.63	4.03	1.26E6	7.02E5
SkateBoarding	4.74	4.92	8.19E5	4.23E5
Swing	4.76	5.00	1.38E6	7.25E5

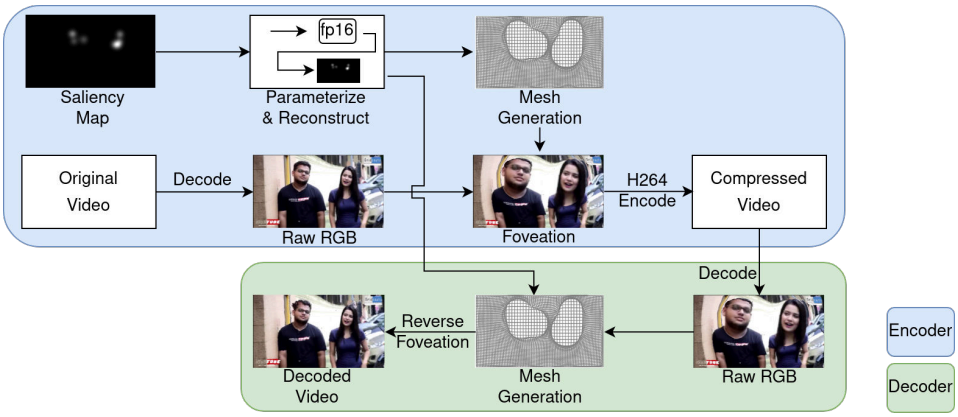

Figure 5.4: Test pipeline.

videos is 720×480. We implement the test pipeline using the Gstreamer framework, as shown in Figure 5.4. First, the input video sequence is decoded into a YUV sequence and then converted to the RGB color space. Then, the saliency map of the corresponding frame is read and decomposed into a combination of

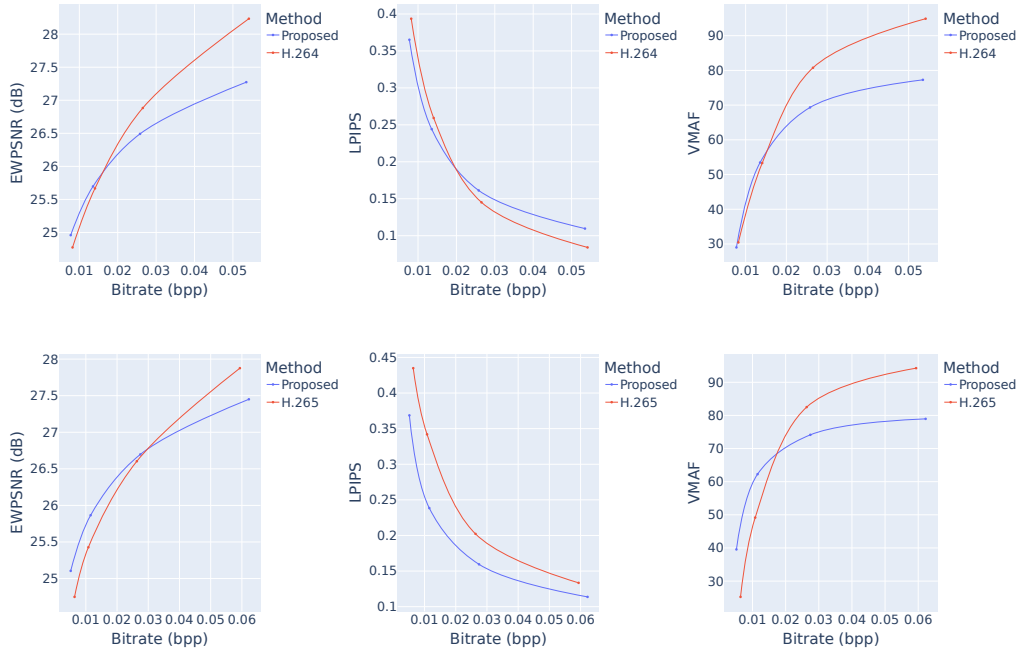

Figure 5.5: Overall rate-distortion curves of the proposed method compared with H.264 and H.265. We conducted two sets of comparisons. The results in the first row shows the comparison between the original H.264 and H.264 incorporating the proposed method. The results in the second row shows the comparison between the original H.265 and H.265 incorporating the proposed method. Three metrics are used for each set of comparison: EWPSNR, VMAF, and LPIPS. For EWPSNR and VMAF, higher is better. For LPIPS, lower is better.

several gaussian distributions. The parameters of these gaussian distributions are then saved for reproducing the saliency maps on the decoder end. To be specific, the parameters are converted to 16-bit floating-point numbers and saved as bytes in a raw text subtitle track using the MKV container. Next, the warped mesh is generated based on the reconstructed saliency map using the gaussian parameters. We then compute a pixel location mapping from the original frame to the compressed frame from the warp mesh parameters. Finally, we construct the compressed RGB frame using the mapping and encode the resulting frame using a video encoder, such as H.264 or H.265.

This process is repeated for all frames in a video. Assuming the eye movement is relatively small during a short interval of time, we only generate a

new warped mesh every 5 frames (about 167ms in a 30 frame/second video) to reduce the computational complexity.

The decoding process is the reverse of the encoding process. First, the compressed video is decoded by a video decoder to produce YUV frames, and then converted to the RBG color space. Then, the saved gaussian parameters are extracted from the raw text subtitle track in the MKV container and used to reconstruct the saliency map. Next, the saliency map is used to compute the warped mesh and a pixel location mapping from the compressed frame to the original frame. Finally, the original frame is reconstructed using the compressed frame and the mapping. Figs. 5.9 and 5.10 show some results from the proposed method, H.264, and H.265 with details magnified. It can be seen that the proposed method retains details in salient areas, while H.264 and H.265 produce blurry blocks and color artifacts.

For the subjective and objective image quality tests, when compared with H.264, the target bitrate settings are 0.054 bpp, 0.026 bpp, 0.014 bpp, and 0.008 bpp for the high, medium, low, and very low settings, respectively. When compared with H.265, the target bitrate settings are 0.06 bpp, 0.026 bpp, 0.01 bpp, and 0.006 bpp for the high, medium, low, and very low settings, respectively.

5.3.1 Subjective Image Quality Assessment

For comparison, we compress the original video sequences using both H.264 and the proposed method. We use four different quality settings for the proposed method, and use the x264 encoder to produce compressed videos that have the same bitrates. We conduct a subjective quality assessment using the double-stimulus impairment scale (DSIS) method specified in Recommendation ITU-R BT.500-14 [99]. A total of 20 videos are randomly selected from the UCF Sports dataset with at least one from each category.

Twelve test subjects are asked to compare a video produced by either the proposed method or x264, and give a response in the five-grade impairment scale. The mean scores and 95% confidence intervals (CI) for every quality setting are summarized in Table 5.3. The results show that the perceptual

video quality of our method is better than H.264 for all quality settings. This is because more bits are used to store information on the salient areas in our method compared to H.264.

Table 5.3: Subjective test results.

Bitrate	Method	Mean Score	95% CI
High 0.054bpp	Proposed	4.91	$(4.36, 5.47)$
	H.264	4.83	$(4.11, 5.56)$
Medium 0.022bpp	Proposed	4.67	$(3.71, 5.64)$
	H.264	4.04	$(2.47, 5.63)$
Low 0.011bpp	Proposed	3.90	$(2.58, 5.17)$
	H.264	2.70	$(0.97, 4.40)$
Very Low 0.006bpp	Proposed	2.88	$(1.38, 4.34)$
	H.264	1.75	$(0.02, 3.53)$

5.3.2 Objective Image Quality Assessment

The proposed method optimizes the perceptual video quality using saliency information. This introduces more distortion in non-salient areas than traditional video compression methods like H.264 and H.265. Video quality metrics based on signal processing techniques, such as the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM)[113], are not suitable for evaluating the perceptual video quality in this case because the relatively large distortion in non-salient areas causes a large decrease in overall PSNR and SSIM. These metrics might give results that do not align with human perception. Thus, we conduct an objective image quality assessment using perceptual quality metrics, specifically using the Eye-tracking Weighted PSNR (EWPSNR)[69], Perceptual Similarity (LPIPS)[125], and Video Multi-Method Assessment Fusion (VMAF)[92] metrics.

The EWPSNR metric is a perceptual objective quality metric incorporating saliency information when calculating the PSNR score.

The LPIPS metric uses deep features trained on supervised, self-supervised, and unsupervised objectives alike, to model low-level perceptual similarity. The results show that LPIPS can outperform traditional metrics like l_2 and SSIM.

VMAF is a perceptual video quality metric that tries to approximate human perception of video quality. It is formulated by Netflix to correlate strongly with subjective mean opinion scores using machine learning techniques.

Figure 5.5 shows the rate-distortion curves. These test results align with our subjective test results at medium and lower bitrates. Compared with H.264, the proposed method performs better at low and very low bitrate settings according to most metrics we use. Compared with H.265, the proposed method performs better at medium and lower bitrate settings. We calculate the BD-EWPSNR, BD-VMAF, and BD-LPIPS scores at medium and lower bitrates, as well as the corresponding BD-Rate values, for the proposed method and H.264/H.265 [16]. The results are shown in Table 6.4. Overall, at medium and lower bitrates, the proposed method achieves better perceptual video quality than either H.264 or H.265.

Table 5.4: BD-EWPSNR, BD-VMAF, and BD-LPIPS scores, as well as the corresponding BD-rate values. For BD-EWPSNR and BD-VMAF, a positive value x indicates that the proposed method can increase the performance by x at the same bitrate. For BD-LPIPS, a negative value $-x$ indicates that the proposed method can increase the performance by $-x$ at the same bitrate. For BD-rate, a negative value $-x$ indicates that the proposed method can achieve the same level of performance with a bitrate saving of $x\%$.

Method	H.264	H.265
BD-EWPSNR	0.028	0.309
BD-Rate	-3.354	-23.974
BD-LPIPS	-0.004	-0.083
BD-Rate	-6.537	-46.962
BD-VMAF	-3.505	7.434
BD-Rate	4.414	-27.268

We also observe that the scores vary for different video categories. We summarize all test results for different categories in Figs. 5.6, 5.7, and 5.8. It can be seen that in "Diving," "Riding," and "Run" categories, the proposed method is better on almost all four bitrate settings. In "Kicking," "Lifting," "Skate-Boarding," and "Swing" videos, the proposed method is better at medium and low settings. However, in "Golf" videos, H.264 and H.265 perform better on

all bitrate settings.

These results show that the proposed method is suitable for videos with fast motion. When watching such videos, people tend to focus only on the main subject in the videos and are less likely to notice the image quality degradation in the background. However, when watching videos with less motion, like videos from the "Golf" category, because the main subject cannot draw enough attention, people are more likely to notice the quality difference between salient and non-salient areas. Thus, the proposed method achieves better results in categories with fast motion.

One problem we notice in the objective test is that the EWPSNR and VMAF scores are relatively low for the proposed method at medium and high bitrates, and they do not align well with the LPIPS and subjective test scores. Our assumption is that the two full-reference image quality metrics are essentially based on the pixel-to-pixel difference between the original and compressed images. However, the proposed method might shift the pixels in the salient regions by a small distance from their original location, as shown in Figure 5.11, because of the warping transform process. This might cause the pixel-to-pixel difference between the original and compressed images to be larger than the actual difference perceived by humans. Thus, the proposed method might achieve better perceptual quality than the full-reference metrics suggest, as the subjective test result indicates.

5.3.3 Impact of Saliency Prediction Accuracy

The accuracy of saliency prediction is an important factor that affects the performance of the proposed method. Thus, we also conducted several additional experiments to study the impact of saliency prediction accuracy on the proposed method. We used the first frame of the Diving-Side-005 video as the input for these experiments. Saliency predictions of different accuracy were generated using the following procedure:

Step 1: Data on fixation points for this image frame is obtained from the dataset.

Step 2: We shift all the fixation points by a random distance between 0 and 50 pixels.

Step 3: 2D Gaussian distributions with standard deviations $\sigma_x = 20, \sigma_y = 20$ are placed on a grid with the center of each distribution being the shifted fixation points.

Step 4: The values in the grid are then normalized to have a minimum of 0 and a maximum of 255. This forms the saliency map.

Step 5: We calculate the Normalized Scanpath Saliency (NSS) score defined in Eq. 5.15 for the generated saliency map. If the NSS score is not within the desirable range, we repeat the process from Step 2.

$$\begin{aligned} NSS &= \frac{\sum^N T_i}{N} \\ \text{where } T &= \frac{S - \bar{S}}{\sigma_S} \circ F \end{aligned} \tag{5.15}$$

In Eq. 5.15, S is the saliency map, $\bar{S}$ is the mean of S, σ_S is the standard deviation of S, F is the fixation map with only 0 and 1 as pixel values in it, T_i is the pixel value in T at location i, and N is the total number of non-zero pixels in F. A higher NSS score indicates a higher saliency prediction accuracy.

The original image and the generated saliency maps are then used as the input for the proposed method. The target scales in the experiments are $s_x = 0.5$, and $s_y = 0.5$. We chose these aggressive scaling factors to make the proposed method more sensitive to the saliency prediction accuracy. This will also cause the EWPSNR scores to be relatively low.

Three saliency maps are used with NSS scores of 8.3277, 2.082, and 0.5184, respectively. A NSS score of 8.3277 is very high and the corresponding saliency map can be seen as the ground truth. A NSS score of 2.082 indicates a moderate saliency prediction accuracy. A NSS score of 0.5184 indicates a poor saliency prediction accuracy. We also experiment with three different saliency threshold (s_t) settings of 1, 50, and 100.

We summarize the results in Figure 5.12.

It can be seen that as the saliency accuracy decreases, the proposed method produces images with lower EWPSNR scores. This is expected because poor saliency prediction results in regions being labeled incorrectly as salient. Consequently, more bits will be assigned to non-salient regions and cause a quality decrease in the actual salient areas. However, the results demonstrate that this problem could be alleviated by using a lower saliency threshold s_t. A lower s_t will result in more regions being labeled as salient, and those regions have a chance to cover the actual salient regions. The resulting warped meshes of different saliency thresholds s_t are shown in Figure 5.13. Thus, as long as the saliency prediction accuracy is not too low, the proposed method can still produce images that preserve the quality of the actual salient regions.

5.3.4 Applications and Limitations

The proposed method can serve as a pre-processing step for any existing video compression method. Experimental results indicate that the proposed method is particularly effective when compressing videos with restricted bitrates, such as those streamed on mobile devices using cellular data. This is because the proposed method has a bigger advantage at medium and low bitrates. Additionally, the proposed method shows potential for compressing videos featuring fast motion, such as sports videos. In this case, peripheral details receive less attention, making the proposed method highly suitable.

The proposed method could be suitable for video compression in survaillance systems and drone applications. In survaillance systems, most of the time the video frames will be almost identical, and only a small portion of the video will contain important information. The survaillance videos look like the "boats" video in our experiments and we expect the proposed method to perform well in this case. In drone applications, the proposed method can be used to compress the live feed before transmitting it to the remote controller. This helps prioritize interesting features, decrease the bandwidth usage, and reduce the response latency.

The proposed method has two limitations. First, it may introduce minor pixel displacements in salient regions compared to their original locations.

This might result in a decrease in PSNR scores. However, it is not expected to significantly impact the perceptual quality of the video. Second, the proposed method may not perform optimally for videos with slow motion, as quality degradation in the background is more likely to be perceptible in such instances.

5.4 Conclusion

We presented a novel approach to video compression, taking into account the characteristics of the human visual system and leveraging foveation to allocate bits to different regions in a video based on their visual saliency. This was achieved through a feature-aware image warping technique that preserves the image quality in salient areas. One of the main advantages of the proposed method is that it can be easily integrated with existing video compression standards without requiring modifications to the bit stream format. Our subjective evaluations show that the proposed method outperforms H.264 and H.265 in terms of perceptual quality, and objective evaluations confirm these findings at medium and low bitrates. These results suggest that our approach has the potential to improve the compression ratio while maintaining the perceptual quality.

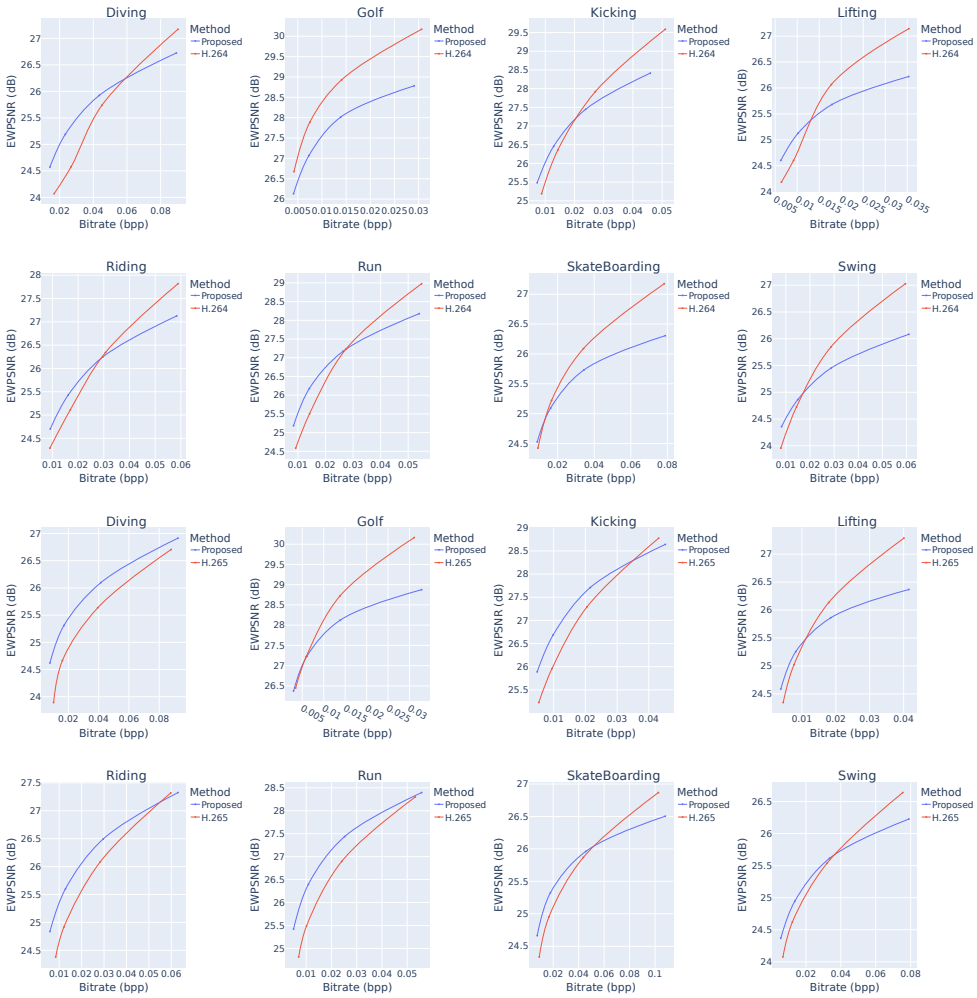

Figure 5.6: Rate-EWPSNR curves of the proposed method compared with H.264 and H.265 on different categories. The first two rows show the comparison with H.264, and the last two rows show the comparison with H.265. Higher is better.

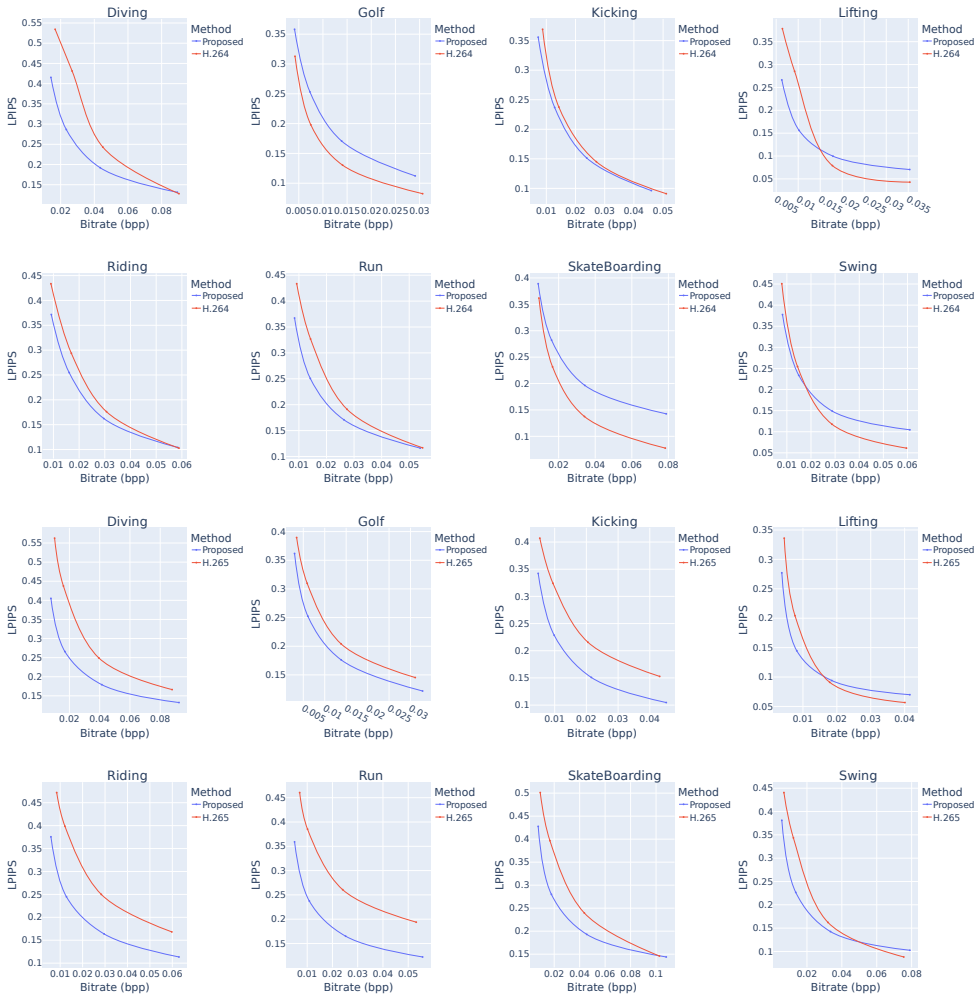

Figure 5.7: Rate-LPIPS curves of the proposed method compared with H.264 and H.265 on different categories. The first two rows show the comparison with H.264, and the last two rows show the comparison with H.265. Lower is better.

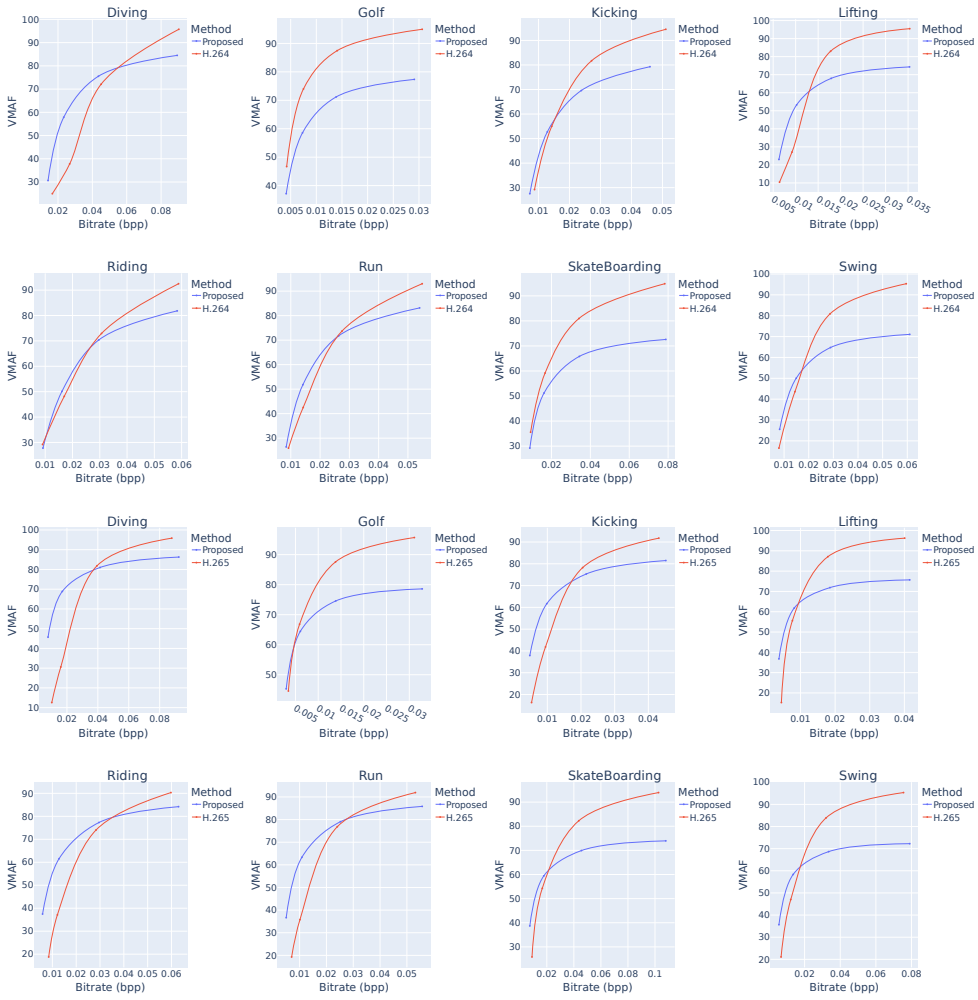

Figure 5.8: Rate-VMAF curves of the proposed method compared with H.264 and H.265 on different categories. The first two rows show the comparison with H.264, and the last two rows show the comparison with H.265. Higher is better.

Figure 5.9: Video quality comparison with H.264. The first and third rows show results produced by the proposed method. The second and fourth rows show results produced by H.264. The results are best viewed in color.

Figure 5.10: Video quality comparison with H.265. The first and third rows show results produced by the proposed method. The second and fourth rows show results produced by H.265. The results are best viewed in color.

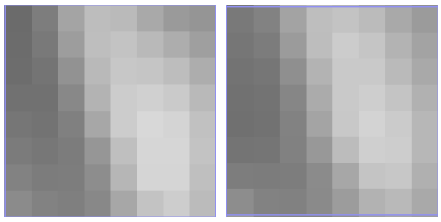

Figure 5.11: The same 8×8 image block from the original video (left) and the video compressed by the proposed method (right). The pixels in the salient regions are shifted up by about one pixel from their original location.

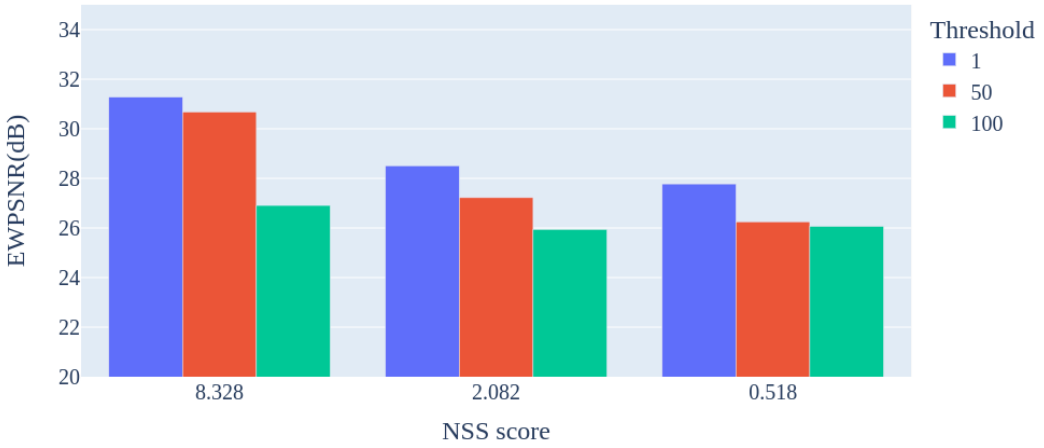

Figure 5.12: EWPSNR scores of the reconstructed images with different saliency prediction accuracy and saliency threshold s_t settings.

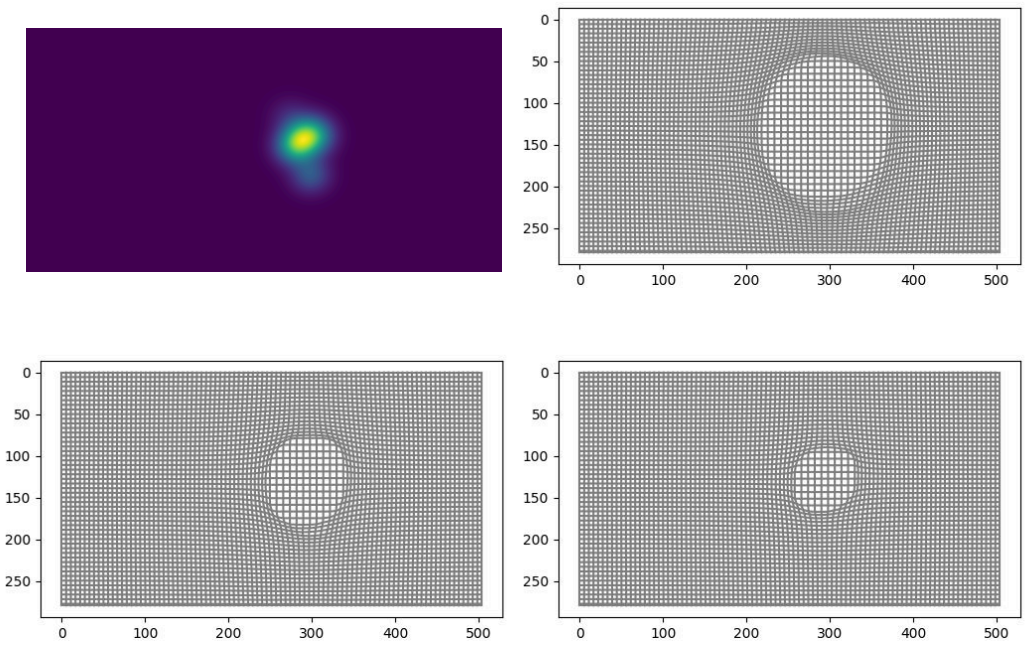

Figure 5.13: The warped meshes for different saliency threshold (s_t) settings are shown. The top left image shows the saliency map with a NSS score of 8.3277. The top right image shows the mesh with $s_t = 1$. The bottom left image shows the mesh with $s_t = 50$. The bottom right image shows the mesh with $s_t = 100$.

Chapter 6

Principal Component Approximation Network for Image Compression

6.1 Introduction

Lossy image compression is a fundamental research topic in image processing. Recently, several sophisticated approaches based on deep learning networks have achieved excellent performance in image compression [2]–[4], [10]–[12], [18], [23], [24], [36]–[39], [43], [44], [53], [54], [62], [63], [65], [68], [79], [80], [82], [86], [95], [96], [102], [103], [106], [107], [110], [114], [120], [121], [127], [130], [132]. However, these methods usually do not include the size of network parameters in bitrate computation. This is because these models consider themselves as general image compression models that can be applied to compress any image. Therefore, the models only need to be transmitted once, and used for all subsequent image compression. However, like many other machine learning models, the performance of these networks might be impacted if the image being compressed is dissimilar to their training images. In addition, we noticed that for the same method, bigger networks tend to provide better compression results. This observation contradicts the assumption that these models are general since the performance of a general model should not be impacted by changing the size of the model, and indicates that some level of memorization still exists in these models, which means there might be redundancies inside these networks. Thus, we propose a novel method, the principal

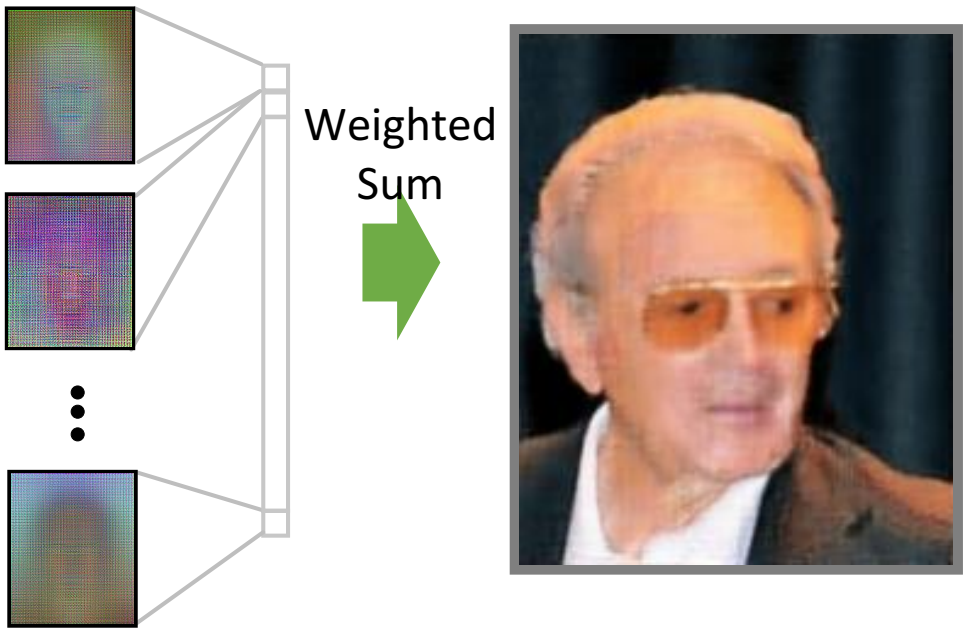

Figure 6.1: Illustration of the proposed principal component approximation network. An image is reconstructed by the weighted sum of several shared feature matrices, and the weight vector can be used as the coding vector for image compression.

component approximation network (PCANet), that reduces redundancies not only in the coding vectors but also in the network parameters. We account for the lack of ability to generalize to all images by including the size of the model parameters into the bitrate computation.

The proposed approach treats image compression as a decomposition problem. The intuition behind this is that common features exist in different images, and images can be represented by a limited number of these features. This is similar to how images are decomposed into different frequency components in traditional image compression methods [25], [108]. Thus, the proposed method decomposes images into several shared feature matrices that resemble common image features. Then, the original images can be reconstructed using the weighted sum of these feature matrices, as shown in Figure 6.1. We regard these feature matrices as the "principal" components, and the proposed approach focuses on learning and approximating these feature matrices using

a network.

The architecture of the proposed network is quite straightforward. First, the vertical and horizontal vectors are multiplied to generate a limited number of basis matrices whose rows are linearly related. Then these basis matrices are linearly transformed into several feature matrices, in which the linear transformation is approximated using a fully connected layer. Finally, the images can be reconstructed by the weighted sum of the feature matrices using weight vectors, which are generated during the training process. During the training of the proposed network, both the network parameters and the input vectors are updated. After completion of the training process, the updated input vectors are used as the weight vectors for the reconstructed images. Details of our approach are discussed in Section 6.2.

The proposed approach tries to reduce redundancy by utilizing two techniques. First, by using a fixed number of basis feature matrices and minimizing reconstruction errors, the model is forced to learn the most representative features among all the images. This technique reduces the redundancy in the model itself as the number of training images increases, since only the most representative features are retained. Second, the weight vectors — which contain the weights of each feature matrix for the images — are generated by flattening covariance matrices of a limited number of independent variables. The weight vectors are the final encoded results of the images. By representing them as covariances of independent variables, only a few independent variables need to be stored for each image. This technique reduces redundancy in the coding results/vectors. Moreover, in order to improve the reconstruction quality, images at different scales are reconstructed independently and the sum of the images at different scales is used as the final reconstruction result.

The proposed method differs from existing image compression methods in several aspects. First, it learns the most representative features for decomposition, while traditional image compression methods use fixed hand-crafted transforms. This allows it to adapt to different images and achieve better compression results. Second, the proposed method is the first learning-based method to consider image compression as a decomposition problem. Third,

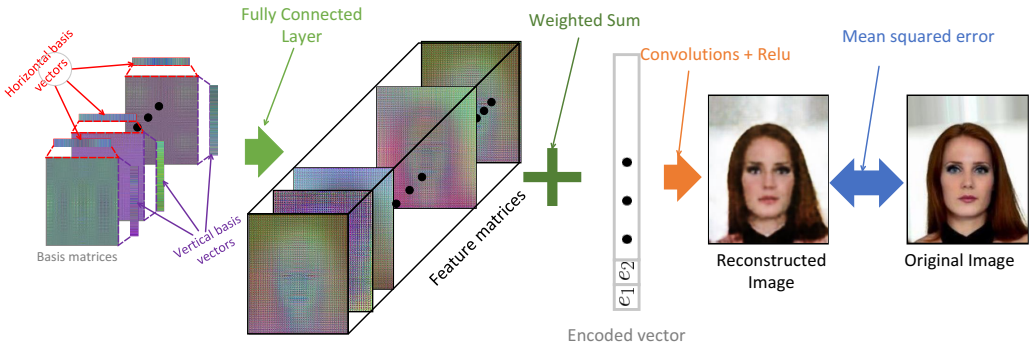

Figure 6.2: The main architecture of the proposed PCANet focusing on learning a series of shared feature matrices for image decomposition.

the proposed method not only reduces redundancy in the coding vectors, but also reduces redundancy in the model itself. This enables the proposed method to be the first to include the size of the model parameters in the bitrate computation and still achieve promising results.

The main contributions of this chapter are:

1. We propose the principal component approximation network to learn shared feature matrices for image compression. The network parameters are used to approximate these shared matrices, thereby reducing information redundancy inside the proposed network. Therefore, the proposed approach achieves promising compression results even after taking into account the size of network parameters.

2. The size of the proposed network is relatively small, containing only around 4 million trainable parameters. The architecture is very straightforward and explainable.

3. Comprehensive experiments based on several standard datasets demonstrate the effectiveness of the proposed approach.

4. A new metric is proposed to evaluate the information redundancy inside the models.

6.2 Principal Component Approximation Network

The proposed approach is based on the assumption that a group of images $\{I_n | n \in [1, N]\}$ can be decomposed into a number of shared feature matrices $\vec{\mathbf{F}}^T = [F_1, F_2, \cdots, F_L]^T$ which are linearly transformed from a limited number of basis matrices $\vec{\mathbf{B}}^T = \{B_1, B_2, \cdots, B_K\}^T$. Moreover, vectors $\{\vec{e}_n | n \in [1, N]\}$ representing the weights of matrices can be used to encode any of the images in this group. By reducing the number of feature matrices, the length of coding vectors can also be reduced, which demonstrates great potential for lossy image compression. This process can be mathematically expressed as Eqn. 6.1:

$$I_n = \vec{e}_n \cdot \vec{\mathbf{F}}^T + \epsilon_n \tag{6.1}$$

where ϵ_n represents the errors between the reconstructed and original images, and n is the index of the image. The main purpose of the proposed approach is minimizing the total errors between the reconstructed and original images with a constraint on the number of feature matrices. Mathematically, this can be shown as:

$$\begin{aligned} &\text{minimize} \quad \sum_{n=1}^{N} \epsilon_n^2 = (I_n - \vec{e}_n \cdot \vec{\mathbf{F}}^T)^2 \\ &\text{subject to} \quad ||\vec{e}_n||_0 = ||\vec{\mathbf{F}}||_0 \leq L \end{aligned} \tag{6.2}$$

where $||\vec{e}_n||_0$ is the length of a coding vector, $||\vec{\mathbf{F}}||_0$ is the number of feature matrices, and L is the maximum value of $||\vec{\mathbf{F}}||_0$. In Eqns. 6.3, 6.5, and 6.6, we show that it is possible to use a neural network to approximate $\vec{\mathbf{F}}^T$ by deriving from singular value decomposition (SVD).

First, we consider that images can be decomposed by SVD into several basis feature matrices with their corresponding weights. Mathematically, this is represented as:

$$I_n = U_n S_n V_n^T = \sum_{i=1}^{M} \sigma_i (\vec{u}_{i,n} \cdot \vec{v}_{i,n}^T) = \sum_{i=1}^{M} \sigma_{i,n} B_{i,n} \tag{6.3}$$

where $\sigma_{i,n}$ denotes the singular values of the image I_n; $B_{i,n}$ is the basis matrix computed from the multiplication of singular vectors $\vec{u}_{i,n}$ and $\vec{v}_{i,n}$; M is the

number of singular values for the basis matrices. $\vec{u}_{i,n}$ is the i-th column vector of U_n and $\vec{v}_{i,n}$ is the i-th column vector of V_n. $\vec{u}_{i,n} \cdot \vec{v}_{i,n}^T$ gives the i-th basis matrix $B_{i,n}$. The singular values $\sigma_{i,n}$ are weights associated with the basis matrices $B_{i,n}$, and the weighted sum of the basis matrices can be used to reconstruct the original image I_n.

The basis matrices of different images are independent. Thus, we introduce a latent variable $a_{n,j} \in \{0, 1\}$ to correlate these independent basis matrices.

$$a_{n,j} = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

The decomposition process can be mathematically expressed as:

$$\begin{aligned} I_n &= U_n S_n V_n^T \\ &= \sum_{i=1}^{M} \sigma_{i,n} \cdot B_{i,n} \\ &= 0 \cdot \sum_{i=1}^{M} \sigma_{i,1} \cdot B_{i,1} + \cdots 1 \cdot \sum_{i=1}^{M} \sigma_{i,n} \cdot B_{i,n} + \cdots \\ &= a_{n,1} \cdot \sum_{i=1}^{M} \sigma_{i,1} \cdot B_{i,1} + \cdots a_{n,n} \cdot \sum_{i=1}^{M} \sigma_{i,n} \cdot B_{i,n} + \cdots \\ &= \sum_{j=1}^{N} a_{n,j} \cdot \sum_{i=1}^{M} \sigma_{i,j} \cdot B_{i,j} = \sum_{j=1}^{N} \sum_{i=1}^{M} \underbrace{a_{n,j} \cdot \sigma_{i,j}}_{w_{n,k}} \cdot B_{i,j} \\ &= \sum_{k=1}^{K} \omega_{n,k} \cdot B_k = \vec{\omega}_n \cdot \vec{\mathbf{B}}^T, \quad ||\vec{\omega_n}||_0 = K = M \cdot N \end{aligned} \tag{6.5}$$

where $w_{n,k} = a_{n,j} \cdot \sigma_{i,j}$ can be used as the coding vectors, $\vec{\mathbf{B}}^T = [B_1, B_2, \cdots B_K]^T$ is the vector containing all the shared matrices, and N is the number of images. In this equation, we expand each term in Eqn. 6.3 into the sum of N terms, which correspond to the singular values and feature matrices of the N images. Each term in the summation is multiplied by a latent variable $a_{n,j}$, which is equal to 1 only when $i = j$ to indicate the selection of a set of singular values and feature matrices. This allows us to further simplify this equation to a matrix multiplication form.

As shown in Eqn. 6.5, it is possible to find a series of shared matrices which can be used to reconstruct all images in the training set using a weighted sum.

However, the length of the weight vectors is very long because the number of matrices is large. Retaining all the matrices and the corresponding weights is not practical for image compression since this will impact the compression ratio. Inspired by how low-frequency components and high-frequency components are treated differently in lossy image compression methods, we made an assumption that the shared feature matrices might not be equally important for image reconstruction. Therefore, we decided that we only need to keep the most important matrices and the corresponding weights. To achieve this, we apply a linear transformation to these basis matrices, then lossy reconstruction can be done with a limited number of transformed matrices that are the feature matrices describing the common parts of different images. This process can be described as:

$$
\begin{aligned}
I_n = \vec{\omega}_n \cdot \vec{\mathbf{B}}^T &= \vec{\omega}_n \cdot \overbrace{(\Psi \cdot \Phi)}^{\Psi \cdot \Phi = I = \mathrm{diag}(1,1,\cdots,1)} \cdot \vec{\mathbf{B}}^T \\
&= \vec{\omega}_n \cdot [\Psi_\alpha \quad \Psi_\beta] \cdot \begin{bmatrix} \Phi_\alpha \\ \Phi_\beta \end{bmatrix} \cdot \vec{\mathbf{B}}^T \\
&= [\vec{\omega}_n \cdot \Psi_\alpha \qquad \vec{\omega}_n \cdot \Psi_\beta] \begin{bmatrix} \Phi_\alpha \cdot \vec{\mathbf{B}}^T \\ \Phi_\beta \cdot \vec{\mathbf{B}}^T \end{bmatrix} \\
&= \underbrace{(\vec{\omega}_n \Psi_\alpha)}_{\vec{e}_n} \cdot \underbrace{(\Phi_\alpha \vec{\mathbf{B}}^T)}_{\vec{\mathbf{F}}^T} + \underbrace{\vec{\omega}_n \Psi_\beta \Phi_\beta \vec{\mathbf{B}}^T}_{\epsilon} \\
&= \vec{e} \cdot \vec{\mathbf{F}}^T + \epsilon_n = \vec{e}_n \cdot (\Phi_\alpha \vec{\mathbf{B}}^T) + \epsilon_n,
\end{aligned}
\tag{6.6}
$$

where $\Psi = \begin{bmatrix} \Psi_\alpha & \Psi_\beta \end{bmatrix}$,$\Phi = \begin{bmatrix} \Phi_\alpha & \Phi_\beta \end{bmatrix}^T$ and $\Psi \cdot \Phi = I$ is an identity matrix. Φ_α is a linear transformation, which can be achieved using a fully connected layer. In this equation, we introduce two matrices Ψ and Φ, which are used as linear transformation matrices applied to $\vec{\omega}_n$ and $\vec{\mathbf{B}}^T$, respectively. To separate the most important components of $\vec{\omega}_n$ and $\vec{\mathbf{B}}^T$, we further separate the columns of Ψ into Ψ_α and Ψ_β, and rows of Φ into Φ_α and Φ_β. Ψ_α and Φ_α correspond to the most important components of $\vec{\omega}_n$ and $\vec{\mathbf{B}}^T$, while Ψ_β and Φ_β correspond to the less important components. We gather all the less important components into ϵ_n and treat them as errors. Eqn. 6.6 shows that a good reconstruction of a set of images can be achieved by finding a set of weight vectors $\vec{\omega}_n$ and shared feature matrices $\vec{\mathbf{F}}^T$ to minimize the error term ϵ_n. Since Ψ_α and Φ_α are linear transformations, they can be approximated using a fully connected

layer.

Based on Eqn. 6.6, the architecture of the proposed principal approximation network is shown in Figure 6.2. The architecture is very straightforward. Multiplications of several basis vectors are used to generate the vector of basis matrices $\vec{\mathbf{B}}^T$, which are then used as the input of a fully connected layer to generate the feature matrices. The fully connected layer is an approximation of the linear transformation Φ_α. Finally, image I_n is reconstructed with these feature matrices and the coding vectors $\vec{e}_n$. During the training of the proposed network, we try to minimize the squared errors ϵ_n^2, which is mathematically shown below:

$$\begin{aligned} &\min \sum_{n=1}^{N} \epsilon_n^2 = (I_n - \vec{e}_n \cdot \vec{\mathbf{F}}^T)^2 = (I_n - \vec{e}_n \cdot \Phi_\alpha \vec{\mathbf{B}}^T)^2 \\ &\Rightarrow f(\vec{e}_n, \Phi_\alpha, \vec{\mathbf{B}}) = \min \sum_{n=1}^{N} (I_n - \vec{e}_n \cdot \Phi_\alpha \vec{\mathbf{B}}^T)^2 \end{aligned} \tag{6.7}$$

where Φ_α and $\vec{\mathbf{B}}^T$ are trainable parameters of the proposed PCANet. In particular, Φ_α represents the parameters of a fully connected layer, $\vec{\mathbf{B}}^T$ is the basis matrix, and $\vec{e}_n$ is the coding vector. In order to further reduce the information redundancy in the coding vectors, the coding vectors are generated by flattening a covariance matrix of a few independent variables, which is mathematically shown as follows:

$$\vec{e}_n = \mathcal{F}(\vec{c}_n^T \cdot \vec{c}_n), \quad \vec{c}_n = [c_{n,1}, c_{n,2} \cdots c_{n,J}]^T, \tag{6.8}$$

where $J^2 = ||\vec{c}_n||_0^2 = ||\vec{e}_n||_0$. $\mathcal{F}$ is the flattening function. J is the final number of variables used to encode an image, and every image is represented by $\vec{c}_n$ which is a vector of the size $1 \times J$. The setting of $J = 240$ is used for all experiments in this paper.

Network Training and Image Encoding: Unlike previous approaches that require that the encoder and decoder be trained separately, in the proposed method one single network functions as both the encoder and decoder. Every training image I_n is assigned a randomly initialized coding vector $\vec{c}_n$ before the training process. The reconstructed images, I'_n, are generated using

these coding vectors and the feature matrices, then the errors between I'_n and I_n are computed and back-propagated. During the back-propagation, both the coding vectors $\vec{c}_n$ and the network parameters are updated to minimize the sum of errors between I'_n and I_n. When the errors are reduced to an acceptable range after training, $\vec{c}_n$ can be directly used as the coding vector of image I_n for a lossy reconstruction.

Additionally, we demonstrate that when a large number of images are used for training, the proposed network can be used for encoding unseen images. Details about this are discussed in the last paragraph of Section 6.3.1.

Bitrate Computation: One concern may be that the proposed network is specific to the training images, which means that the network only works for images from the training set. It may therefore be seen as unfair to compare it with previous networks which are considered to be general methods for all images. Thus, we include the size of the proposed network into the bpp computation for comparisons with state-of-the-art methods, as shown in Section 6.3.1. In this way, the comparison is fair because we consider the parameters of the network as part of the encoded results while the other methods do not.

After training the network parameters, these parameters are quantized and encoded using arithmetic coding to further reduce the information redundancy inside the proposed network. In the bitrate computation, the size of the quantized parameters is evenly distributed among all the images in the training set. The number of bytes used to encode an image is calculated as the sum of the size of the coding vector $\vec{c}_n$ and the shared parameters. In addition, as the number of training images approaches ∞, the size of the shared parameters can be ignored, which can be mathematically expressed as:

$$\mathcal{C}(I_n) = \lim_{N\to\infty}(||\vec{c}_n||_0 + \frac{||\Phi_\alpha||_0 + ||\vec{\mathbf{B}}||_0}{N}) = ||\vec{c_n}||_0 + 0 \tag{6.9}$$

where $||\Phi_\alpha||_0 + ||\vec{\mathbf{B}}||_0$ denotes the size of the proposed network, and $||\vec{c}_n||_0 = 240$ is the size of the coding vectors. In the evaluation experiments, the maximum number of training images is 20,480. Given this number, every image shares less than 200 bytes of parameters, which is smaller than the coding vector $\vec{c}_n$ of size 1×240. The proposed approach achieves promising results even when

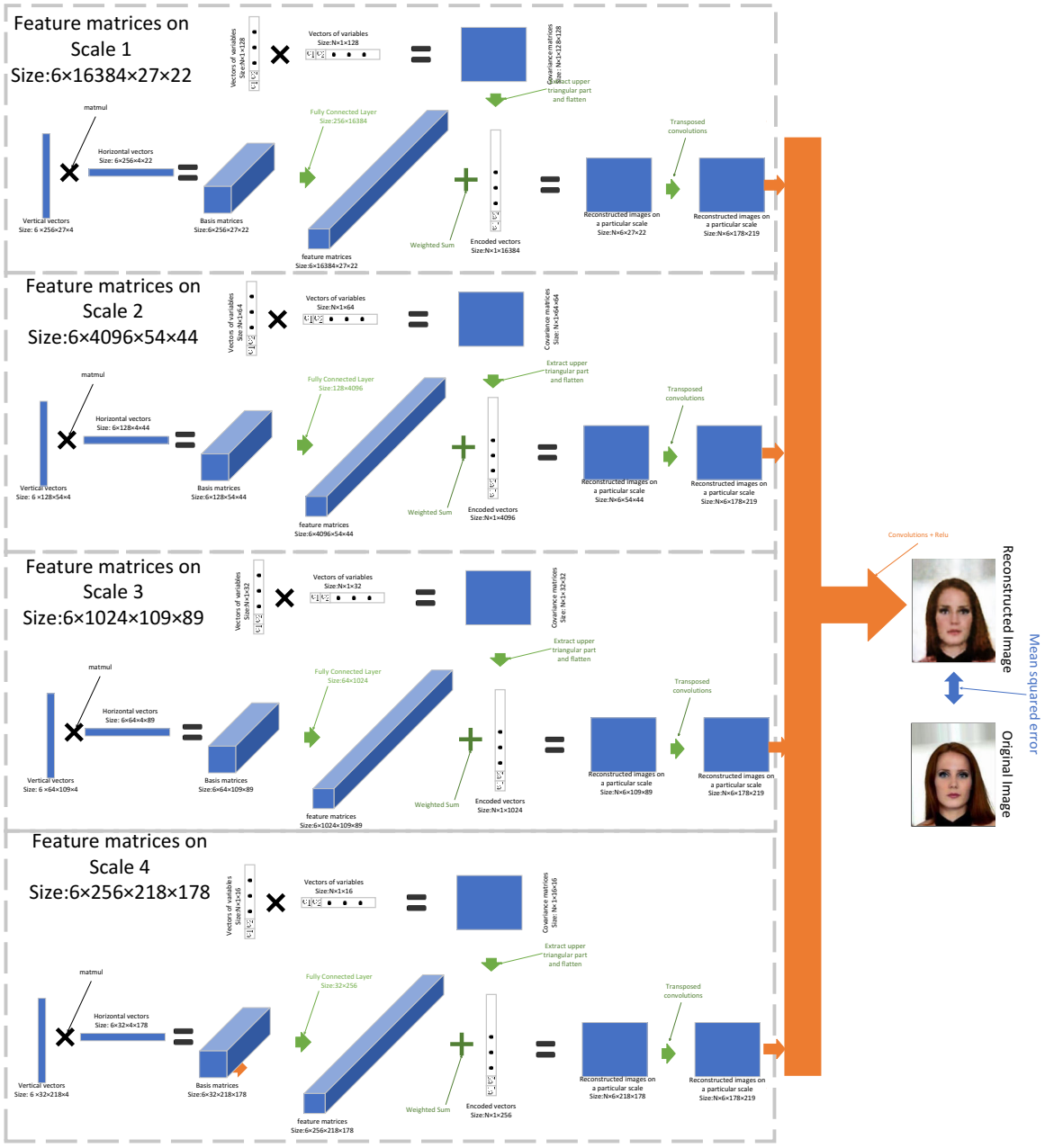

Figure 6.3: Detailed architecture of the proposed PCANet. This example shows the size of intermediate tensor shapes for four different paths when the input image shape is 178. Results from different paths are up-sampled and summed to produce the final reconstruction result.

compared with BPG, which is one of the best non-deep learning compression methods.

Multi-scale Strategy: We use a multi-scale strategy to improve the qual-

ity of image reconstruction. There are multiple paths in the proposed network, and each path is used to reconstruct images at a different scale. One path produces images at the original scale, and the other paths produce down-sampled images. The results from the different scales are then up-sampled by a few transposed convolution layers to produce images at the original scale. The final reconstruction result is the sum of the images at different scales. A detailed architecture of the proposed PCANet is illustrated in Figure 6.3. The process of training and inferencing on this architecture is explained below.

Algorithm overview: The detailed training process is as follows: first we initialize the parameters c for generation of coding vectors and b for generation of basis matrices. Each training image corresponds to one coding vector, which is generated by a row of c. Each row of b generates one basis matrix for each n scale paths. Then, each row of c is separated into n parts $c_i = [c_{i,0} \;\; c_{i,1} \cdots \;\; c_{i,n}]$ that correspond to n scale paths used in the network. Each part is used to generate the initial coding vectors for the corresponding scale path: $e_{i,j} = c_{i,j}^T \cdot c_{i,j}$. The final coding vectors are the concatenation of all the flattened $e_{i,j}$: $e_i = [\mathcal{F}(e_{i,0}) \;\; \cdots \;\; \mathcal{F}(e_{i,n})]$. Next, the basis matrices are generated in a similar way. Each row of b is separated into n parts $b_i = [b_{i,0} \;\; b_{i,1} \cdots \;\; b_{i,n}]$ that correspond to n scale paths used in the network. Each part is used to generate the initial basis matrix for the corresponding scale path: $B_{i,j}^T = b_{i,j}^T \cdot b_{i,j}$. Then, we load one batch of training images I from the dataset and start the forward pass. The feature matrices F^T are generated by passing the basis matrices B^T through a fully connected layer. This is equivalent to a linear transformation, which approximates Φ_α. Next, the results from different scale paths are up-sampled using different transposed convolutions. The up-sampled weighted sums have the same size as the original images. The final reconstruction result I' is the summation of all up-sampled weighted sums. Finally, the backward pass is carried out by first calculating the reconstruction loss: $L_{batch} = \frac{1}{N_b} \sum_{k=1}^{N_b} L(I_k - I'_k)$, where L is the loss function. Then, the parameters c and b are updated using gradient descent. This process is illustrated in Algorithm 2. After training is finished,

Algorithm 2: Training/Encoding procedure

1. $c, b \leftarrow$Initialize parameters;

while *convergence not reached* **do**

2. $\vec{e}_n \leftarrow \mathcal{F}(\vec{c}_n^T \cdot \vec{c}_n)$, where $\vec{c}_n = [c_{n,1}, c_{n,2} \cdots c_{n,J_1}]^T$, and $\vec{c} \in \mathbf{R}^{(N,J_1)}$;
3. $\vec{B}_n^T \leftarrow \vec{b}_n^T \cdot \vec{b}_n$, where $b_n = [b_{n,0} \;\; b_{n,1} \cdots \;\; b_{n,J_2}]$ and $\vec{b} \in \mathbf{R}^{(M,J_2)}$;
4. $\vec{I} \leftarrow$ load one batch of images from the dataset;

Stage 1: Forward pass

5. $\vec{F}^T \leftarrow \Phi_\alpha \cdot \vec{B}^T$, where Φ_α is approximated by a fully connected layer;
6. $\vec{S}_n \leftarrow \vec{e}_n \cdot \vec{F}_n^T$;
7. $\vec{I'} \leftarrow \sum C_t(\vec{S}_n)$, where C_t indicates a transposed convolution operation;

Stage 2: Backward pass

8. $L_{batch} \leftarrow \frac{1}{N_b} \sum_{k=1}^{N_b} L(I_k - I'_k)$, where L is the loss function;
9. $c \leftarrow c - \eta \nabla L(c, b)$, $b \leftarrow b - \eta \nabla L(c, b)$, where η is the learning rate;

end

each training image has a corresponding coding vector c_i. b is shared by all images.

To decode a set of images, we first load the trained parameters b and the coding vectors c for the images to be decoded. The decoding process is the same as the forward pass discussed in Algorithm 2.

6.3 Experiments

Experimental design: The proposed approach was evaluated on around 50,000 images from four different datasets including the Kodak, COCO [71], CelebA [73], and CDNet [6] datasets. The Kodak dataset is a widely used dataset for evaluating image compression algorithms. The COCO dataset is a large dataset containing images of various categories. The CelebA dataset contains images of human faces, and the Boat dataset contains consecutive frames from the video "boats" from the CDNet dataset. We chose the Kodak and COCO datasets to evaluate the performance of the proposed method on general images. Additionally, we used the CelebA and Boat datasets to

demonstrate that the proposed method is effective in decomposing images into common feature matrices.

For each dataset, we train a separate model for the proposed method. Then, we evaluate the quality of the reconstructed images using the peak signal-to-noise-ratio (PSNR) metric and the bitrate using the bits per pixel (bpp) metric. Since separate models are needed for each dataset, the size of the network parameters is included in the bitrate computation. The results are shown in Figure 6.4, Tables 6.1, 6.2, 6.3, and Figure 6.8, respectively.

The compared methods include those based on deep learning, such as CVPR22 [110], ICLR19 [63], CVPR18 [79], CVPR17 [107], ICASSP20 [19], TPAMI21 [44], and non-deep learning methods including JPEG [108], JPEG 2000 [25] and BPG. The results of the compared methods are generated by the source code and pre-trained models provided by the authors. Since these methods are assumed to be general, the size of the network parameters are not included in the bitrate computation. We discuss why this assumption might not be true in the "PRNR-pb" subsection and propose a new metric to measure the information redundancy inside network parameters.

At the end of this section, we include the BD-PSNR and BD-rate comparison of the proposed method and the best performing method, TPAMI21 [16], [44]. The BD-PSNR calculates the average PSNR difference between two rate-distortion curves. The BD-rate calculates the average bitrate difference between two rate-distortion curves. The BD-PSNR score shows a PSNR gain of the proposed method over TPAMI21 at the same bitrate, and the BD-rate score shows a bitrate reduction of the proposed method over TPAMI21 at the same PSNR score.

Training details: The proposed PCANet is implemented using PyTorch. During training, the learning rate is set to 0.0001 and the Adam optimizer with default parameters is used. Most of the experiments, including evaluation on Kodak, COCO, and CelebA, are run on a Nvidia GTX 1080 GPU with 8GB of VRAM. The experiment on video "boats" is run on a Nvidia RTX 3090 GPU with 24GB of VRAM.

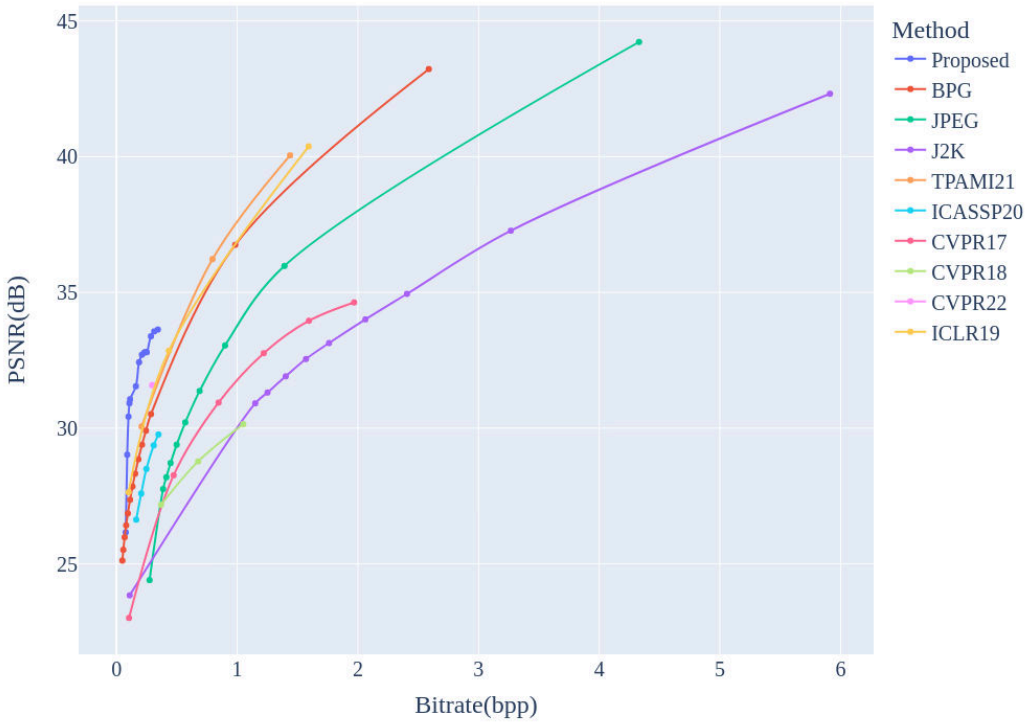

Figure 6.4: The comparisons between the proposed PCANet and state-of-the methods on Kodak dataset.

6.3.1 Evaluation of the proposed PCANet

Experiments on Kodak: The Kodak dataset is one of the most commonly used datasets used to evaluate image compression algorithms. Thus, the proposed approach is first evaluated on the Kodak dataset. During the evaluations, images are divided into patches of resolution 128×128, to reduce the memory required to run the training. The network parameters of the proposed network and previous networks are not yet included into the bitrate computation. The comparisons are shown in Figure 6.4, where a curve close to the top left corner indicates better performance. It can be seen from the graph that the proposed method outperforms all other compared methods in terms of PSNR. This means the proposed method offers better image quality at the same bitrate compared to other methods. We also compare the performance of the proposed method and the state-of-the-art method, TPAMI2021, using

the BD-PSNR and BD-rate [16] metrics. The results are shown in Table 6.4. As indicated, the proposed method can increase the PSNR by 1.8826 dB at the same bitrate, or reduce the bitrate by -48.290% for the same PSNR score, compared to TPAMI2021.

PRNR-pb: However, one problem with this evaluation method is that the size of the network parameters is not considered in the bitrate computation. Unlike deep learning-based methods, traditional image compression methods do not need any extra network parameters to encode and decode images. Therefore, the size of the network should be included in the bitrate computation, unless there is evidence that the prior information learned by networks is general for all images. Otherwise, the information redundancy might simply be moved from the coding vectors into the network parameters. Sometimes the size of the network parameters can be much larger than the size of the dataset on which the models are evaluated. For example, there are only 24 images in the Kodak dataset. Hence, the total size of the data for the Kodak dataset is $24 \times 512 \times 768 \times 3/(2^{20}) \approx 27$MB, which is much smaller than the size of a few networks evaluated on this dataset.

The best way to address this concern is to evaluate deep learning-based compression methods on a large number of images, with the size of the networks included in the bitrate computation. Thus, we propose a new metric called the Peak Signal-to-Noise Ratio per byte (PSNR-pb) to measure the information redundancy inside the network parameters. Specifically, PSNR-pb measures the contribution of every byte of trainable parameters to reduce the mean squared errors between reconstructed and original images; it also measures the quality of the reconstructed images. This can be mathematically expressed as:

$$\begin{aligned}\text{PSNR-pb} =& 10 \cdot \log\left(\frac{\text{MAX}^2}{\text{MSE}} \cdot \frac{1}{||\mathcal{G}||_0^N}\right) \\ =& \text{PSNR} - 10N \cdot \log(||\mathcal{G}||_0)\end{aligned}$$

where MSE is the mean squared error between the reconstructed and original images; MAX denotes the maximum possible value of a pixel which is 255 for an 8-bit image; N is a constant value used to generate results in a reasonable

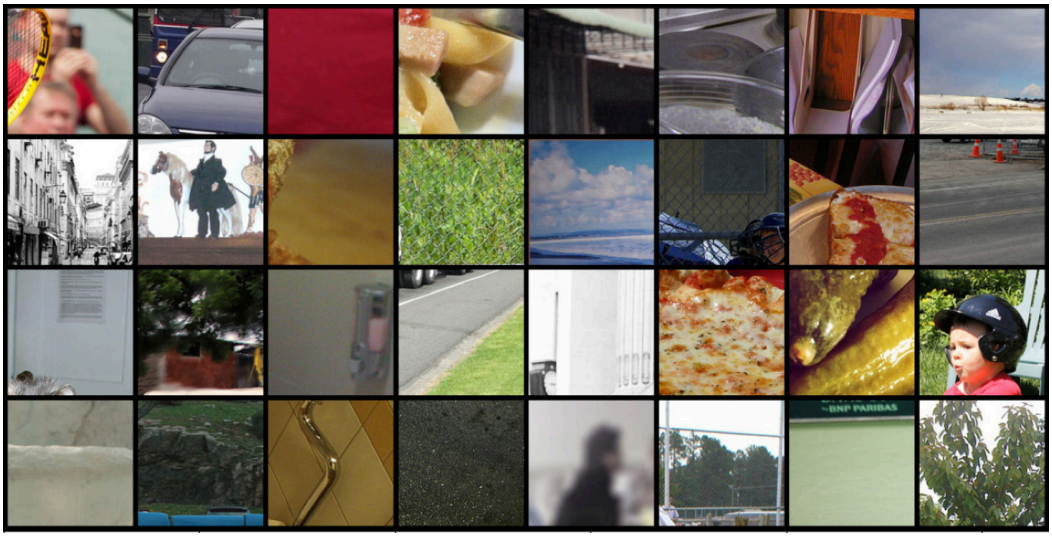

Figure 6.5: Illustration of training images extracted from the COCO dataset.

range (0.1 is used for N in this paper); and $||\mathcal{G}||_0$ denotes the total number of bytes occupied by the trainable parameters inside networks. In the remaining part of this section, the PSNR-pb values of the methods based on deep learning networks are shown for reference.

Experiments on COCO dataset: The COCO dataset [71] contains a large number of images from diverse natural scenes at different resolutions. We randomly select a subset and divide it into 20,480 images of 128×128 resolution for evaluation. A few samples of training images are shown in Figure 6.5. The size of the raw data for all training images is around $20 \times 1024 \times 128 \times 128 \times 3/(2^{20}) \approx 960$MB. During the computation of the bpp metric, the size of the proposed network is included, but the size of the network of other methods continues to be excluded from the bpp computation. For reference, the proposed PSNR-pb values of all methods based on deep learning are listed for reference, to demonstrate the information redundancy inside networks. In the comparisons, we select the smallest network provided by other methods to generate the lowest bitrate compression results, which is supposed to lead to the lowest information redundancy in their networks. This is a compromise since these networks usually require significant computational resources and the encoding of 20,480 images is extremely time-consuming.

The evaluation is shown in Table 6.1. According to the PSNR values of

Table 6.1: Evaluation of the proposed approach on 20,480 images extracted from the COCO dataset.

	PSNR	bpp	Net Size	PSNR-pb
CVPR22 [110]	30.71	0.368	463MB	22.02
ICLR19 [63]	27.39	0.125	26MB	19.95
CVPR18 [79]	25.98	0.300	115MB	17.89
CVPR17 [107]	22.61	0.300	95MB	14.60
ICASSP20 [19]	25.97	0.152	262MB	17.53
TPAMI21 [44]	29.65	0.283	81MB	21.72
JPEG2000	23.68	0.270	-	-
JPEG	24.59	0.385	-	-
BPG	24.96	0.086	-	-
our PCANet	24.32	0.153	2.64MB	18.70

the proposed approach, the results can be considered promising compared to state-of-the-art methods. The PSNR-pb score of the proposed approach is better than CVPR17 [107] and ICASSP20 [19]. This is a reasonable result since we have similar PSNR values of around 24.32 but our network is much smaller. It indicates that every byte of trainable parameters of the proposed approach has been effectively used to reduce mean squared loss. The size of the proposed network is only 2.64MB. The total size of the data used to encode the raw data of these 20,480 images is around 6MB, which is much smaller than pre-trained models proposed in other works. These results demonstrates its potential in real applications. However, the proposed approach shows no advantage compared to BPG, which is one of the best non-deep-learning based methods, because there are not many common features among the images, which are extracted from diverse natural scenes. In order to demonstrate the advantage of the proposed approach, we evaluate it on images with more common features, such as face images. The CelebA dataset [73] is thus used as the third dataset to evaluate the proposed approach.

Experiments on CelebA dataset: CelebA [73] is a dataset with more than 200 thousand celebrity images with diverse faces. We randomly select 20,480 face images of 218×178 resolution from the dataset. The total size of raw images is around $20 \times 1024 \times 178 \times 218 \times 3/(2^{30}) \approx 2.22$GB. A few samples of images are shown in Figure 6.6. Even though the faces vary, some common

Figure 6.6: Sample images from the CelebA dataset.

features can be found among these images such as eyes, nose, and mouth.

Comparisons of the proposed approach with state-of-the art methods are shown in Table 6.2. The proposed approach achieves almost the same results as BPG, which is better than that achieved on the COCO dataset. In contrast to images in the COCO dataset, face images are assumed to have more common features, where the proposed approach provided a better compression result. This validates our assumption that a group of images with common parts can be decomposed into several shared feature matrices. Given more common features in these images, better compression results can be attained by the proposed PCANet. In addition, the results of the proposed PCANet are still promising compared to other deep learning networks, such as CVPR17 [107], CVPR18 [79], and ICASSP20 [19]. It is worth highlighting that the PSNR-pb values between ICASSP20 [19] and the proposed PCANet are very close, but the bitrate of ICASSP20 [19] is almost twice as PCANet's bitrate of 0.075. To further demonstrate the advantage of the proposed PCANet, frames from the video "boats" from the CDNet dataset [6] are used for the last evaluation experiment.

Experiments on Boats dataset: The video "boats" is from the CDNet

Table 6.2: Evaluation of the proposed approach on 20,480 images extracted from the CelebA dataset.

	PSNR	bpp	Net Size	PSNR-pb
CVPR22 [110]	32.19	0.280	463MB	23.50
ICLR19 [63]	28.54	0.131	26MB	21.10
CVPR18 [79]	27.17	0.273	115MB	19.08
CVPR17 [107]	22.05	0.202	95MB	14.05
ICASSP20 [19]	26.85	0.133	262MB	18.41
TPAMI21 [44]	30.58	0.232	81MB	22.65
J2K	24.14	0.205	-	-
JPEG	24.51	0.359	-	-
BPG	25.47	0.076	-	-
Our PCANet	25.15	0.075	3.28MB	18.40

dataset [6]. It is recorded by a stationary camera in a dynamic background scenario, such as running water. We extract a few sample frames as well as the differences between two consecutive frames for illustration, which are shown in Figure 6.7. There are a total of 7,999 frames in this video, and the resolution is 320×240. The total size of the raw data is around $7999 \times 320 \times 240 \times 3/2^{30} \approx 1.71GB$. As shown in Figure 6.7, although the frames are generally similar, they still have different detail features.

For example, as shown by the magnified part of the difference images in Figure 6.7, the variation between the two frames is significant, due to the running water. In reality, details on the running water are highly unpredictable. Thus, previous networks, e.g., TPAMI2021 [44] and ICLR2019 [63], had a lower chance to learn such features. The advantage of the proposed approach in this area is obvious, and better results are achieved by the proposed approach compared with previous methods based on deep learning networks, such as TPAMI2021 [44] and ICLR2019 [63]. To demonstrate the advantage of the proposed approach, we evaluate all methods on the "boats" video with different bitrate settings. At the same time, the coding vectors of the proposed approach are also extended to generate results with different bitrates. The comparison results are shown in Figure 6.8, where the proposed approach achieves excellent results at very low bitrates. The PSNR and PSNR-pb values of the compared methods at their respective lowest bitrate settings are

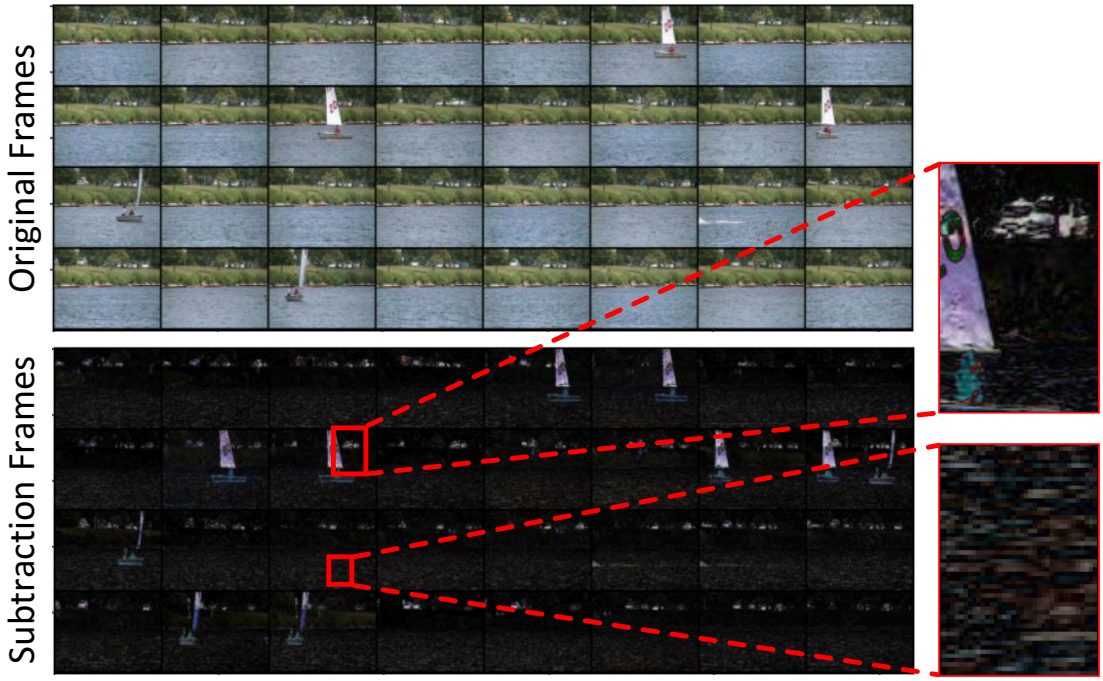

Figure 6.7: Illustration of sample frames and differences between consecutive frames from the video "boats."

shown in Table 6.3, as well as the PSNR and PSNR-pb values at the lowest and highest bitrate settings of the proposed approach. It can be seen from the plot that the rate-distortion curve of the proposed approach is closer to the top left corner than other methods, which indicates better performance.

The proposed approach achieves a PSNR of 25.95 dB with 0.058 bpp, which is better than all traditional methods. Moreover, at the highest bitrate setting, the proposed approach achieves a PSNR of 28.65 dB with 0.147 bpp, which is better than the state-of-the-art, TPAMI2021 [44]. This shows that previous methods based on deep learning networks are not general enough; otherwise, they should show similar performance with different datasets. The BD-PSNR and BD-rate scores comparing the proposed method to TPAMI2021 are shown in Table 6.4. It shows that on the average, the proposed method can increase the PSNR by 1.1091 dB at the same bitrate, or reduce the bitrate by -57.1026% for the same PSNR score, compared to TPAMI2021.

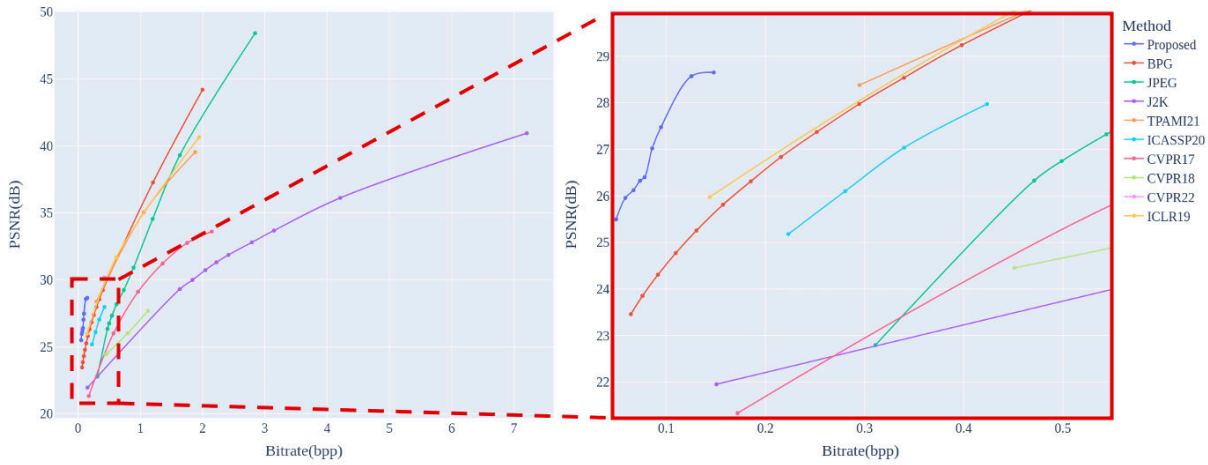

Figure 6.8: Comparison between the proposed approach and state-of-the-art methods on the boats video.

Table 6.3: Evaluation of the proposed approach on images from video "boats."

	PSNR	bpp	Net Size	PSNR-pb
CVPR22 [110]	30.17	0.425	463MB	21.48
ICLR19 [63]	25.97	0.143	26MB	18.53
CVPR18 [79]	24.45	0.451	115MB	16.36
CVPR17 [107]	21.33	0.171	95MB	13.33
ICASSP20 [19]	25.18	0.223	262MB	16.74
TPAMI21 [44]	28.38	0.294	81MB	20.45
J2K	21.95	0.150	-	-
JPEG	22.79	0.310	-	-
BPG	23.45	0.064	-	-
Our PCANet	25.95	0.058	2.96MB	19.45
	28.65	0.147	3.51MB	21.92

6.3.2 Discussion

In prior work, image compression methods based on deep learning were usually trained with a large number of images to produce models that were assumed to be general. Based on this assumption, the size of the network parameters was excluded from the bitrate computation. However, there is no guarantee that

Table 6.4: BD-PSNR and the corresponding BD-rate values comparing the proposed method and TPAMI2021. For BD-PSNR, a positive value x indicates that the proposed method can increase the performance by x at the same bitrate. For BD-rate, a negative value $-x$ indicates that the proposed method can achieve the same level of performance with a bitrate saving of $x\%$.

Dataset	BD-PSNR	BD-rate
Kodak	1.88	-48.29
Boats	1.11	-57.10

pre-trained networks perform adequately for unseen images since networks rely on training images and overfitting problems always exist. In contrast, the proposed PCANet network is deliberately trained for overfitting since an overfitted network on a certain dataset can better reconstruct the data from that dataset. This overfitting might be a problem for other methods because it is preferable to have a general model that only needs to be transmitted once and can be used for all images is preferred. But since the proposed method produces a very small network, we can encode the network parameters along with the coding vectors and transmit them together without significantly increasing the bitrate. Thus, we included the size of the proposed network into the bitrate computation.

The advantage of the proposed PCANet is that it not only reduces redundancy in the coding vectors but also the redundancy in the model itself. This is a result of the aforementioned intentional overfitting which allows the model to learn the most representative features in the training images, and leads to a better reconstruction. While other learning based methods aim to train a general model, it might be difficult to find common features among images from different categories. To alleviate this problem, a larger model and large quantities of training data are needed. This limits their ability to remove the internal redundancy in the models and might lead to worse reconstruction quality in a specific category.

The proposed method achieves better performance than other methods because of this reduced redundancy. A smaller model enables us to have separate models targeting different image categories. We demonstrated the benefit of

(a) Reconstructed images. (b) Original images

Figure 6.9: Demonstration of unseen images reconstructed by the proposed PCANet, with a PSNR of 26.43, and a bpp of 0.158.

this in the experiments section. Moreover, a smaller model requires fewer computing resources to encode and decode images. This allows for faster runtime and lower energy consumption, which is important for real-world applications.

Even though the proposed PCANet is overfitted, it can still be used for unseen images. The reason for this is that the proposed network learns how to decompose images into shared feature matrices, and these feature matrices can be used to reconstruct unseen images if they contain similar features. To demonstrate this, we encoded one thousand unseen faces images using a PCANet trained on the CelebA dataset. The results are shown in Figure 6.9, where the proposed PCANet achieves a PSNR of 26.43 dB with a bpp of 0.158. This demonstrates the potential of the proposed PCANet to generalize if trained on a large number of images.

However, it is worth noting that currently the proposed method is unable to take advantage of higher bitrates to provide better image quality because it is designed to reduce the redundancy as much as possible. In a future research, it might be possible to solve this problem by relaxing the constraint

on redundancy removal and allowing the proposed method to use more bits to improve the image quality.

6.4 Conclusion

We proposed the Principal Component Approximation Network (PCANet) for image compression, based on the assumption that a group of images can be decomposed into several shared feature matrices. PCANet was devised to learn these shared feature matrices and the weights corresponding to each image for reconstruction using weighted sums. PCANet contains multiple scale paths to capture common features at different scales. The weights corresponding to feature matrices are learned by minimizing the errors between the reconstructed and original images.

The proposed method differentiates itself from other deep learning-based methods by treating image compression as a decomposition problem. This allows us to not only reduce redundancy in the coding vectors but also redundancy in the model itself. Consequently, the proposed method produces a very small network, which can be encoded along with the coding vectors without significantly increasing the bitrate. Comprehensive evaluations comparing our approach to state-of-the-art methods show promising performance and demonstrate its potential for use in image and video compression.

Chapter 7

Conclusion and Future Work

In this thesis, we incorporated several perceptual factors into the design of image and video processing algorithms, where existing methods were insufficient to address problems such as single image lighting enhancement, naturalness and quality of lighting enhancement, incorporation of human visual system (HVS) characteristics, and effective removal of redundancy inside a neural network model.

Chapters 3 and 4 incorporate perceptual factors such as brightness, contrast, saturation, and global content dependencies to improve the perceptual quality of single image lighting enhancement.

In Chapter 3, we proposed a image lighting enhancement method based on fusion pyramid, which addresses the over-enhancement problem and improves the quality of enhanced images in single image lighting enhancement. A future direction of research could investigate the possibility of using the proposed method in video lighting enhancement.

In Chapter 4, we designed a neural network based on the self-attention mechanism to model the long distance dependencies in the image, which addresses the problem of artifacts and improves the image quality. We also introduced a new loss function based on the characteristics of HDR images, which alleviate the color shift/artifacts in the output images. A future direction of research can involve adding the ability to model time-domain dependencies in the network to enable multi frame/video enhancement.

Chapters 5 and 6 utilize perceptual factors such as foveation, visual saliency,

and pattern sensitivity to boost the performance of image and video compression algorithms.

In Chapter 5, we achieved foveated video compression using a foveation process based on per-quad image warping. This process can produce non-uniform subsampling of the video frames based on different visual saliency levels, consequently increasing the overall compression rate while preserving the image quality of salient regions. By using the per-quad image warping, we also enable a more precise quality control of salient regions because saliency data can be incorporated at a lower granularity. Moreover, while existing methods introduce modifications to the current video compression pipeline, the proposed method is independent from traditional encoding processes, making it applicable to improve most existing compression methods. A future direction of research can involve exploring other ways to achieve non-uniform subsampling that might have less computational cost. For example, using an adaptive algorithm to create a quad grid made of quads of different sizes. Furthermore, some other mesh generation methods can be investigated to achieve better quality control of salient regions, such as elliptic mesh generation.

In Chapter 6, we proposed a neural network for image compression. Unlike existing learning based compression methods, the proposed method treats the image compression problem as a decomposition problem. The proposed method learns the most representative features among the training images and decompose the images as linear combinations of the features. This approach allows the proposed method to not only reduce redundancy in the coding vectors, but also in the network model itself. Therefore, the proposed method can achieve promising compression rate even with the size of the model included in the bitrate calculation. A future direction of research can involve relaxing the redundancy reduction constraint in the current model to allow higher bitrate and achieve better image quality. It would also be interesting to investigate how image enhancement will impact the performance of the proposed method.

References

[1] S. S. Agaian, B. Silver, and K. A. Panetta, "Transform Coefficient Histogram-Based Image Enhancement Algorithms Using Contrast Entropy," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 741–758, Mar. 2007, ISSN: 1941-0042. DOI: 10.1109/TIP.2006.888338. 9

[2] E. Agustsson, F. Mentzer, M. Tschannen, *et al.*, "Soft-to-hard vector quantization for end-to-end learning compressible representations," *Advances in neural information processing systems*, vol. 30, 2017. 6, 16, 67

[3] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, *Generative Adversarial Networks for Extreme Learned Image Compression*, Aug. 2019. DOI: 10.48550/arXiv.1804.02958. arXiv: 1804.02958 [cs]. (visited on 06/27/2022). 6, 16, 67

[4] E. Ahanonu, M. Marcellin, and A. Bilgin, "Lossless Image Compression Using Reversible Integer Wavelet Transforms and Convolutional Neural Networks," in *2018 Data Compression Conference*, Mar. 2018, pp. 395–395. DOI: 10.1109/DCC.2018.00048. 6, 16, 67

[5] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974. 15, 16

[6] Y. W. et al., "Cdnet 2014: An expanded change detection benchmark dataset," in *CVPRW*, Jun. 2014, pp. 393–400. DOI: 10.1109/CVPRW.2014.126. 78, 84, 85

[7] H. Amirpour, A. Pinheiro, M. Pereira, F. J. P. Lopes, and M. Ghanbari, "Efficient Light Field Image Compression with Enhanced Random Access," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 18, no. 2, 44:1–44:18, Mar. 2022, ISSN: 1551-6857. DOI: 10.1145/3471905. (visited on 10/17/2023). 15

[8] V. G. An and C. Lee, "Single-shot high dynamic range imaging via deep convolutional neural network," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec. 2017, pp. 1768–1772. DOI: 10.1109/APSIPA.2017.8282319. 12

[9] *Andrews* and *Pratt*, "Fourier transform coding of images," in *Proc. Hawaii Int. Conf. System Sciences*, 1968, pp. 677–679. 15, 16

[10] M. H. Baig, V. Koltun, and L. Torresani, "Learning to inpaint for image compression," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf. 6, 16, 67

[11] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *2017 5th International Conference on Learning Representations*, 2016. 6, 16, 67

[12] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *2018 6th International Conference on Learning Representations*, 2018. 6, 16, 67

[13] A. Banitalebi-Dehkordi, M. Azimi, M. T. Pourazad, and P. Nasiopoulos, "Compression of High Dynamic Range Video Using the HEVC and H.264/AVC Standards," *arXiv:1803.04823 [eess]*, Mar. 2018. arXiv: 1803.04823 [eess]. (visited on 08/13/2021). 11

[14] J. Bankoski, P. Wilkins, and Y. Xu, "Technical overview of VP8, an open source video codec for the web," in *2011 IEEE International Conference on Multimedia and Expo*, Jul. 2011, pp. 1–6. DOI: 10.1109/ICME.2011.6012227. 40

[15] A. Basu and K. J. Wiebe, "Enhancing videoconferencing using spatially varying sensing," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 2, pp. 137–148, Mar. 1998, ISSN: 1558-2426. DOI: 10.1109/3468.661143. 6, 14, 44

[16] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *ITU SG16 Doc. VCEG-M33*, 2001. 56, 79, 81

[17] J. Chen, H. Song, K. Zhang, B. Liu, and Q. Liu, "Video saliency prediction using enhanced spatiotemporal alignment network," *Pattern Recognition*, vol. 109, p. 107 615, Jan. 2021, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2020.107615. (visited on 06/02/2021). 14

[18] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, "End-to-end learnt image compression via non-local attention optimization and improved context modeling," *IEEE Transactions on Image Processing*, vol. 30, pp. 3179–3191, 2021. DOI: 10.1109/TIP.2021.3058615. 6, 16, 67

[19] T. Chen and Z. Ma, "Variable Bitrate Image Compression with Quality Scaling Factors," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 2163–2167. DOI: 10.1109/ICASSP40776.2020.9053885. 79, 83–85, 87

[20] I. Cheng and A. Basu, "QoS based video delivery with foveation and bandwidth monitoring," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2675–2686, Nov. 2003, ISSN: 0167-8655. DOI: 10.1016/S0167-8655(03)00110-7. (visited on 09/27/2021). 6, 14

[21] I. Cheng, A. Basu, and Y. Pan, "Parametric Foveation for Progressive Texture and Model Transmission," *Eurographics (Posters)*, 2003. 6, 14

[22] I. Cheng, M. Mohammadkhani, A. Basu, and F. Dufaux, "Foveated High Efficiency Video Coding for Low Bit Rate Transmission," in *2015 IEEE International Symposium on Multimedia (ISM)*, Dec. 2015, pp. 547–552. DOI: 10.1109/ISM.2015.37. 6, 14

[23] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learning image and video compression through spatial-temporal energy compaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019. 6, 16, 67

[24] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020. 6, 16, 67

[25] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE transactions on consumer electronics*, vol. 46, no. 4, pp. 1103–1127, 2000. 15, 16, 68, 79

[26] R. Cong, J. Lei, H. Fu, M.-M. Cheng, W. Lin, and Q. Huang, "Review of Visual Saliency Detection with Comprehensive Information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 2941–2959, Oct. 2019, ISSN: 1051-8215, 1558-2205. DOI: 10.1109/TCSVT.2018.2870832. arXiv: 1803.03391. (visited on 01/29/2021). 14

[27] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting Human Eye Fixations via an LSTM-Based Saliency Attentive Model," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, Oct. 2018, ISSN: 1941-0042. DOI: 10.1109/TIP.2018.2851672. 14

[28] P. Didyk, R. Mantiuk, H. Hein, and H.-P. Seidel, "Enhancement of Bright Video Features for HDR Displays," *Computer Graphics Forum*, vol. 27, no. 4, pp. 1265–1274, 2008, ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2008.01265.x. (visited on 12/16/2019). 10

[29] X. Dong, G. Wang, Y. Pang, *et al.*, "Fast efficient algorithm for enhancement of low lighting video," in *2011 IEEE International Conference on Multimedia and Expo*, Jul. 2011, pp. 1–6. DOI: 10.1109/ICME.2011.6012107. 11

[30] D. L. Donoho and I. M. Johnstone, "Adapting to Unknown Smoothness via Wavelet Shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, Dec. 1995, ISSN: 0162-1459. DOI: 10.1080/01621459.1995.10476626. (visited on 12/16/2019). 10

[31] R. Droste, J. Jiao, and J. A. Noble, "Unified Image and Video Saliency Modeling," *arXiv:2003.05477 [cs]*, vol. 12350, pp. 419–435, 2020. DOI: 10.1007/978-3-030-58558-7_25. arXiv: 2003.05477 [cs]. (visited on 05/20/2021). 14

[32] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger, "HDR image reconstruction from a single exposure using deep CNNs," *Acm Transactions on Graphics*, vol. 36, no. 6, p. 178, Nov. 2017, ISSN: 0730-0301. DOI: 10.1145/3130800.3130816. 5, 10, 12, 34

[33] F. Frieß, M. Braun, V. Bruder, S. Frey, G. Reina, and T. Ertl, "Foveated Encoding for Large High-Resolution Displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1850–1859, Feb. 2021, ISSN: 1941-0506. DOI: 10.1109/TVCG.2020.3030445. 5, 15

[34] X. Fu, D. Zeng, Y. Huang, X.-P. Zhang, and X. Ding, "A Weighted Variational Model for Simultaneous Reflectance and Illumination Estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2782–2790. DOI: 10.1109/CVPR.2016.304. 11

[35] R. Gal, O. Sorkine, and D. Cohen-Or, "Feature-aware texturing," in *Proceedings of the 17th Eurographics Conference on Rendering Techniques*, ser. EGSR '06, Goslar, DEU: Eurographics Association, Jun. 2006, pp. 297–303, ISBN: 978-3-905673-35-7. (visited on 11/15/2021). 44

[36] G. Gao, P. You, R. Pan, *et al.*, "Neural image compression via attentional multi-scale back projection and frequency decomposition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 14 677–14 686. 6, 16, 67

[37] W. Gao, S. Sun, H. Zheng, Y. Wu, H. Ye, and Y. Zhang, "OpenDMC: An open-source library and performance evaluation for deep-learning-based multi-frame compression," in *ACM International Conference on Multimedia (ACM MM)*, Jul. 2023. 16, 67

[38] K. Gregor, F. Besse, D. Jimenez Rezende, I. Danihelka, and D. Wierstra, "Towards conceptual compression," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016. 6, 16, 67

[39] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 1462–1471. 6, 16, 67

[40] X. Guo, Y. Li, and H. Ling, "LIME: Low-Light Image Enhancement via Illumination Map Estimation," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 982–993, Feb. 2017, ISSN: 1941-0042. DOI: 10.1109/TIP.2016.2639450. 5, 6, 11, 20, 34

[41] J. Han, B. Li, D. Mukherjee, *et al.*, "A Technical Overview of AV1," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1435–1462, Sep. 2021, ISSN: 1558-2256. DOI: 10.1109/JPROC.2021.3058584. 40

[42] S. W. Hasinoff, D. Sharlet, R. Geiss, *et al.*, "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Transactions on Graphics*, vol. 35, no. 6, 192:1–192:12, Nov. 2016, ISSN: 0730-0301. DOI: 10.1145/2980179.2980254. (visited on 08/30/2020). 11, 31

[43] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 5718–5727. 6, 16, 67

[44] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning end-to-end lossy image compression: A benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4194–4211, 2022. DOI: 10.1109/TPAMI.2021.3065339. 6, 16, 67, 79, 83, 85–87

[45] Hu, H., de Haan, Gerard, Otten, Ralph H.J.M., and Electronic Systems, "Video enhancement : Content classification and model selection," Ph.D. dissertation, Technische Universiteit Eindhoven, 2010. (visited on 12/16/2019). 9

[46] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952, ISSN: 2162-6634. DOI: 10.1109/JRPROC.1952.273898. 15

[47] T. T. M. Huynh, T.-D. Nguyen, M.-T. Vo, and S. V. T. Dao, "High Dynamic Range Imaging Using A 2x2 Camera Array with Polarizing Filters," in *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, Sep. 2019, pp. 183–187. DOI: 10.1109/ISCIT.2019.8905122. (visited on 10/10/2023). 2

[48] G. Illahi, T. van Gemert, M. Siekkinen, E. Masala, A. Oulasvirta, and A. Ylä-Jääski, "Cloud Gaming with Foveated Video Encoding," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, pp. 1–24, Mar. 2020. DOI: 10.1145/3369110. 5, 15

[49] "Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 12: Image File Format," International Organization for Standardization, Geneva, CH, Standard, Sep. 2022. 4

[50] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998, ISSN: 1939-3539. DOI: 10.1109/34.730558. 14

[51] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, "DeepVS: A Deep Learning Based Video Saliency Prediction Approach," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 625–642, ISBN: 978-3-030-01264-9. DOI: 10.1007/978-3-030-01264-9_37. 14

[52] Y. Jiang, X. Gong, D. Liu, *et al.*, "EnlightenGAN: Deep Light Enhancement without Paired Supervision," *arXiv:1906.06972 [cs, eess]*, Jun. 2019. arXiv: 1906.06972 [cs, eess]. (visited on 10/29/2019). 5, 11, 12, 34

[53] N. Johnston, D. Vincent, D. Minnen, *et al.*, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018. 6, 16, 67

[54] J.-H. Kim, B. Heo, and J.-S. Lee, "Joint global and local hierarchical priors for learned image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 5992–6001. 6, 16, 67

[55] S. Y. Kim, J. Oh, and M. Kim, "Deep SR-ITM: Joint Learning of Super-Resolution and Inverse Tone-Mapping for 4K UHD HDR Applications," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 3116–3125, ISBN: 978-1-72814-803-8. DOI: 10.1109/ICCV.2019.00321. (visited on 09/03/2020). 5, 12, 34

[56] R. P. Kovaleski and M. M. Oliveira, "High-quality brightness enhancement functions for real-time reverse tone mapping," *The Visual Computer*, vol. 25, no. 5, pp. 539–547, May 2009, ISSN: 1432-2315. DOI: 10.1007/s00371-009-0327-3. (visited on 12/16/2019). 10

[57] S. S. S. Kruthiventi, K. Ayush, and R. V. Babu, "DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations," *arXiv:1510.02927 [cs]*, Oct. 2015. arXiv: 1510.02927 [cs]. (visited on 01/30/2021). 14

[58] M. Kümmerer, T. S. A. Wallis, and M. Bethge, "DeepGaze II: Reading fixations from deep features trained on object recognition," *arXiv:1610.01563 [cs, q-bio, stat]*, Oct. 2016. arXiv: 1610.01563 [cs, q-bio, stat]. (visited on 01/30/2021). 14

[59] D. Kundu, D. Ghadiyaram, A. C. Bovik, and B. L. Evans, "No-Reference Quality Assessment of Tone-Mapped HDR Pictures," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2957–2971, Jun. 2017, ISSN: 1941-0042. DOI: 10.1109/TIP.2017.2685941. 13, 32

[60] E. H. Land and J. J. McCann, "Lightness and Retinex Theory," *JOSA*, vol. 61, no. 1, pp. 1–11, Jan. 1971. DOI: 10.1364/JOSA.61.000001. (visited on 08/13/2021). 11

[61] K. G. Larkin, *Reflections on Shannon Information: In search of a natural information-entropy for images*, Sep. 2016. DOI: 10.48550/arXiv.1609.01117. arXiv: 1609.01117 [cs, math]. (visited on 01/04/2023). 51

[62] J.-H. Lee, S. Jeon, K. P. Choi, Y. Park, and C.-S. Kim, "Dpict: Deep progressive image compression using trit-planes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 16 113–16 122. 6, 16, 67

[63] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," in *the 7th Int. Conf. on Learning Representations*, May 2019. 6, 16, 67, 79, 83, 85, 87

[64] S. Lee, "An efficient content-based image enhancement in the compressed domain using retinex theory," *Ieee Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 2, pp. 199–213, Feb. 2007, ISSN: 1051-8215. DOI: 10.1109/TCSVT.2006.887078. 10

[65] J. Lei, X. Liu, B. Peng, D. Jin, W. Li, and J. Gu, "Deep stereo image compression via bi-directional coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 19 669–19 678. 6, 16, 67

[66] J. Li and P. Fang, "HDRNET: Single-Image-based HDR Reconstruction Using Channel Attention CNN," in *Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing*, ser. ICMSSP 2019, New York, NY, USA: Association for Computing Machinery, May 2019, pp. 119–124, ISBN: 978-1-4503-7171-1. DOI: 10.1145/3330393.3330426. (visited on 01/11/2021). 12, 26

[67] M. Li, J. Liu, W. Yang, X. Sun, and Z. Guo, "Structure-Revealing Low-Light Image Enhancement Via Robust Retinex Model," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2828–2841, Jun. 2018, ISSN: 1941-0042. DOI: 10.1109/TIP.2018.2810539. 12

[68] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018. 6, 16, 67

[69] Z. Li, S. Qin, and L. Itti, "Visual attention guided bit allocation in video compression," *Image and Vision Computing*, vol. 29, no. 1, pp. 1–14, Jan. 2011, ISSN: 0262-8856. DOI: 10.1016/j.imavis.2010.07.001. (visited on 09/15/2021). 55

[70] C.-Y. Lin, K.-R. Jheng, and T. K. Shih, "Objective HDR image quality assessment," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 1547–1567, Jan. 2019, ISSN: 1573-7721. DOI: 10.1007/s11042-018-6139-6. (visited on 10/10/2023). 2

[71] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755. 78, 82

[72] P. Linardos, E. Mohedano, J. J. Nieto, N. E. O'Connor, X. Giro-i-Nieto, and K. McGuinness, "Simple vs complex temporal recurrences for video saliency prediction," *arXiv:1907.01869 [cs]*, Jul. 2019. arXiv: 1907.01869 [cs]. (visited on 06/01/2021). 14

[73] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, Dec. 2015. 78, 83

[74] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1683–1698, 2019. 15, 16

[75] H. Malm, M. Oskarsson, E. Warrant, P. Clarberg, J. Hasselgren, and C. Lejdfors, "Adaptive enhancement and noise reduction in very low light-level video," in *2007 IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4409007. 10

[76] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, "HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions," *ACM Transactions on Graphics*, vol. 30, no. 4, 40:1–40:14, Jul. 2011, ISSN: 0730-0301. DOI: 10.1145/2010324.1964935. (visited on 09/03/2020). 13

[77] D. Marnerides, T. Bashford-Rogers, J. Hatchett, and K. Debattista, "ExpandNet: A Deep Convolutional Neural Network for High Dynamic Range Expansion from Low Dynamic Range Content," *arXiv:1803.02266 [cs]*, Sep. 2019. arXiv: 1803.02266 [cs]. (visited on 09/30/2021). 5, 10, 12, 34

[78] S. Mathe and C. Sminchisescu, "Actions in the Eye: Dynamic Gaze Datasets and Learnt Saliency Models for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1408–1424, Jul. 2015, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2014.2366154. 51

[79] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018. 6, 16, 67, 79, 83–85, 87

[80] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.

neurips.cc/paper/2018/file/53edebc543333dfbf7c5933af792c9c4-Paper.pdf. 6, 16, 67

[81] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "Completely Blind" Image Quality Analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, Mar. 2013, ISSN: 1558-2361. DOI: 10.1109/LSP.2012.2227726. 13, 22, 32

[82] M. Muckley, J. Juravsky, D. Severo, M. Singh, Q. Duval, and K. Ullrich, *Neuralcompression*, https://github.com/facebookresearch/NeuralCompression, 2021. 6, 16, 67

[83] D. Mukherjee, J. Han, J. Bankoski, *et al.*, "A Technical Overview of VP9 – The Latest Open-Source Video Codec," in *SMPTE 2013 Annual Technical Conference Exhibition*, Oct. 2013, pp. 1–17. DOI: 10.5594/M001518. 40

[84] M. Narwaria, M. Perreira Da Silva, and P. Le Callet, "HDR-VQM: An objective quality measure for high dynamic range video," *Signal Processing: Image Communication*, vol. 35, pp. 46–60, Jul. 2015, ISSN: 0923-5965. DOI: 10.1016/j.image.2015.04.009. 2, 24

[85] Y. Niu, J. Wu, W. Liu, W. Guo, and R. W. H. Lau, "HDR-GAN: HDR Image Reconstruction from Multi-Exposed LDR Images with Large Motions," *arXiv:2007.01628 [cs, eess]*, Jul. 2020. arXiv: 2007.01628 [cs, eess]. (visited on 01/12/2021). 12

[86] Y. Patel, S. Appalaraju, and R. Manmatha, "Saliency driven perceptual image compression," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 227–236. 6, 16, 67

[87] S. M. Pizer, E. P. Amburn, J. D. Austin, *et al.*, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, Sep. 1987, ISSN: 0734-189X. DOI: 10.1016/S0734-189X(87)80186-X. (visited on 08/13/2021). 11

[88] A. Polakovič, R. Vargic, G. Rozinaj, and G.-M. Muntean, "An Approach to Video Compression Using Saliency Based Foveation," in *2018 International Symposium ELMAR*, Sep. 2018, pp. 169–172. DOI: 10.23919/ELMAR.2018.8534631. 15

[89] W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard transform image coding," *Proceedings of the IEEE*, vol. 57, no. 1, pp. 58–68, 1969. 15, 16

[90] Y. Rao and L. Chen, "A survey of video enhancement techniques," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 3, no. 1, pp. 71–99, 2012. 9, 18

[91] Y. Rao, W. Y. Lin, and L. Chen, "Image-based fusion for video enhancement of night-time surveillance," *Optical Engineering*, vol. 49, no. 12, p. 120 501, Dec. 2010, ISSN: 0091-3286, 1560-2303. DOI: 10.1117/1.3520553. (visited on 12/16/2019). 5, 10

[92] R. Rassool, "VMAF reproducibility: Validating a perceptual practical video quality metric," in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Jun. 2017, pp. 1–2. DOI: 10.1109/BMSB.2017.7986143. 55

[93] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '02, New York, NY, USA: Association for Computing Machinery, Jul. 2002, pp. 267–276, ISBN: 978-1-58113-521-3. DOI: 10.1145/566570.566575. (visited on 06/30/2021). 34

[94] X. Ren, M. Li, W.-H. Cheng, and J. Liu, "Joint Enhancement and Denoising Method via Sequential Decomposition," *arXiv:1804.08468 [cs]*, Apr. 2018. arXiv: 1804.08468 [cs]. (visited on 08/13/2021). 12

[95] H. Rhee, Y. I. Jang, S. Kim, and N. I. Cho, "Lc-fdnet: Learned lossless image compression with frequency decomposition network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 6033–6042. 6, 16, 67

[96] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, Aug. 2017, pp. 2922–2930. 6, 16, 67

[97] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv:1505.04597 [cs]*, May 2015. arXiv: 1505.04597 [cs]. (visited on 08/26/2020). 25

[98] V. Sanchez, A. Basu, and M. K. Mandal, "Prioritized region of interest coding in JPEG2000," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 9, pp. 1149–1155, Sep. 2004, ISSN: 1558-2205. DOI: 10.1109/TCSVT.2004.833168. 5, 14

[99] *Series*, "Methodology for the subjective assessment of the quality of television pictures," *Recommendation ITU-R BT*, vol. 500, pp. 500–14, Oct. 2019. 54

[100] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948, ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x. 50

[101] J. Shen, R. H. Deng, Z. Cheng, L. Nie, and S. Yan, "On robust image spam filtering via comprehensive visual modeling," *Pattern Recognition*, Discriminative Feature Learning from Big Data for Visual Recognition, vol. 48, no. 10, pp. 3227–3238, Oct. 2015, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2015.02.027. (visited on 10/17/2023). 15

[102] J. Shen and N. Robertson, "BBAS: Towards large scale effective ensemble adversarial attacks against deep neural network learning," *Information Sciences*, vol. 569, pp. 469–478, Aug. 2021, ISSN: 0020-0255. DOI: 10.1016/j.ins.2020.11.026. (visited on 10/17/2023). 16, 67

[103] M. Song, J. Choi, and B. Han, "Variable-rate deep image compression through spatially-adaptive feature transform," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 2380–2389. 6, 16, 67

[104] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012, ISSN: 1558-2205. DOI: 10.1109/TCSVT.2012.2221191. 40

[105] J. Tang, E. Peli, and S. Acton, "Image enhancement using a contrast measure in the compressed domain," *IEEE Signal Processing Letters*, vol. 10, no. 10, pp. 289–292, Oct. 2003, ISSN: 1558-2361. DOI: 10.1109/LSP.2003.817178. 10

[106] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," in *2017 5th International Conference on Learning Representations*, 2017. 6, 16, 67

[107] G. Toderici, D. Vincent, N. Johnston, *et al.*, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. 6, 16, 67, 79, 83–85, 87

[108] G. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, Feb. 1992, ISSN: 1558-4127. DOI: 10.1109/30.125072. 15, 16, 68, 79

[109] B. A. Wandell, *Foundations of Vision*. Oxford University Press, Incorporated, 1995, ISBN: 978-0-87893-853-7. 3, 4, 40

[110] D. Wang, W. Yang, Y. Hu, and J. Liu, "Neural data-dependent transform for learned image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 17 379–17 388. 6, 16, 67, 79, 83, 85, 87

[111] S. Wang, J. Zheng, H.-M. Hu, and B. Li, "Naturalness Preserved Enhancement Algorithm for Non-Uniform Illumination Images," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3538–3548, Sep. 2013, ISSN: 1941-0042. DOI: 10.1109/TIP.2013.2261309. 5, 11

[112] W. Wang, J. Shen, F. Guo, M.-M. Cheng, and A. Borji, "Revisiting Video Saliency: A Large-scale Benchmark and a New Model," *arXiv:1801.07424 [cs]*, May 2018. arXiv: 1801.07424 [cs]. (visited on 06/01/2021). 14

[113] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, ISSN: 1941-0042. DOI: 10.1109/TIP.2003.819861. 13, 55

[114] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Ieee, vol. 2, 2003, pp. 1398–1402. 6, 16, 67

[115] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep Retinex Decomposition for Low-Light Enhancement," *arXiv:1808.04560 [cs]*, Aug. 2018. arXiv: 1808.04560 [cs]. (visited on 09/03/2020). 5, 12, 34

[116] K. J. Wiebe and A. Basu, "Improving image and video transmission quality over ATM with foveal prioritization and priority dithering," *Pattern Recognition Letters*, vol. 22, no. 8, pp. 905–915, Jun. 2001, ISSN: 0167-8655. DOI: 10.1016/S0167-8655(01)00032-0. (visited on 09/27/2021). 6, 14

[117] O. Wiedemann, V. Hosu, H. Lin, and D. Saupe, "Foveated Video Coding for Real-Time Streaming Applications," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2020, pp. 1–6. DOI: 10.1109/QoMEX48832.2020.9123080. 5, 15

[118] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003, ISSN: 1558-2205. DOI: 10.1109/TCSVT.2003.815165. 40

[119] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987, ISSN: 0001-0782. DOI: 10.1145/214762.214771. (visited on 06/27/2022). 15

[120] M. Wödlinger, J. Kotera, J. Xu, and R. Sablatnig, "Sasic: Stereo image compression with latent shifts and stereo attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 661–670. 6, 16, 67

[121] L. Wu, K. Huang, and H. Shen, "A GAN-based Tunable Image Compression System," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2020, pp. 2323–2331. DOI: 10.1109/WACV45572.2020.9093387. 6, 16, 67

[122] Q. Yan, D. Gong, Q. Shi, *et al.*, "Attention-Guided Network for Ghost-Free High Dynamic Range Imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1751–1760. (visited on 01/12/2021). 12

[123] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-Attention Generative Adversarial Networks," *arXiv:1805.08318 [cs, stat]*, Jun. 2019. arXiv: 1805.08318 [cs, stat]. (visited on 01/30/2021). 26

[124] Q. Zhang, Y. Nie, L. Zhang, and C. Xiao, "Underexposed Video Enhancement via Perception-Driven Progressive Fusion," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 6, pp. 1773–1785, Jun. 2016, ISSN: 1941-0506. DOI: 10.1109/TVCG.2015.2461157. 5, 9, 10

[125] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018. 55

[126] Y. Zhang and S. Liu, "Non-uniform Illumination Video Enhancement Based on Zone System and Fusion," in *2018 24th International Conference on Pattern Recognition (ICPR)*, Aug. 2018, pp. 2711–2716. DOI: 10.1109/ICPR.2018.8545189. 10

[127] L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu, "Variational autoencoder for low bit-rate image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2617–2620. 6, 16, 67

[128] S. Zhu, Q. Chang, and Q. Li, "Video saliency aware intelligent HD video compression with the improvement of visual quality and the reduction of coding complexity," *Neural Computing and Applications*, vol. 34, no. 10, pp. 7955–7974, May 2022, ISSN: 1433-3058. DOI: 10.1007/s00521-022-06895-1. (visited on 01/28/2023). 5, 15

[129] S. Zhu and Z. Xu, "Spatiotemporal visual saliency guided perceptual high efficiency video coding with neural network," *Neurocomputing*, vol. 275, pp. 511–522, Jan. 2018, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.08.054. (visited on 01/28/2023). 5, 15

[130] X. Zhu, J. Song, L. Gao, F. Zheng, and H. T. Shen, "Unified multivariate gaussian mixture for efficient neural image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 17 612–17 621. 6, 16, 67

[131] H. Zimmer, A. Bruhn, and J. Weickert, "Freehand HDR Imaging of Moving Scenes with Simultaneous Resolution Enhancement," *Computer Graphics Forum*, vol. 30, no. 2, pp. 405–414, 2011, ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2011.01870.x. (visited on 12/16/2019). 10

[132] R. Zou, C. Song, and Z. Zhang, "The devil is in the details: Window-based attention for image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 17 492–17 501. 6, 16, 17, 67