

Image Compression and Quality Assessment based on Deep Learning

深層学習に基づく画像圧縮と品質評価

February 2020

Zhengxue CHENG

程 正雪

Image Compression and Quality Assessment based on Deep Learning

深層学習に基づく画像圧縮と品質評価

February 2020

Waseda University

Graduate School of Fundamental Science and Engineering
Department of Computer Science and Communications Engineering
Research on Image Information

Zhengxue CHENG

程 正雪

ABSTRACT

Image compression and quality assessment are fundamental research topics in the field of signal processing, due to their significant influence on data transmission, storage and perceived quality by human beings. Besides, there is a strong correlation between these two topics, because subjective quality is one of important criteria for image compression task. Therefore, how to evaluate the subjective quality perceived by human beings accurately and how to compress images efficiently maintaining high subjective reconstruction quality are two important topics that I would like to focus during my doctoral course.

Recently, deep learning has been successfully applied into many computer vision tasks and has also attracted great attention from signal processing community, especially on image compression and quality assessment. Benefitting from the advantages of deep learning techniques, deep learning-based quality assessment could achieve no-reference assessment and high accuracy using neural networks. On the other hand, deep learning-based image compression is expected to adapt to new image formats quickly and achieve perceptual quality optimization and potential coding efficiency results.

However, there are several critical issues to be addressed for deep learning-based image compression and quality assessment, respectively. Regarding image quality assessment (IQA), two main problems are the high-complexity of convolutional neural networks (CNN) approaches and the unknown distortion types for CNN models. The details are explained as follows. (1) Conventional image quality prediction methods use hand-crafted features from distorted images, which are not efficient, while CNN features are more efficient but brings high time complexity. (2) There are different distortion types blind to quality assessment, but one pre-trained CNN model is difficult to achieve robust performance for all the distortion types, such as Gaussian noises, blur distortions, different compression distortions and so on. Regarding the learning-based image compression, which is an emerging topic, some key problems exist, such as the selection of many neural network architectures, spatial redundancy reduction and fair and rigorous comparison with traditional codecs when developing end-to-end learning-based compression approaches. The detailed explanations are as followed. (1) The selection of neural network not only refers to the types of neural networks, such as generative adversarial networks (GAN) and convolutional autoencoder (CAE), but also includes the structure of neural networks, such as the number of convolutional layers, the number of filters for compressed codes, the design of downsampling and upsampling units and so on. Extensive experiments are necessary for performance evaluation. (2) Typically, the spatial redundancy reduction contributes to high coding

efficiency and classical digital coding theory has revealed it. However, few studies have discussed reducing spatial redundancy for learning-based compression. (3) Perceptual quality study and comparison with traditional MPEG hybrid algorithms has not been conducted.

This thesis aims to utilize deep learning approaches to propose a novel image quality assessment approach with high accuracy and low complexity and an efficient image compression approach with high coding performance. Our main contributions can be summarized into three aspects. (1) To address the above-mentioned two problems of IQA, a fully-blind and fast image quality predictor using convolutional neural networks is proposed to achieve high accuracy and low-complexity for all distortion types. This approach is effective for both natural scene images and screen content images. (2) To design a neural network for learning-based image compression, I compare the performance of generative adversarial networks, super-resolution and convolutional autoencoder. After that, a lossy image compression based on convolutional autoencoder is proposed with principle component analysis to de-correlate the feature maps across channels. Besides, the energy compaction property for non-linear neural networks are analyzed to add a penalty term into loss function. A learning-based novel image compression through energy compaction is proposed to outperform the compression standard HEVC intra in terms of quality metric MS-SSIM. (3) To extend the image compression to the video compression, I propose a spatial-temporal energy compaction strategy for videos, by using an interpolation loop and adaptive interpolation period selection, based on the motion characteristics of sequences. Together with learned image compression networks, my method achieves comparable coding performance with H.264.

The thesis is composed of five chapters.

In Chapter 1 [**Introduction**], the background and motivation to select deep learning-based image compression and quality assessment as my research topics are firstly introduced. Besides, the merits and demerits of applying deep learning to compression and quality assessment is discussed in detail. Next, main contributions during my doctoral course are explained. Finally, the organization of this thesis is present.

In Chapter 2 [**Learning-based Image Quality Assessment through Convolutional Neural Networks**], to achieve a high accuracy image quality prediction network, I have proposed a novel method by applying the usage of convolutional neural networks (CNNs) to image quality assessment (IQA). Compared with the existing conventional IQA methods, experimental results are very appealing in terms of both accuracy and complexity performance. This chapter includes three aspects. First, by analyzing the relationship between image saliency information and CNN prediction error, I utilize a pre-saliency map to skip the non-salient patches for IQA acceleration.

Second, I propose a distortion clustering strategy based on the distribution function of intermediate-layer results of CNNs to make the quality assessment fully blind. Third, I extend this IQA method from natural scene images to screen content images, which are the mixture of natural images, computer graphics, texts, documents and other components. Experimental results demonstrate that our method can achieve the high accuracy (0.935-0.978) with subjective quality scores, outperforming existing IQA methods. Moreover, the proposed method is highly computationally appealing, achieving a time reduction of 52.7% compared with conventional IQAs without the pre-saliency map. The proposed method also achieves high accuracy for screen content image dataset with subjective quality scores.

In Chapter 3 [**Learning-based Image Compression through Convolutional Autoencoder**], to present a learning-based image compression architecture, at first, I compare the performance of convolutional autoencoder (CAE), Generative adversarial network (GAN), and super resolution for image compression and propose a deep residual learning architecture. The results are submitted to CVPR challenge CLIC 2018 and 2019. Based on this comparison results, I select the convolutional autoencoder as our base architecture. Then I design a symmetric CAE structure with multiple down-sampling and up-sampling units to generate feature maps with low dimensions. this CAE is optimized using an approximated rate-distortion loss function. To generate a more energy-compact representation, I propose a principal components analysis (PCA)-based rotation to generate more zeros in the feature maps. Then, the quantization and entropy coder are utilized to compress the data further. Experimental results demonstrate that our method outperforms JPEG and JPEG2000 in terms of PSNR and achieves a 13.7% BD-rate decrement compared to JPEG2000 with the popular Kodak database images. However, PCA cannot be incorporated into end-to-end learning process, I further analyze the energy compaction property of CAE, and formulate the optimum bit allocation problem as a normalized coding gain metric, which measures the compression capability from the viewpoint of the energy compaction property. Based on the proposed metric, an energy compaction-based bit allocation method is present by adding a penalty term to the loss function, and the CAE training strategy is given to achieve a larger coding gain. The experimental results demonstrate that our proposed bit allocation method can maximize the coding gain and achieve higher coding efficiency in comparison with other bit allocation methods. Moreover, our method achieves significantly better MS-SSIM in comparison with the image compression standard HEVC-intra (BPG). Moreover, it outperforms state-of-the-art learning-based compression methods at high bit rate. Finally, a thorough perceptual quality study on deep learning-based approaches and traditional coding standards are conducted to validate the high subjective quality of reconstructed images by our

proposed method.

In Chapter 4 [**Learning-based Video Compression through Interpolation Network**], based on learning-based image compression algorithm, I extend the learned image compression to learned video compression, by considering not only spatial energy compaction, but also temporal energy compaction. Including the autoencoder for learning-based image compression, I append an interpolation loop to learn the temporal correlation of neighboring frames. The group of pictures size is adaptively selected according to the motion characteristics of input sequences. This mechanism is motivated by bi-directional prediction frames. Results illustrate the performance of proposed learning-based video compression is comparable with the video coding standards H.264. Visualization results have also validated the visually pleasant results generated by learning based codec compared to classical video coding standards.

In Chapter 5 [**Conclusions and Future Work**], I present the conclusion and future work for my doctoral thesis. As for the future work, there are four directions along two dimensions, i.e. lossy/ lossless and image/video. As for lossy image compression, I can continue working on it to outperform versatile video coding (VVC) intra. Regarding lossy video compression, one possible topic is frame prediction, targeted at compression of video P-frames. The goal is to outperform video coding standards HEVC or VVC. The other two directions are lossless image/video compression. Recently some works, such as PixelCNN and learning-based probability models, have been proposed. I can learn from them to achieve better performance for lossless compression.

The key contribution of this thesis is to utilize deep learning to design novel approaches for image compression and quality assessment. The novelty of this thesis is threefold. First, I present a fully-blind and fast image quality predictor using convolutional neural networks in Chapter 2, to achieve high accuracy with low-complexity. Second, I propose a learning-based image compression based on convolutional autoencoder in Chapter 3. I use a principle component analysis (PCA) to generate more zeros for feature maps or analyze optimal bit allocation problem to add an energy-compact term into loss function. My method is better than previous methods in terms of MS-SSIM. Third, I extend it to a learning-based video compression in Chapter 4, to achieve comparable coding efficiency with H.264.

ACKNOWLEDGMENTS

Firstly, I would like to show my great gratitude to my respected supervisor Professor Jiro Katto for providing a valuable opportunity to study under his supervision at Waseda University. He shows great interest in the latest technology, and always follows the emerging technical development of image processing fields and machine learning fields. He always encourages me to explore the research topics and gives me very helpful guidance on my research works. His attitude towards research will have a deep impact on me during my whole life.

Secondly, I would like to express my thanks to my deputy supervisor Professor Hiroshi Watanabe for his constant guidance and valuable suggestion on my research. He is always energetic and very professional in the multimedia research areas. Every year I will report my research progress to Prof. Watanabe, and he can quickly give me really helpful comments and advice on my next research direction. I also would like to thank Professor Wataru Kameyama to be my referee. He is also very experienced in MPEG activities and Multimedia system to give me very helpful comments on my research results through my doctoral program.

Thirdly, I would also thank other advisors, including Professor Shinji Kimura from Graduate School of IPS at Waseda University, Professor Xin Jin from Tsinghua University, Dr. Sei Naito from KDDI research. They not only check my research results through Qualifying Examination but also give me valuable guidance from the different viewpoints. Without their guidance, I could not continue working to achieve new research results.

Next, I would like to thank Professor Ebrahimi Touradj at EPFL, who is my supervisor during my overseas internship. His guidance and advice have a great impact on my research. Besides, I would like to thank the laboratory members Akyazi Pinar, Viola Irene, Alexiou Evangelos, Upenik Evgeniy. We become good friends and discuss our research areas freely. Without them, I cannot have a good stay during my overseas stay.

Furthermore, I want to thank our lab members, Kenji Kanai-san, Heming Sun-san, Masaru Takeuchi-san, and other students. Our laboratory has a good environment to work together, exchange ideas and discuss freely on weekly seminar.

Additionally, I would like to express great thanks to Graduate Program for Embodiment Informatics, which gives me many valuable opportunities, such as experiencing an overseas intern, reaching different research fields through joint project in Kobo, interacting with students from Tsukuba University through Summer school, participating in English training program at

UC Davis, and so many other activities. I also want to thank all the professors and students involved by leading program. Without them, I could not enjoy my doctoral program and learn so many different kinds of knowledge.

Last, I want to thank my parents for their constant support and understanding.

Contents

Table of Contents	ix
1 Introduction	1
1.1 Research Background	1
1.2 Thesis Organization	4
2 Learning-based Image Quality Assessment through Convolutional Neural Networks	7
2.1 Related Work	7
2.1.1 Natural Scene Images (NSIs)	8
2.1.2 Screen Content Images (SCIs)	10
2.2 Proposed Pre-saliency Map method	11
2.2.1 Patch-level CNN Design	11
2.2.2 Pre-Saliency Map based Quality Aggregation Algorithm	13
2.3 Proposed Distortion Clustering strategy	16
2.3.1 CNN design for distortion recognition	17
2.3.2 Posterior Observations on distribution of intermediate layers	17
2.3.3 K-means based Distortion Clustering	19
2.4 Proposed Screen content image-targeted method	20
2.4.1 Patch-level CNN Design	20
2.4.2 SCIs-oriented Quality Aggregation Acceleration	21
2.5 Experimental Results	24
2.5.1 Effect of CNN Structure	25
2.5.2 Performance Evaluation on NSIs-targeted Approach	26
2.5.2.1 Accuracy on TID2008 dataset	27
2.5.2.2 Accuracy on LIVE dataset	28
2.5.2.3 Complexity Performance Analysis	29
2.5.2.4 Cross Database Validation	31
2.5.3 Performance Evaluation on SCIs-targeted Approach	31

2.5.3.1	Accuracy Evaluation	33
2.5.3.2	Complexity Evaluation	34
2.6	Chapter Summary	35
3	Learning-based Image Compression through Convolutional Autoencoder	37
3.1	Related Work	37
3.1.1	Hand-crafted Image Compression	38
3.1.2	Learning-based Image Compression	39
3.2	Architecture Discussion	41
3.2.1	Performance Comparison of CAEs, GANs and SR-based approaches . . .	41
3.2.1.1	Convolutional Autoencoders for Compression	41
3.2.1.2	Generative Adversarial Networks for Compression	42
3.2.1.3	Super-Resolution for Compression	44
3.2.1.4	Comparison Results	46
3.2.2	Deep Residual Learning for Learned Image Compression	46
3.2.2.1	From Small Kernel Size to Large Kernel Size	47
3.2.2.2	From Shallow Network to Deep Residual Network	49
3.2.2.3	Upsampling Operations at Decoder Side	50
3.2.2.4	Implementation Details	50
3.2.2.5	Compression Results	53
3.3	Proposed Learned Image Compression through Principle Component Analysis (PCA)	54
3.3.1	Proposed Convolutional Autoencoder (CAE) Network	55
3.3.2	Principle Component Analysis (PCA) Rotation	58
3.4	Proposed Learned Image Compression through Energy Compaction	60
3.4.1	Convolutional Autoencoder (CAE) Architecture	60
3.4.2	Proposed Coding Gain Metric based on Mathematical Analysis of Energy Compaction Property	63
3.4.3	Proposed Energy Compaction-based Bit Allocation Method	68
3.5	Experimental Results	70
3.5.1	Experimental Results of Image Compression through PCA	70
3.5.1.1	Coding Efficiency Performance	71
3.5.1.2	Complexity Performance	72
3.5.2	Experimental Results of Image Compression through Energy Compaction .	73
3.5.2.1	Network Architecture	75
3.5.2.2	Different Bit Allocation Methods	77
3.5.2.3	Performance Comparison	83

3.5.2.4	Visualization	83
3.6	Perceptual Quality Study on Learning based Image Compression	86
3.6.1	Codec Architecture	86
3.6.2	Subjective Quality Evaluations	88
3.6.2.1	Dataset	88
3.6.2.2	Test Methodology	89
3.6.3	Results and Discussion	90
3.7	Chapter Summary	93
4	Learning-based Video Compression through Interpolation Network	95
4.1	Related Work	95
4.1.1	Hand-crafted Video Compression	96
4.1.2	Learning-based Video Compression	97
4.2	Proposed Interpolation Loop	97
4.2.1	Problem Formulation	97
4.2.2	Interpolation Loop	99
4.3	Proposed Spatial-Temporal Energy Compaction for Video Compression	100
4.4	Implementation Details	101
4.4.1	Dataset	101
4.4.2	Training Details	102
4.4.3	Measurements	103
4.5	Experimental Results	103
4.5.1	Frame-level Results	103
4.5.2	Performance Comparison	104
4.5.3	Visualization	105
4.6	Chapter Summary	107
5	Conclusions and Future Work	109
5.1	Conclusions	109
5.2	Future Work	110
	Bibliography	113
	List of Figures	123
	List of Tables	127
	Publications Lists	129

My View of Embodiment Informatics

133

Chapter 1

Introduction

1.1 Research Background

Image compression and image quality assessment are very fundamental and important research topics in the field of signal processing, due to their significant influence on data transmission, storage and perceived quality by users. Besides, there is a strong correlation between these two topics, because subjective quality assessment is one of important criterions for many multimedia processing tasks, including compression task. Therefore, how to evaluate the subjective quality perceived by human beings accurately and how to compress images with high subjective quality efficiently are two important topics that I would like to focus during my doctoral course.

In the past decades, image/video compression and image quality assessment have witnessed great progress through many research works. For instance, many image compression standards have been proposed, such as JPEG, JPEG2000, BPG (Better Portable Graphics), which uses the intra-coded HEVC algorithms. Typically they rely on the hand-crafted encoder-decoder (codec) architecture, and use fixed and linear transform matrix, such as the discrete cosine transform (DCT) or the discrete wavelet transform (DWT), together with a quantization and an entropy coder. Good compression performance has been achieved through efforts spanning several decades. Besides, there are many widely-used quality metrics, such as PSNR, SSIM, MS-SSIM and so on. However, along with the fast development of multimedia applications, the proliferation of high-resolution devices and the emergency of novel image formats, existing approaches are gradually facing with some problems and challenges. Current quality metrics are not fit for subjective human visual systems and need undistorted reference images, which sometimes do not exist in some applications. Existing compression standards are not expected to be a general,

perceptual and optimal compression solution for all types of image content.

Recently, deep learning has grown rapidly and has been successfully applied into many computer vision tasks. Meanwhile, deep learning has also attracted great attention from signal processing community, especially on image compression and quality assessment. Specifically, deep learning-based image quality assessment has attracted increased attention, owing to higher prediction accuracy, compared to classical methods. On the other hand, along with prospective larger image data by emerging super high-resolution (8K/16K), 360-degree and virtual reality (VR), traditional compression standards are faced with the bottleneck of compression performance. Recently, ISO/IEC Motion Picture Experts Group (MPEG) has launched the next-generation coding standard as Versatile Video Coding (VVC), which will be finalized by the end of 2020, and some deep learning-based coding tools are discussed to be used in VVC partially in MPEG meeting. Meanwhile, top conference CVPR in the field of computer vision has also held the world-wide event workshop and challenge on learned image compression (CLIC) since 2018 to call for proposals. Deep learning-based image compression topic has attracted great attention from many academic and industrial organizations, such as Google, Twitter, ETH and so on. Therefore, benefiting from the rapid development of deep learning, the usage of deep learning in next generation image/video compression and image quality assessment is an appealing trend. However, deep learning-based image/video compression and quality assessment have not been fully investigated yet. Adopting deep learning techniques to compression and quality assessment can solve some problems, but bring new critical issues. The merit and demerit of adopting deep learning into these topics are listed in Fig. 1.1.

Many issues need to be addressed for image compression and quality assessment, respectively. In this thesis, we focus on some of these problems. Specifically, regarding image quality assessment (IQA), two main problems are the high-complexity of convolutional neural networks (CNN) approaches and the unknown distortion types for CNN models. In details, conventional image quality prediction methods use hand-crafted features from natural scene images, which are not efficient, while CNN-based methods are more efficient but brings high complexity. On the other hand, no-reference image quality assessment is blind to different distortion types, but one pre-trained CNN model is difficult to achieve good results for all the distortion types, such as Gaussian noises, different blurring distortions, different compression distortions and so on.

Regarding the learned image/video compression, which is an emerging topic, we would like to focus on three issues, such as the selection of many neural network architectures, spatial redundancy reduction and rigorous comparison when developing end-to-end learning-based compression approaches. The detailed explanations are as followed. Firstly, the selection of

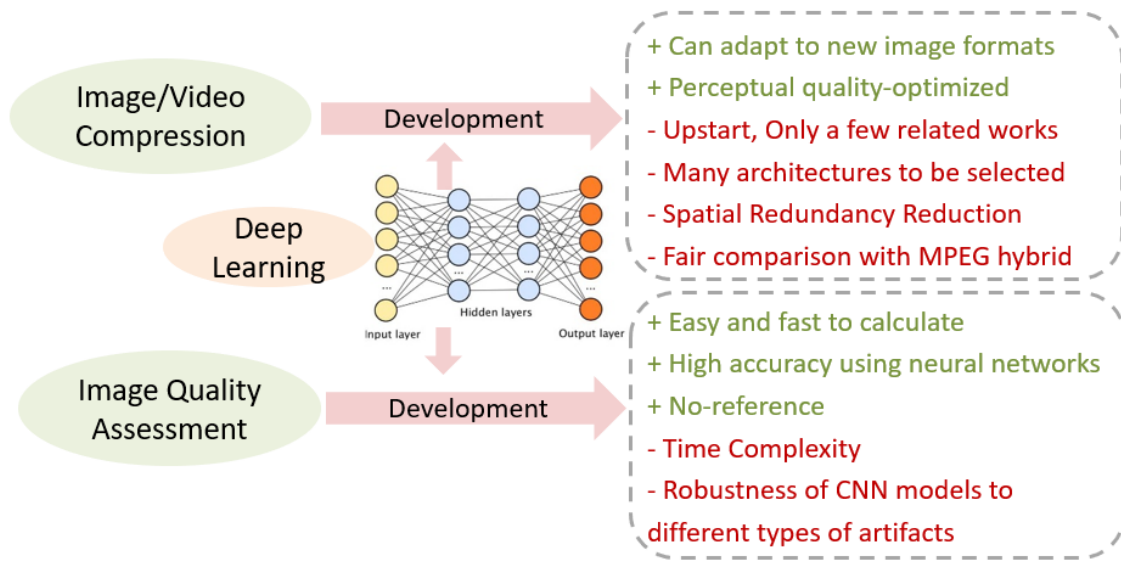


Fig. 1.1 The merit and demerit of adopting deep learning to compression and quality assessment

neural network not only refers to the types of neural networks, including generative adversarial networks (GANs), convolutional autoencoder(CAE) and Super-resolution based approaches, but also includes the structure of neural networks, such as the number of convolutional layers, the number of feature maps for compressed codes, the design of downsampling and upsampling units and so on. Extensive experiments are necessary for performance evaluation. Secondly, the spatial redundancy reduction contributes to high coding efficiency and classical digital coding theory has revealed it. However, few studies have discussed reducing spatial redundancy for learning-based compression. Third, deep learning based approach can use MS-SSIM as distortion term, but traditional compression algorithms usually use Sum of Square Difference (SSD) for rate-distortion optimization, which is corresponding to PSNR. A fair and thorough perceptual quality study and comparison with traditional MPEG hybrid algorithms has not been conducted.

This thesis aims to propose a novel image quality assessment approach with high accuracy and an efficient image compression method with high coding performance by utilizing the deep learning approaches. The main contributions of this thesis can be summarized into three aspects.

- Learning-based Image Quality Assessment. I present a fully-blind and fast image quality predictor using convolutional neural networks by two strategies, i.e. pre-saliency map and distortion clustering. These approaches achieve high accuracy with low-complexity and are blind for all the distortion types for both natural scene images and screen content images.
- Learning-based Image Compression. I compare three architectures for learned image

compression. Based on that, I propose a lossy image compression based on convolutional autoencoder. To generate an energy-compact representation for compression, I use a principle component analysis to generate more zeros for feature maps or analyze the bit allocation problem to add a energy-compact term into loss function. The coding efficiency of my method is better than previous methods in terms of MS-SSIM and my method achieves better subjective quality than previous methods. A rigorous perceptual quality study has been conducted through subjective quality evaluation.

- Learning-based Video Compression. I extend it to a learning-based novel video compression through spatial-temporal energy compaction and incorporate an interpolation network to reduce the temporal redundancy. My approach achieves better coding efficiency than H.264 and recent learning-based approaches in terms of MS-SSIM. Besides, the visual quality is also better than other video codecs.

1.2 Thesis Organization

The thesis is composed of five chapters.

In Chapter 1 [**Introduction**], the background and objectives on deep learning-based image compression and quality assessment are firstly introduced, by giving a brief review on merits and demerit of adopting deep learning techniques into these topics. After that, the organization of this thesis is present.

In Chapter 2 [**Learning-based Image Quality Assessment through Convolutional Neural Networks**], to achieve a high accuracy image quality prediction network, I have proposed a novel method by applying the usage of convolutional neural networks to image quality assessment (IQA). Compared with the existing conventional IQA methods, experimental results are very appealing in terms of both accuracy and complexity performance. This method includes three aspects. First, I propose a distortion clustering strategy based on the distribution function of intermediate-layer results in the convolutional neural network (CNN) to make IQA fully blind. Second, by analyzing the relationship between image saliency information and CNN prediction error, I utilize a pre-saliency map to skip the non-salient patches for IQA acceleration. Third, I also extend this IQA method from natural scene image to screen content images, which are the mixture of natural images, computer graphics, texts, documents and other components. Experimental results demonstrate that our method can achieve the high accuracy (0.960-0.978) with subjective quality scores, outperforming existing IQA methods. Moreover, the proposed method is highly computationally appealing, achieving a time reduction of 52.7% compared with

conventional IQAs without the pre-saliency map. The proposed method also achieves high accuracy for screen content image dataset.

In Chapter 3 [**Learning-based Image Compression through Convolutional Autoencoder**], In the beginning, I compare the performance of convolutional autoencoder (CAE), Generative adversarial network (GAN), and super resolution for image compression. The results are submitted to CVPR workshop CLIC. Based on this comparison results, I select the convolutional autoencoder as our base architecture. Then, I present a novel CAE structure with multiple down-sampling and up-sampling units to generate feature maps with low dimensions. this CAE is optimized using an approximated rate-distortion loss function. To generate a more energy-compact representation, I propose a principal components analysis (PCA)-based rotation to generate more zeros in the feature maps. Experimental results illustrate that my method outperforms JPEG and JPEG2000 in terms of PSNR and achieves a 13.7% BD-rate decrement compared to JPEG2000 with the popular Kodak database images. However, PCA cannot be incorporated into end-to-end learning process, I further analyze the energy compaction property of CAE, and formulate the optimum bit allocation problem as a normalized coding gain metric, which measures the compression capability from the viewpoint of the energy compaction property. Based on the proposed metric, an energy compaction-based bit allocation method is present by adding a penalty term to the loss function, and the CAE training strategy is given to achieve a larger coding gain. The experimental results validate that our proposed bit allocation method can maximize the coding gain and achieve higher coding efficiency in comparison with other bit allocation methods. Moreover, our method achieves significantly better MS-SSIM in comparison with the image compression standard HEVC-intra (BPG). Moreover, it outperforms existing learning-based compression methods at high bit rate. A perceptual quality study on deep learning-based approaches and traditional coding standards are conducted to validate the high subjective quality of reconstructed images by our proposed method.

In Chapter 4 [**Learning-based Video Compression through Energy Compaction**], based on learned image compression algorithm, I extend the learned image compression to learned video compression. Not only the spatial energy compaction strategy, but also spatial-temporal energy compaction, I proposed a deep learning-based novel image and video compression. Including autoencoder, I append an interpolation network to learn the spatial correlation of neighboring frames. The group of pictures size is adaptively selected according to the temporal energy correlation. Experiments show that the performance of proposed learning-based video compression is much better than MPEG-4, on par with the video coding standards H.264, and sometimes can reach the performance of HEVC for some HEVC test sequences. Visualization results have also validated the visually pleasant results generated by learning based codec

compared to classical video coding standards.

In Chapter 5 [**Conclusion and Future Work**], I present the conclusion and future work for my doctoral thesis. As for the future work, there are three directions. One is the algorithm enhancement of lossy compression, by continuing working on image compression to outperform versatile video coding (VVC) intra, and working on video compression to outperform existing standards. Second is to develop efficient neural network-based lossless compression. By exploring the PixelCNN and other deep learning-based probability models, I can achieve state-of-the-art performance for lossless image/video compression and even extend it to image generation. Third is frame prediction for videos, targeted at compression of video P-frames.

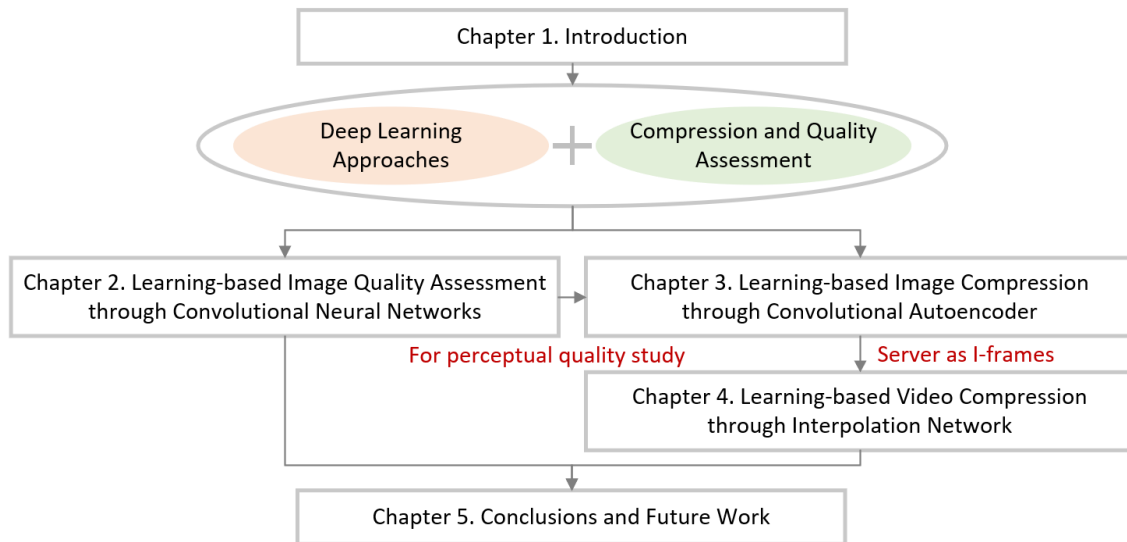


Fig. 1.2 The overall diagram and relation of contents for different chapters

Figure 1.2 shows the relationship between different chapters. The key concept is to utilize deep learning approaches to enhance the image/video compression and image quality assessment. The knowledge of Chapter 2 can help to conduct the perceptual quality study on different compression algorithms, and the image compression algorithm of Chapter 3 can serve as I-frames in the video compression algorithms.

Chapter 2

Learning-based Image Quality Assessment through Convolutional Neural Networks

In this chapter^{}, to achieve image quality assessment with high accuracy, I discuss the natural scene images and screen content images in this chapter, separately. For nature scene images, I propose a fully-blind and fast image quality predictor using convolutional neural networks, which mainly consists of two strategies. One is called distortion clustering strategy. This strategy is proposed based on the distribution function of the intermediate layer of the CNNs to make IQA achieve better performance for different kinds of distortion artifacts. Second is pre-saliency map. After analyzing the relation between saliency map and CNN prediction errors, I accelerate the IQA to adaptively apply CNN computation and assign weights for non-skipped patches. Experimental results demonstrate our proposed method achieve high correlation coefficient with subjective quality scores, and outperform previous IQA methods. Besides, by taking advantage of pre-saliency map, the proposed method can achieve low time complexity. For screen content images, I modify these strategies to make them also work for SCIs by considering the textural regions. My results can also achieve better accuracy and flexible complexity among existing methods.*

2.1 Related Work

Along with the rapid development in multimedia technology, large amount of visual medias consume large traffic share on social network. We have to face to a situation where the end-user expects high subjective image quality available. Therefore, image quality assessment (IQA) is

^{*}This chapter is adapted from the work published in [1], [2], [3].

becoming a popular research topic for both academic and industrial progress.

According to the availability of reference images, IQA can be divided into three categories, full reference (FR) IQA, reduced-reference (RR) IQA and no-reference (NR) IQA or Blind IQA. The most representative FR-IQAs are very widely used, include PSNR, SSIM [4], VIF [5], FSIM [6] and VSI [7]. RR-IQA will process the extracted information from the reference image, instead of the reference image itself. RR-IQA lies between FR and NR, such as [8–10], which obtains reduced information from reference images to estimate the quality of distorted images. However, typical services will not provide the ideal original reference image to end users. Thus, various approaches have been proposed to investigate the blind IQA. In [11], a distortion identification based image verity and integrity evaluation (DIIVINE) was proposed by utilizing natural scene statistical properties. In [12], the authors proposed a blind natural scene statistical approach (BLIIND-II) using the DCT domain. The BRISQUE algorithm was proposed without the transformation, but used scene statistics of locally normalized luminance coefficients to quantify the distortion in [13]. In [14], the CORNIA algorithm was proposed by using the raw patch as local descriptor and obtaining a learning dictionary to measure the distortion. In [15], Min et al. introduced a pseudo most distorted images to measure the quality loss without the reference images. In this work, our proposed method is classified as NR-IQA.

In the following, I will discuss the related works for natural scene images and screen content images, separately.

2.1.1 Natural Scene Images (NSIs)

Generally, conventional features are hand-crafted features from natural scenes images (NSIs) as shown in Fig. 2.1, which are not efficient. Recently, convolutional neural network (CNN) achieves success in various computer vision tasks and can be applied in IQA. In [16], the author firstly applied CNN into the quality estimation and calculated the image score by averaging the predicted patch scores, which is not fit for interest of regions for human perception. Some works designed a deep CNN for one complete image with 224×224 pixels in [17] and [18], which involved many convolutional layers. Some works applied existing image classification CNN to fine-tune the parameters in [19] and [20]. The work [21] used the saliency detection as the weighted mask for patches. Zuo et al. [22] applied different weights for text regions and picture regions for screen images. However, most works only achieve high accuracy with subjective quality scores, but brings high complexity, which is not fit for real-time quality monitoring.

Moreover, each type of image distortion has its own distortion-specific features, unavoidably resulting in training different parameters in CNN. Most works did not consider the unknown



Fig. 2.1 Examples of some NSIs in LIVE database [35]

distortion types except [19] [23] [24]. Sun et al. [19] combined the deep CNN and complex local content features to achieve the robustness to distortion types. Kim et al. [23] took advantage of the combination of full-reference IQA methods and CNN to deal with the different distortion types. Wang et al. [24] proposed a linear local information model and a distortion-specific compensation strategy to make CNN offset the effect of different distortion types. However, distortion-specific CNN training can achieve high accuracy, but not blind. Thus, it motivates us to give a distortion clustering strategy for grouping similar distortion types before CNN.

Saliency models have been investigated in previous studies [25] [26] to describe the extent to which a local region can attract the attention from human visual system [7] [27], thus, it motivates us to incorporate visual saliency models to IQA. The most representative saliency models are [25] and [26]. Itti in [25] proposed a saliency-based visual map by using the color, intensity, orientation information to generate multiple features in a dynamical neural network. J. Harel in [26] proposed a saliency map based on graph computation to highlight the conspicuous region. Incorporating the visual saliency model to IQA is reasonable for human visual system, however, the investigation has not been fully studied. Although [19] [21] applied saliency map into IQA, they did not study the relation between saliency map and CNN prediction errors. Different from them, I will not only use saliency map as the weight, but also use the saliency map to reduce the computation redundancy of CNN for acceleration.

2.1.2 Screen Content Images (SCIs)

The above related works aim at general images, but can not achieve high accuracy for screen content images (SCIs), because general images mainly consist of pictures, while SCIs contain both texts and pictures as shown in Fig. 2.2. Several SCIs-oriented IQA methods have been proposed in [22, 28–31]. In [22], Zuo et al. firstly applied convolutional neural networks (CNNs) to SCIs and assigned different weights for textual regions and picture regions to improve the accuracy significantly. In [28], a saliency guided quality assessment was proposed by analyzing the fixation and saccade in SCIs. In [29], Ni et al. obtained edge contrast map and edge width map to measure the distortion. In [30], luminance and texture features were extracted to rate the quality. In [31], Zhou et al. used a local and global dictionary to learn the features. These works mainly extracted features using conventional image processing methods and used a support vector regression (SVR) for quality aggregation. Therefore, CNN can be expected to enhance the performance of SCIs quality prediction.



Fig. 2.2 Examples of some SCIs in SIQAD database [37]

Based on the review of previous works, therefore, this chapter will discuss the natural scene images and screen content images separately. Section 2.2 and Section 2.3 discuss the proposed strategies for natural scene images, including pre-saliency map and distortion clustering strategies.

Section 2.4 discusses the image quality assessment for screen content images, by extending the above two strategies to SCIs and achieving high accuracy than related works. Section 2.5 discusses the performance. Section 2.6 gives a summary for this chapter.

2.2 Proposed Pre-saliency Map method

As Fig.2.3, the overall proposed IQA for natural scene images using pre-saliency map mainly includes two steps. Firstly, I split the image into patches to train a CNN for patch-level quality prediction, which is served for the following image quality aggregation. Secondly, by analyzing the relation between saliency information and prediction error of the CNN, I apply pre-saliency map in prior of CNN to skip non-salient patches and assign weights. Thus, in this section, part A) introduces the patch-level CNN design. Part B) gives a detailed description on quality aggregation algorithm.

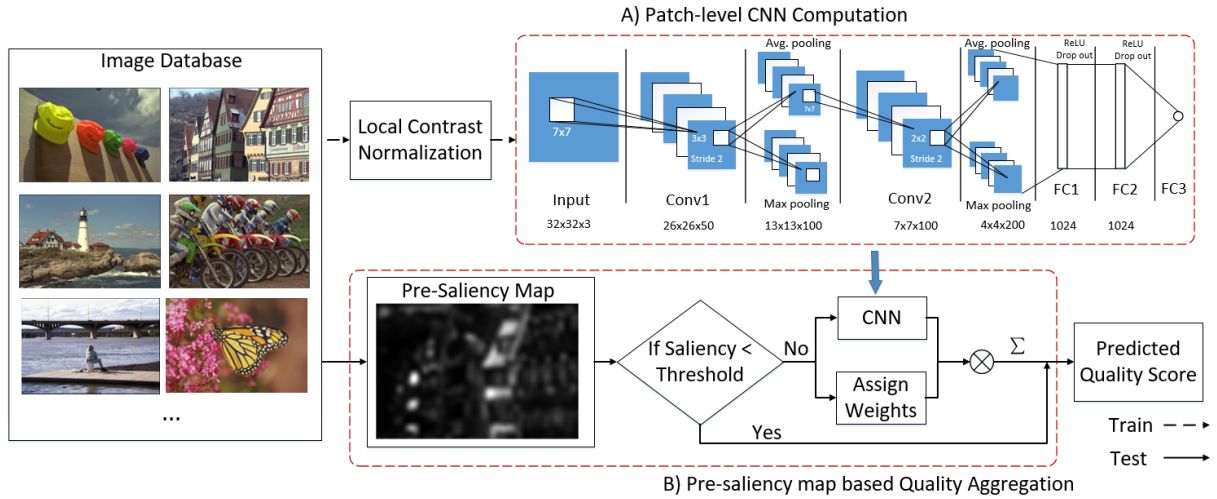


Fig. 2.3 The Proposed IQA framework for natural scene images using pre-saliency map.

2.2.1 Patch-level CNN Design

The CNN architecture is shown in Fig.2.3, which is a $32 \times 32 \times 3 - 26 \times 26 \times 50 - 13 \times 13 \times 100 - 7 \times 7 \times 100 - 4 \times 4 \times 200 - 1024 - 1024 - 1$ structure. Since CNN processes the data with fixed input size, I split the raw images into 32×32 patches with three RGB components. Generally speaking, only one subjective quality score can be obtained for each image when conducting the subjective quality tests, thus the ground-truth quality score for each patch is not available. I assume the labels of patches are equal to quality scores of corresponding images, which is used to train the CNN models.

Since the neural networks are usually sensitive to the mean value of input data, a local contrast normalization is used to pre-process the input pixel. Given an image, the normalized pixel is computed by local mean subtraction and divisive normalization [13].

$$\hat{X}(i, j) = \frac{X(i, j) - \mu(i, j)}{\sigma(i, j) + C} \quad (2.1)$$

where (i, j) are the position of the pixel. M, N are the height and width of images. C is a constant to prevent the case where the denominator is zero ($C = 1$). The mean and variance are calculated by

$$\mu(i, j) = \sum_{k=-K}^K \sum_{l=-L}^L \omega_{k,l} X_{k,l}(i, j) \quad (2.2)$$

$$\sigma(i, j) = \sqrt{\sum_{k=-K}^K \sum_{l=-L}^L \omega_{k,l} (X_{k,l}(i, j) - \mu(i, j))^2} \quad (2.3)$$

where $\omega_{k,l}$ is a 2D circularly-symmetric Gaussian weighting function sampled out to 3 standard deviations and rescaled to unit volume [13]. Here I set K, L as 3 to guarantee window size quite small, since performance decreases when the window size becomes fairly large in [13].

The architecture of CNN is determined by conducting the experiments. The effect of the kernel size and the number of feature maps on the accuracy which is discussed in experimental results. Firstly, the normalized patches are convolved with 50 kernels with the size of 7×7 and each kernel generates a feature map. Then a feature map is pooled into one max value and one average value. The pooling size is 3×3 with a stride of 2. Then a second convolution layer using 100 kernels with the size of 7×7 is applied. The size of the second pooling layer is 2×2 with a stride of 2. Two fully connected layers (FC1, FC2) with 1024 nodes come after convolution and pooling. As for activation function, rectified linear unit (ReLU) function is proved to have better training performance than sigmoid or tanh activation function, thus I use ReLU in FC1 and FC2. Note ReLU only allows the non-negative values to pass through the activation function, thus I do not use ReLU but use linear neurons in convolutional and pooling layers since I have many negative features after normalization. Dropout is a technique to prevent over-fitting and I apply dropout after FC1 and FC2 with a ratio of 0.5. The last layer outputs a one-dimensional quality score, as a predicted value.

The objective function is defined by

$$J(\theta; x^{(i)}, y^{(i)}) = \frac{1}{2N} \sum_{i=1}^N \|f(\theta; x^{(i)}) - y^{(i)}\|^2 \quad (2.4)$$

where $J(\theta; x^{(i)}, y^{(i)})$ is a objective function and I use the euclidean loss. $x^{(i)}$ represents the input

image and $y^{(i)}$ denotes the ground-truth score from training set. $f(\theta; x^{(i)})$ denotes the CNN predicted score. N represents the batch size during the training and N is set as 128.

Updating the network weights with momentum is a popular strategy, and the weights update are shown as follows:

$$v = \gamma v + \alpha \nabla_{\theta} J(\theta; x^{(i)}, y^{(i)}) \quad (2.5)$$

$$\theta = \theta - v \quad (2.6)$$

where $\nabla_{\theta} J$ is the gradient of $J(\theta)$. θ is the parameter vector. v is current velocity vector which is of the same dimension as θ . α is learning rate and I keep α at a fixed value of 0.0001. γ is the momentum which determines how many iterations in the previous gradients are incorporated into the current update. We set γ as 0.9 at the training process.

2.2.2 Pre-Saliency Map based Quality Aggregation Algorithm

To prevent the complexity overhead caused by saliency map (SM) calculation, I use a fast SM model from [32] to benefit the quality aggregation. Fast SM in [32] is a simplified model of [25] by only utilizing the color and intensity to generate a single-scale feature map. The average time consumption for one image of different SM methods [25] [26] [32] is shown in Table 2.1. We can observe that fast SM can achieve 89% time reduction. Meanwhile, fast SM can approximately predict the location of salient patches as shown in Fig. 2.4. Since the input of CNN prediction focuses on 32×32 patch instead of each pixel, approximate saliency prediction is sufficient. Accuracy comparison will be given in experimental results.

Table 2.1 Time Comparison between different saliency map methods with LIVE images.

Method	Itti [25]	Harel [26]	Fast SM [32]
Time (s)	0.1208	0.2457	0.0133

According to the human visual system, the distortion in salient regions is more likely to be perceived by human beings than the distortion in homogenous regions (i.e. non-salient regions). Then the image quality can be formulated as

$$\hat{Q} = w_s Q_s + w_h Q_h \quad (2.7)$$

where \hat{Q} is the ground truth of image quality score, Q_s, Q_h are actual quality score in salient regions and homogenous regions, respectively. w_s, w_h are the corresponding weights and satisfy that $w_s \geq w_h$ and $w_s + w_h = 1$. However, due to the lack of ground-truth quality score for each local patch, I

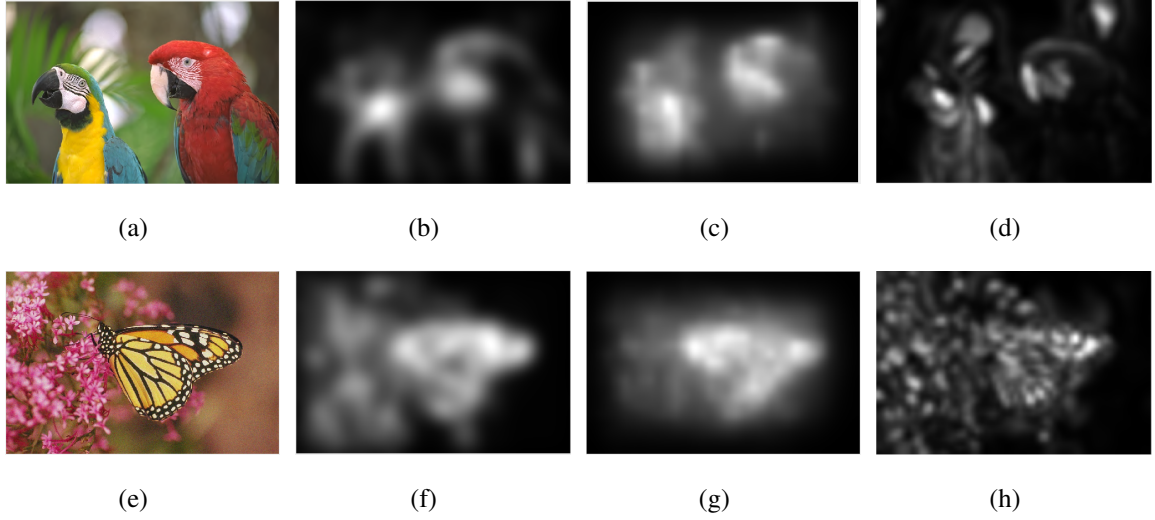


Fig. 2.4 Examples of images and corresponding saliency map: (a)(e)Distorted image "parrots", "monarch", (b)(f)Saliency map using Itti [25], (c)(g)Saliency map using Harel [26], (d)(h)Fast Saliency Map [32].

assume the quality score of each patch is the same as the score of the whole image similar to other CNN-based IQA works. This assumption will definitely lead to different CNN prediction error in salient or homogenous regions. From Eq.(5), I can obtain $(w_s + w_h)\hat{Q} = w_s Q_s + w_h Q_h$, then the absolute value of prediction error (denoted as E_s, E_h) should satisfy

$$\frac{|E_s|}{|E_h|} = \frac{|Q_s - \hat{Q}|}{|Q_h - \hat{Q}|} = \frac{w_h}{w_s} \leq 1 \quad (2.8)$$

This equation implies a priori assumption that the patches in salient regions are likely to have smaller prediction error than that in homogenous regions if CNN can estimate local patch score accurately.

Some statistical analyses are present. Denote the average saliency values for one patch with $N \times N$ pixels is $\tilde{S}(i) = \frac{\sum_{j=1}^{N \times N} S(j)}{N \times N}$, where j denotes each pixel, i denotes each patch. To halve the number of patches, I use the median value of saliency $M(\tilde{S}(i))$ as a threshold to judge whether current patch is in saliency regions (i.e. $\tilde{S}(i) \geq M(\tilde{S}(i))$) or in homogenous regions (i.e. $\tilde{S}(i) < M(\tilde{S}(i))$). Halving the number of patches using median value can guarantee the fairness of statistic analysis. Then I compute the average prediction error in homogenous regions and salient regions for 197 test images from Live database [35]. The results are shown in Fig. 2.5. It can be observed that when the prediction error is small, the difference between salient regions and homogenous regions is not significant. But when prediction error is large, $|E_s|$ is much smaller than $|E_h|$. We calculate the average amplitude of $|E_s|$ and $|E_h|$, which are 8.7496 and 9.9045, respectively. It means the prediction error in saliency regions is 13.2% smaller than that in homogenous regions

on average.

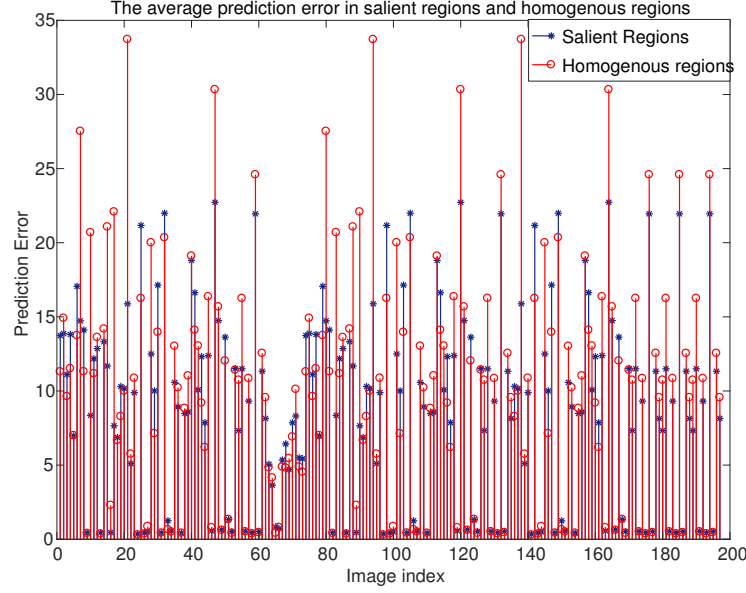


Fig. 2.5 The average prediction error in salient/homogenous regions.

Algorithm 1 Pre-Saliency Map based Quality Aggregation

Input: Tested Image $I(X, Y)$ and Saliency Map $S(X, Y)$

Split I into $N \times N$ patches $\{P_1, P_2, \dots, P_M\}$

while $i \in [1, M]$ **do**

Calculate the average saliency value in P_i :

$$\bar{S}_i = \frac{\sum_{x=1}^N \sum_{y=1}^N S(x, y)}{N \times N}$$

if $\bar{S}_i \leq \varepsilon$ **then**

Skip the CNN computation for P_i ;

else

$$q_i = f_{CNN}(P_i);$$

Assign weights by $\omega_i = \text{Norm}(\sum_{j=1}^{N \times N} S(j))$;

end if

$i = i + 1$;

end while

Final quality score for I is $Q = \frac{\sum_{i=1}^M \omega_i \times q_i}{\sum_{i=1}^M \omega_i}$.

To avoid the large prediction error in homogeneous regions, I propose a pre-saliency map based quality aggregation algorithm to remove some patches with small saliency values, summarized in

Algorithm 3. Firstly, I compute the saliency map values of the patch and adaptively apply CNN computation based on whether the pre-saliency value exceeds a given threshold. If the saliency value is small, current patch is skipped. Otherwise, the patch score is calculated by CNN. After the computations for all the patches are finished, I rescale the saliency value to $[0,1]$ as the weight for each patch, then calculate the weighted sum of predicted patch scores for the final predicted image. Applying pre-saliency map will have two benefits, One benefit is to improve the accuracy by removing the non-salient patches which are likely to have large error. The other benefit is to reduce the computation time by reducing the number of patches which need to be computed.

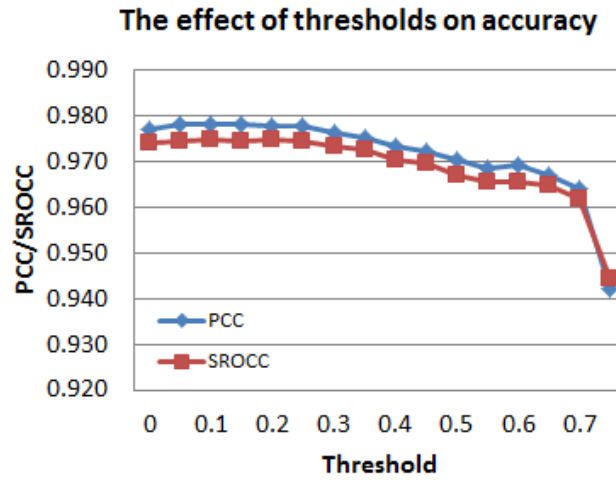


Fig. 2.6 The effect of threshold on IQA accuracy.

The value of threshold is different for different textures of images. To further determine the value of thresholds, I discuss how performance varies with different thresholds, whose range is from 0 to 0.75. The accuracy is evaluated by Pearson correlation coefficient (PCC) and Spearman rank-order correlation coefficient (SROCC). The accuracy performance is shown in Fig. 2.6. It can be observed that PCC and SROCC almost keeps constant and even slightly higher than the baseline ($\epsilon = 0$) when $\epsilon \leq 0.25$, since removing the patches in non-salient regions will not affect the prediction accuracy. When $\epsilon > 0.3$, PCC and SROCC decreases because some patches are falsely skipped. When $\epsilon > 0.7$, PCC and SROCC drops dramatically. We set ϵ as 0.25 for high accuracy. Complexity analysis will be given in experimental results.

2.3 Proposed Distortion Clustering strategy

As previously mentioned, each type of image distortion has its own distortion-specific features. When these features differ a lot from each other or the number of distortion types are large, CNN

cannot predict the quality score accurately for all the types of distortion. To address these difficulties to achieve fully-blind IQA, I propose a distortion clustering strategy to cluster several types of distortion with similar features, so that the IQA accuracy can be further improved.

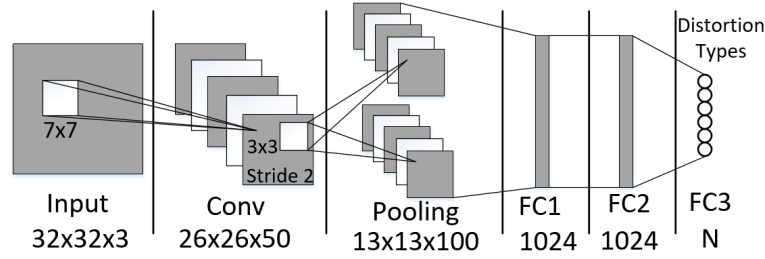


Fig. 2.7 The architecture of CNN for Distortion Recognition.

The distortion clustering strategy consists of three steps, and I will describe the detailed techniques one by one.

2.3.1 CNN design for distortion recognition

The first step is to implement a CNN for distortion recognition. Similarly to IQA-oriented CNN, I use a simplified CNN with only one convolutional layer to reduce the complexity overhead, as shown in Fig. 2.7. In addition, the final layer uses the softmax classifier to identify the type of distortion. We train this CNN using the TID2013 database [34], and validate it in the TID2008 database [36] with the first 13 distortions. The reason for choosing the first 13 types of distortions is that there are 17 types of distortions shared with TID2008, including additive Gaussian noise (WN), additive noise in color components (WNC), spatially correlated noise (SCN), masked noise (MN), high frequency noise (HFN), impulse noise (IN), quantization noise (QN), Gaussian blur (BLUR), image denoising (IDN), JPEG compression (JPEG), JPEG2000 compression (JP2K), JPEG transmission errors (JPEGTE), JPEG2000 transmission errors (JP2KTE), non-eccentricity pattern noise (NEPN), local block-wise distortions of different intensity (LBD), mean shift (MS), and contrast change (CC), and I do not evaluate the last four types because these are highly subjective tasks evaluating with reference images for humans, so that they cannot be handled with no-reference IQA, as indicated in [14]. The experiments show that the classification accuracy is 99.6%, which verifies that CNN can recognize distortion types accurately.

2.3.2 Posterior Observations on distribution of intermediate layers

The second step is posterior observations on the distribution of intermediate layers in the CNN, since our assumption is that CNN can recognize different distortion types by producing

intermediate-layer results with different characteristic statistical properties. The first fully connection layer (FC1) is in the middle of CNN and produces a vector that can be easily observed, thus the histogram of FC1 for a reference image and distorted images is plotted in Fig.2.8.

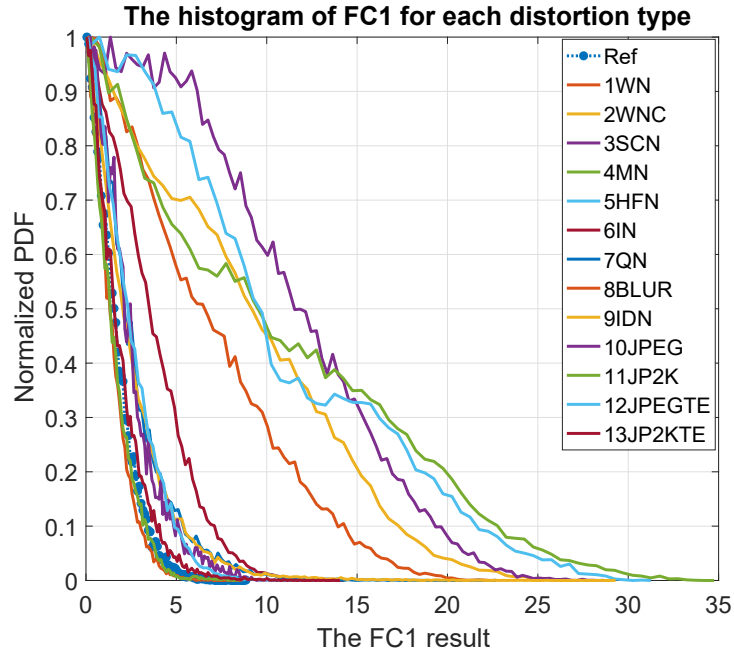


Fig. 2.8 The histogram of FC1 results for a reference image and various distorted versions.

It can be observed that the shape and tail behaviors of each distribution differ significantly. For example, spatially correlated noise and high frequency noise can cause significant tail behavior, while JPEG2000 and Gaussian blur almost maintain the same distribution with reference images. Then, I use a general Gaussian function [33] to describe the distribution function of a distorted image with different shapes and tail behaviors, as defined by:

$$f(x; \alpha, \sigma^2) = \frac{\alpha}{2\beta\Gamma(1/\alpha)} \exp\left(-\left(\frac{|x|}{\beta}\right)^\alpha\right) \quad (2.9)$$

$$\beta = \alpha \sqrt{\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}} \quad (2.10)$$

$$\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt, a > 0 \quad (2.11)$$

where $\Gamma(\cdot)$ is the gamma function. The variable α controls the shape of the distribution. The variable σ^2 represents the variance. The mean value of the distribution is zero, because FC1 is always larger than zero after ReLU, and then I expand it to a symmetric function by assigning $f(-x) = f(x)$.

2.3.3 K-means based Distortion Clustering

The third step is the distortion clustering. We estimate (α, σ^2) using the moment-matching method in [33], and utilize K-means to cluster the distortion types using the values of α and σ^2 . We perform the clustering using 13 types of distortions, and each type includes 100 distorted images from the TID2008 database. The number of clusters is set as four in this experiment. The corresponding clustering figures are shown in Fig. 2.9. Each point corresponds to the distribution function of an image, and its color represents the distortion type. By applying K-means clustering to all the images, the whole space is divided into four regions. The cross mark represents the centroid (mean value) of all points with the same distortion type. If the centroid of a specific distortion type is in the region i , then this distortion type will be in group i . The clustering results are listed in Table 2.2. For an unknown distortion type, I can also find its corresponding cluster.

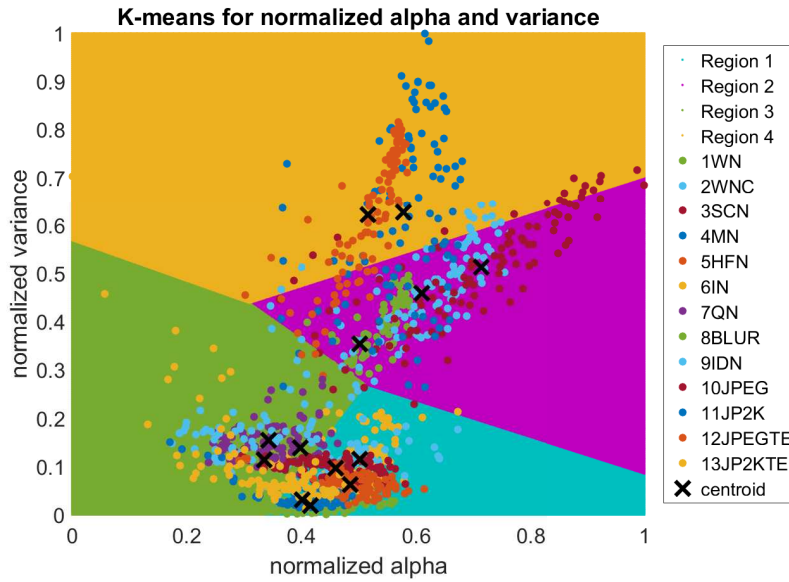


Fig. 2.9 The clustering results of distortion types with K-means.

Table 2.2 The mapping table of distortion types.

Group	Types of Distortion
1	WN, WNC, SCN
2	MN, HFN
3	QN, IDN, JP2KTE
4	IN, BLUR, JPEG, JP2K, JPEGTE

Therefore, by incorporating the distortion clustering strategy into the pre-salienc map method,

the overall of proposed fully-blind method is shown in Fig. 2.10. This proposed method is fully blind and fast image quality predictor, and I denote it as FFIQP in the following experimental results.

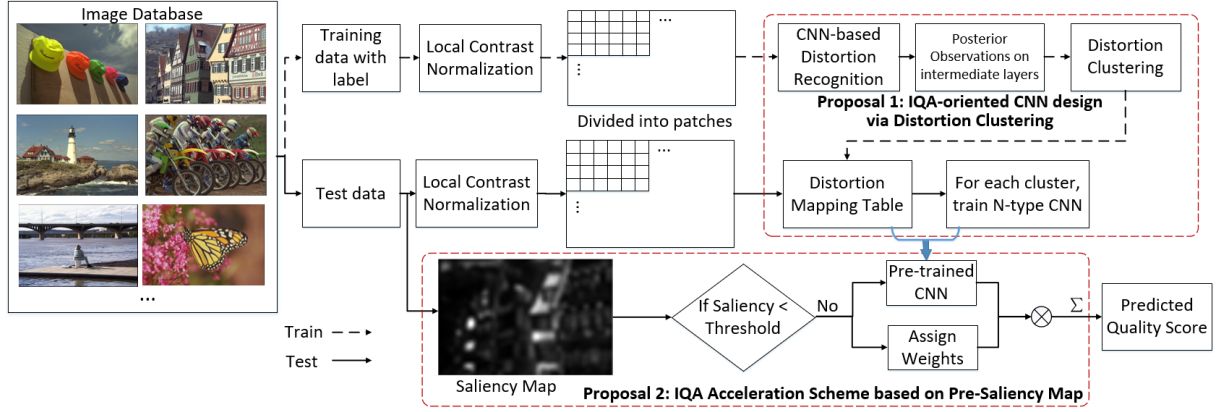


Fig. 2.10 Overview of the proposed fully-blind IQA framework for natural scene images.

2.4 Proposed Screen content image-targeted method

Except for natural scene images, screen content images (SCIs) is also significantly increasing in various multimedia applications and communication system over the Internet. Typically, the visual quality of SCIs has a great influence on the quality of experiences from the view point of end users. Thus, the image quality prediction designed for SCIs become quite important, because the quality of SCIs can not only be used to evaluate the performance of multimedia applications, but also be embedded into the optimization algorithms for various processing systems. An accurate and fast SCIs quality prediction is highly desired in both research and industry perspectives.

However, typically natural scene images targeted image quality assessment cannot achieve the same high accuracy for screen content images, due to complex characteristics and the special attention to textual regions for human visual systems. Therefore, I modify my proposals to make it also work for SCIs. The detailed explanation is discussed in the following.

2.4.1 Patch-level CNN Design

In this section, I present a CNN structure to predict the scores for patches, which is similar to natural scene images. By splitting the whole images into patches, small patches will have no obvious semantics differences between SCIs and general images. I split the raw image into 32×32 patches with RGB components as the input data of the CNN. Local mean subtraction and

divisive normalization [13] is applied before feeding the data into CNNs. The final CNN architecture is depicted as Fig.2.11, which is a $32 \times 32 \times 3 - 26 \times 26 \times 50 - 13 \times 13 \times 100 - 7 \times 7 \times 100 - 4 \times 4 \times 200 - 1024 - 1024 - 1$ structure. The quality score for each patch is the same as the score of raw image, which is used as the label together with the patch data to train the network.

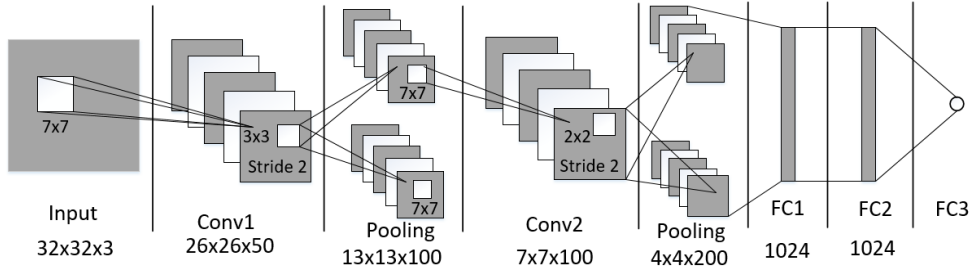


Fig. 2.11 The architecture of IQA-oriented CNN.

We adopt a Euclidean loss layer for the objective function:

$$J(\theta; x^{(i)}, y^{(i)}) = \frac{1}{2N} \sum_{i=1}^N \|f(\theta; x^{(i)}) - y^{(i)}\|^2 \quad (2.12)$$

where $(x^{(i)})$ represents the input images, and $y^{(i)}$ is the label data; θ is the parameter vector; $f(\theta; x^{(i)})$ denotes the CNN predicted score; N denotes the batch size, and I set it as 128 during the training process. All the implementation settings are the same as natural scene images. Empirically, we use 20 epoches to reach the convergence performance. The convergence curves are also given in Experimental results.

2.4.2 SCIs-oriented Quality Aggregation Acceleration

After obtaining the quality scores q_i of each patch, I need to calculate the whole quality scores Q of one SCI. Basically, the quality score of the general images can be calculated as the mean quality scores of patches as

$$Q = \frac{\sum_{i=1}^n q_i}{n} \quad (2.13)$$

where n is the total number of patches processed by CNN.

Using the mean value is intuitive and simple, but cannot bring the time reduction and performance enhancement for SCIs quality prediction. If I could select some representative patches and only conduct the CNN computation for selected patches, the running time can be greatly reduced. In previous studies [1], for general images, salient patches can be regarded as representative patches for general images. However, conventional saliency map did not work well

for SCIs due to several reasons. First, conventional saliency maps usually target at general images with natural scenes. For natural scenes, human visual system obtain the semantics information by comparing the differences of colors, intensity and orientations between backgrounds and foregrounds. Examples of conventional saliency maps for SCIs in SIQAD database are shown in the first four columns of Fig. 2.12. It is clearly observed that conventional methods [25] [26] [32] will give large salient value to large images parts instead of textual parts. Second, different from general images, SCIs are the mixture of natural images, computer graphics, texts, documents and other components. Especially, human eyes need to pay an attention to textual parts to read the information and obtain the semantics information, as explained in [28]. Sometimes picture information is even supplementary information to help viewers understand texts well.

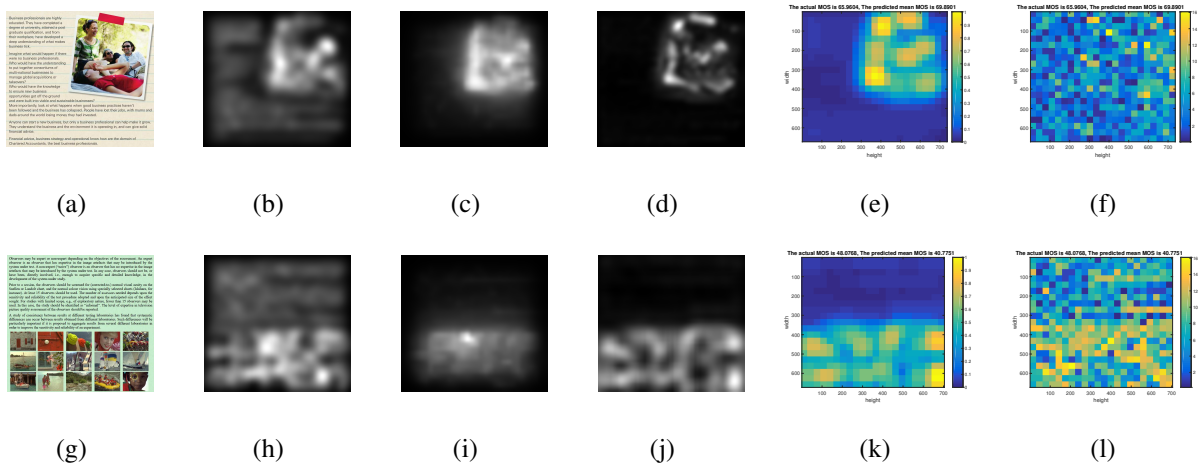


Fig. 2.12 Examples of images, corresponding saliency maps and prediction errors: (a)(g) Distorted images, (b)(h) saliency map using Itti [25], (c)(i) saliency map using Harel [26], (d)(j) fast saliency map [32], (e)(k) S value for each patch, (f)(l) the absolute value of prediction errors produced by CNNs for each patch.

Table 2.3 Time comparison between different saliency map methods with SIQAD images.

Method	Itti [25]	Harel [26]	Fast SM [32]
Time (s)	0.1416	0.4383	0.0178

Although conventional saliency map cannot be used to describe the focus of human visual system on SCIs, but I observe that fast saliency map (SM) calculation [32] is a good way to distinguish the textual regions from picture regions, as shown in (d)(j) of Fig. 2.12. That is because fast SM method is a simplified algorithm of the saliency model in [25] and only uses the color and intensity information to generate a single-scale feature map. The color and intensity

Algorithm 2 Quality Aggregation Acceleration Algorithm

Input: Tested Image $I(X, Y)$

Split I into $N \times N$ patches $\{P_1, P_2, \dots, P_M\}$

while $i \in [1, M]$ **do**

 Calculate the value S for each P_i as Eq. (6):

if $L \leq S \leq U$ **then**

$q_i = f_{CNN}(P_i)$;

else

 Skip the CNN computation for P_i ;

end if

$i = i + 1$;

end while

Calculate the final quality score Q using Eq. (5).

features can partition SCIs into texts and pictures segments well. Using fast saliency map has another benefit than the other two methods, that is, it will not bring too much complexity overhead as shown in Table 2.3. Since I do not need the pixel-level salient value, I will use the maximal value in one patch to reflect the characteristic of this region as

$$S = \max\{s(x, y)\}, x \in [1, N], y \in [1, N] \quad (2.14)$$

where $N \times N$ is the patch size. Examples of S value and predicted error for each patch are illustrated in the last two columns of Fig. 2.12. From (e)(k), I can observe that small S value implies that this regions are likely to be textual parts. We calculate the absolute value of the prediction error produced by the CNN for each patch as shown in (f)(l) of Fig. 2.12. It can be observed that large prediction errors are more likely to occur in the picture regions than textual regions.

We give a assumption that SCIs are uniformly distorted and no large regions need to be inpainting, which is satisfied in typical multimedia applications. Thus, I propose to select a group of patches with S in a given range $[L, U]$, where L, U are the lower bound and upper bound. After selecting representative patches, I calculate the mean quality score for the whole image. The quality aggregation algorithm is summarized in Algorithm 3. Based on the above analysis, the given range $[L, U]$ is likely to be small to select the textual regions. By experiments, L and U can be set as 0 and 0.1, respectively. The selection of range has a great impact on performance and complexity.

Then I discuss the complexity performance comparison with different ranges. Our platform is a PC with 4.20 GHz CPU, 16 GB RAM and GeForce GTX 1080 GPU. The experiments are

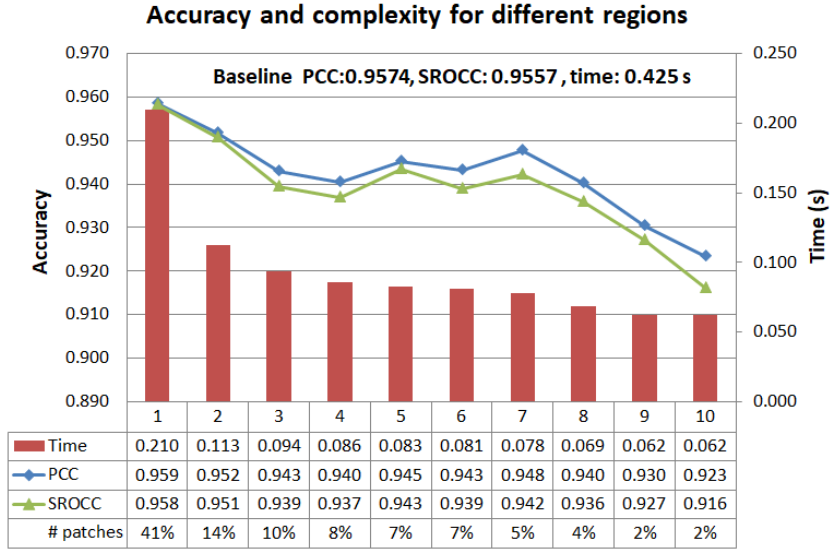


Fig. 2.13 Accuracy and complexity with different ranges.

implemented using MATLAB script in a MATLAB R2016b environment. The results about the range $[L, U]$ is discussed in Fig. 2.13. We divide the total range $[0, 1]$ into ten intervals equally. In the first interval, L is 0 and U is 0.1. In the second interval, L is 0.1 and U is 0.2 and so on. The baseline refers to the results of averaging all the patch scores, that is L is 0 and U is 1.

In terms of PCC and SROCC, using the patches with small S value brings higher accuracy, because small S implies that this region is likely to be textual regions. Especially, when (L, U) is equal to $(0, 0.1)$, the PCC (0.959) and SROCC (0.958) are even better than the baseline. In terms of the complexity, the running time of $[0, 0.1]$ is only 0.21 second, which is only half of the baseline (0.425 s) and only 41% of patches are processed in the range of $[0, 0.1]$. Using different L and U can achieve flexible accuracy and complexity performance.

2.5 Experimental Results

The experiments are conducted on three publicly available image databases.

- 1) The LIVE database [35] consist of 29 reference images and 779 distorted images. Five distortion types are provided: JPEG 2000 compression (JP2K), JPEG compression (JPEG), White Gaussian (WN), Gaussian blur (BLUR) and a Rayleigh fast fading channel simulation (FF). Differential mean opinion scores (DMOS) is provided for each image and higher value of DMOS (from 0 to 100) means low visual quality of the image.
- 2) The TID 2008 database [36] contains 25 reference images and 1700 distortion images with

17 different types of distortions and four levels of distortions. MOS for each distorted image is provided as subjective scores and higher value of MOS (from 0 to 9) corresponds to higher visual quality of the image.

- 3) The SIQAD database [37] contains 20 reference images with 7 distortion types including Gaussian noise (GN), Gaussian blurring (GB), motion blurring (MB), contrast change (CC), JPEG compression, JPEG2000 compression and layer-segmentation based compression (LSC), and 7 distortion levels for each distortion type. Thus, SIQAD has 980 distorted images and corresponding DMOS values with a range of [0, 100].

The CNN training is implemented in Caffe [38]. The performance is evaluated by how well the predicted image quality score correlates with subjective test results. Thus, Pearson's correlation coefficient (PCC) and the Spearman rank-order correlation coefficient (SROCC) are used to provide quantitative measures of the proposed IQA method.

Pearson's correlation coefficient is denoted as r_p , as defined as Eq.(5). Pearson's correlation coefficient, denoted as r_p , measures the prediction accuracy of the new metric with respect to subjective results. For a set of N data pairs (x_i, y_i) with means \bar{x} and \bar{y} , r_p is defined as

$$r_p = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \quad (2.15)$$

Spearman rank-order correlation, denoted as r_s , measures the prediction monotonicity which determines how well the estimated result reflects an increase or decrease in the actual subjective result regardless of the magnitude of increase or decrease. For a set of N data pairs (X_i, Y_i) with midranks \bar{X} and \bar{Y} , r_s is defined as

$$r_s = \frac{\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^N (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^N (Y_i - \bar{Y})^2}} \quad (2.16)$$

The value of PCC and SROCC ranges from 0 to 1, where high value of correlation coefficient indicates the predicted score match the actual score better. 0 means they have no correlation.

2.5.1 Effect of CNN Structure

In this section, I analyze the effect of the CNN structure on the prediction performance. This CNN structure is used for both natural scene images and screen content images.

To alleviate the effect of the pre-saliency map, I do not use any proposed distortion clustering or the pre-saliency map. The final image score is only calculated by the mean value of the patch

score. The effect of the kernel size and the number of kernels on accuracy needs to be investigated. The kernel size is important for extracting the features from patches. We apply kernel sizes of 3×3 , 5×5 , 7×7 , and 9×9 . The results are presented in Table 2.4. The CNN performance maintains a high accuracy, independent of the kernel size. Furthermore, the effect of the number of feature maps, varied from 50 to 200, is analyzed in Table 2.5. In line with the increase of the number of feature maps, the accuracy increases slightly. To achieve a low complexity, I choose 50 feature maps in the first convolution layer.

Table 2.4 The effect of the kernel size in the convolution layer.

Kernel Size	PCC	SROCC	RMSE
3×3	0.9509	0.9484	8.7770
5×5	0.9498	0.9453	8.8795
7×7	0.9527	0.9494	8.6262
9×9	0.9486	0.9459	8.9846

Table 2.5 The effect of the number of features in the convolution layer.

Features	PCC	SROCC	RMSE
50	0.9527	0.9494	8.6262
100	0.9574	0.9530	8.2819
200	0.9579	0.9525	8.3142

The convergence curve of CNN is shown in Fig.2.14. The number of iterations is set to 150000 during the training process. The total number of training patches is $\frac{758 \times 512}{32 \times 32} \times 982 \times 80\% = 301670$. Because the batch size is 128, about 2356 ($\frac{301670}{128} = 2356$) iterations are needed to finish one epoch. According to Fig.2.14, the training loss converges quickly, and around 20 epochs is sufficient to achieve good convergence results. As long as the number of training epochs is sufficient, the final prediction results are accurate.

2.5.2 Performance Evaluation on NSIs-targeted Approach

For natural scene images, my proposal includes two strategies, pre-saliency map and distortion clustering. This is a Fully-blind and Fast Image Quality Predictor, so I denote it as **FFIQP** in the following. To evaluate the accuracy performance, I compare the results on the TID2008 and LIVE databases, respectively. The threshold ϵ of saliency map is set as 0.25 for LIVE database and is set

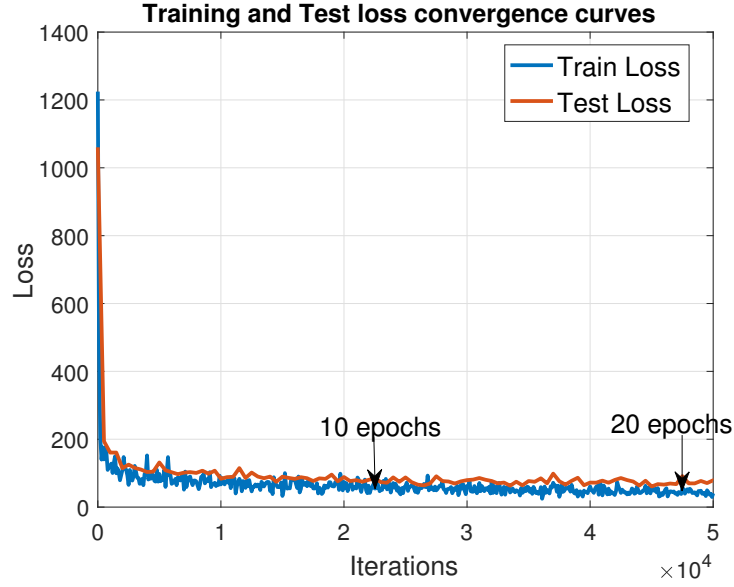


Fig. 2.14 The convergence curves of the training loss and test loss.

as 0 for TID2008 databases. Because the ranges of different IQA metrics are different, a non-linear regression with a five-parameter logistic function [35] is applied. RMSE is omitted on account of the different ranges of values in different IQA methods.

2.5.2.1 Accuracy on TID2008 dataset

On the TID2008 database, I use four groups of distortion types, as listed in Table 2.2, to train four CNNs. We train each CNN by randomly choosing 80% images from TID2008 database [36] for training and use the other 20% images for testing, as other works. To prevent overfitting I adopt the horizontal flipping strategy for data augmentation, because the horizontal flipping will not affect the subjective quality perception for humans. The PCC and SROCC results are listed in Table 2.6 and Table 2.7, respectively.

Table 2.6 Comparison results of PCC on the TID2008 database.

Works	WN	WNC	SCN	MN	HFN	IN	QN	BLUR	IDN	JPEG	JP2K	JPEGTE	JP2KTE	ALL
PSNR	0.934	0.911	0.952	0.870	0.971	0.906	0.891	0.874	0.945	0.900	0.877	0.769	0.794	0.651
SSIM [4]	0.756	0.778	0.801	0.822	0.846	0.704	0.801	0.851	0.963	0.946	0.969	0.891	0.833	0.851
VSI [7]	0.923	0.904	0.939	0.790	0.959	0.820	0.867	0.946	0.972	0.986	0.986	0.924	0.882	0.915
CORNIA [14]	0.911	0.932	0.852	0.886	0.933	0.922	0.892	0.932	0.908	0.963	0.929	0.732	0.762	0.837
Sun [19]	0.928	0.867	0.945	0.939	0.966	0.942	0.861	0.963	0.964	0.965	0.972	0.829	0.814	0.919
FFIQP	0.922	0.949	0.963	0.924	0.981	0.888	0.950	0.877	0.927	0.909	0.990	0.929	0.948	0.935

The results of PSNR, SSIM and VSI are obtained from our experiments by using the

Table 2.7 Comparison results of SROCC on the TID2008 database.

Works	WN	WNC	SCN	MN	HFN	IN	QN	BLUR	IDN	JPEG	JP2K	JPEGTE	JP2KTE	ALL
PSNR	0.911	0.907	0.923	0.849	0.932	0.918	0.870	0.867	0.938	0.901	0.830	0.766	0.777	0.667
SSIM [4]	0.795	0.780	0.793	0.788	0.872	0.712	0.793	0.955	0.955	0.916	0.968	0.884	0.850	0.876
VSI [7]	0.923	0.912	0.930	0.773	0.925	0.830	0.871	0.953	0.970	0.962	0.985	0.916	0.894	0.911
CORNIA [14]	0.913	0.928	0.868	0.879	0.921	0.924	0.886	0.932	0.887	0.929	0.919	0.726	0.785	0.813
Sun [19]	0.910	0.848	0.932	0.920	0.940	0.934	0.860	0.956	0.932	0.931	0.952	0.809	0.835	0.926
BIECON [23]	0.913	0.835	0.903	0.835	0.931	0.913	0.893	0.953	0.917	0.943	0.956	0.862	0.827	0.923
LLM [24]	0.946	0.898	0.935	0.755	0.952	0.833	0.906	0.945	0.948	0.954	0.970	0.846	0.928	0.924
FFIQP	0.921	0.921	0.932	0.901	0.927	0.882	0.925	0.840	0.926	0.851	0.976	0.923	0.937	0.931

definition and their public code from [4] [7], because they did not release their PCC or SROCC results on TID2008 database. The other results are from their papers. For BIECON [23] and LLM [24], only SROCC is provided in their papers. We can observe that PSNR works effectively for noise distortion, such as WN, SCN, and HFN, but does not achieve good results for transmission distortion. SSIM [4] detects the structure similarity to work effectively for JPEG and JPEG2000, but does not perform well for impulsive noises. VSI [7] utilizes the visual saliency information to measure the distortion, but cannot work for marked noise (MN). CORNIA [14] is proposed based on unsupervised feature learning, and the accuracy is dependent on the codebook construction. Sun [19], BIECON [23], and LLM [24] take advantage of the deep learning method to further improve the prediction accuracy, but still cannot achieve a high correlation for JPEGTE and JP2KTE. FFIQP achieves competitive performance compared with other methods on TID2008 database.

2.5.2.2 Accuracy on LIVE dataset

Only five distortion types are contained on the LIVE database, and so the distortion clustering is not conducted. We randomly choose 80% of images for training and the remaining 20% for testing. The PCC and SROCC are listed in Table 2.8. It is significant that our method achieves a high accuracy in particular for the JPEG, JP2K, and WN. The comparison results are shown in Table 2.13. FFIQP achieves higher correlation than state-of-the-art IQA metrics. Some corresponding scatter plots between actual DMOS and predicted quality scores for tested images are shown in Fig. 2.15. It can be observed that FFIQP achieves higher linearity with subjective image quality scores and has fewer outlier points than related works. These effectiveness on LIVE dataset has been validated in the paper [1].

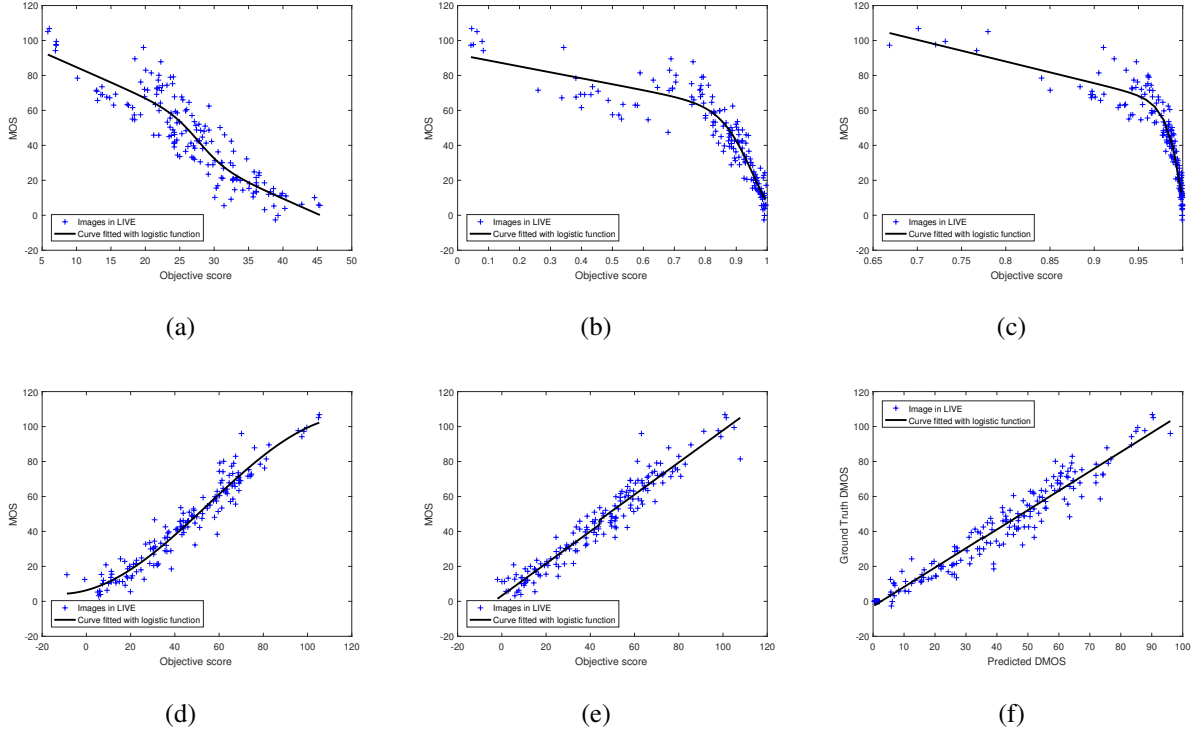


Fig. 2.15 Scatter plots of actual DMOS with (a) PSNR, (b) SSIM [4], (c) VSI [7], (d) BRISQUE [13], (e) CORNIA [14], (f) FFIQP on LIVE database.

Table 2.8 Accuracy performance on the LIVE database.

Distortion	JP2K	JPEG	WN	BLUR	FF	ALL
PCC	0.979	0.986	0.986	0.917	0.914	0.978
SROCC	0.969	0.974	0.978	0.956	0.883	0.974
RMSE	5.730	4.465	4.560	6.666	8.646	5.963

2.5.2.3 Complexity Performance Analysis

In this section, the complexity performance comparison with different IQA metrics is discussed. Our platform is a PC with 4.20GHz CPU and GeForce GTX 1080 GPU. Complexity experiments are implemented using MATLAB script in a MATLAB R2016b environment. The experiments are conducted with LIVE database and I randomly select 80% of images for training and the remaining 20% for testing, which keeps the same as Section 4.2.

We compare the performances of non-SM, post-SM, and pre-SM in Table 2.10. Non-SM refers to the case omitting the calculation of the saliency map, but taking the mean values of patch scores as the image scores. Post-SM means only assigning weights by different saliency models after the

Table 2.9 Comparison results on the LIVE database.

Works	PCC	SROCC
PSNR	0.868	0.873
SSIM [4]	0.913	0.906
FSIM [6]	0.961	0.965
VSI [7]	0.948	0.952
DIIVINE [11]	0.916	0.917
BLIINDS-II [12]	0.930	0.931
BRISQUE [13]	0.940	0.942
CORNIA [14]	0.942	0.935
Kang [16]	0.953	0.956
Li [17]	0.956	0.935
VeNICE [18]	0.960	0.950
Sun [19]	0.958	0.959
Bosse [20]	0.972	0.960
Pan [21]	0.969	0.968
FFIQP	0.978	0.974

Table 2.10 Complexity reduction with the pre-saliency map.

Method	PCC	SROCC	Time(s)	TR(%)
non-SM	0.953	0.949	0.404	0
post-SM by [25]	0.977	0.975	0.530	+31.2
post-SM by [26]	0.979	0.977	0.658	+62.9
post-SM by [32]	0.977	0.974	0.432	+6.9
FFIQA ($\varepsilon = 0.25$)	0.978	0.974	0.191	-52.7

CNN computation. Post-SM introduces a complexity overhead (6.9%-62.9%), but [32] achieves a comparable accuracy with [25] [26] with the lowest complexity. FFIQA utilizes the pre-saliency map before CNN to accelerate the IQA computation. The results validate that FFIQA improves the prediction accuracy and achieves a 52.7% time reduction compared to the case without the saliency map.

The time comparison for different IQAs is presented in Table 2.15. We can achieve a flexible complexity performance by using different thresholds according to the requirements. FFIQP with

ε set to 0.25 achieves a lower complexity than DIIVINE [11], BLIIND-II [12], and CORNIA [14], while obtaining a higher accuracy. FFIQP with ε at 0.70 requires less time consumption than SSIM [4], VSI [7], and BRISQUE [13], with a higher accuracy.

Table 2.11 Complexity comparison results (in ascending order).

Works	Time(s)
FFIQP ($\varepsilon = 0.70$)	0.042
SSIM [4]	0.051
VSI [7]	0.106
BRISQUE [13]	0.144
FFIQP ($\varepsilon = 0.25$)	0.191
CORNIA [14]	2.973
DIIVINE [11]	9.610
BLIIND-II [12]	26.90

2.5.2.4 Cross Database Validation

Cross-database validation is performed to demonstrate that our FFIQP method is independent of the database. We train the CNN using the LIVE database and test it on the TID2008 database, because the image contents in the two databases are different. 4 common types of distortions, i.e. JP2K, JPEG, WN, and BLUR, which are owned by both LIVE and TID2008, are used to conduct experiments for fair comparison, similar to related works. The comparison results are shown in Table 2.12. FFIQP achieves a competitive performance with other IQA metrics. The corresponding scatter plots are shown in Fig 2.16. The plot for Pan [21] is not included because they did not release their source code and the results in Table 2.12 are from their paper. It can be observed that FFIQP can predict the quality more accurately than other works for cross-database testing.

2.5.3 Performance Evaluation on SCIs-targeted Approach

The experiments are only conducted with the publicly available image databases SIQAD [37]. I perform experiments on SIQAD database by randomly choosing 70% data for training, 10% for validation and 20% for testing. Training is implemented using Caffe [38]. The performance is evaluated by how well the predicted image quality score correlates with subjective scores, thus, Pearson's correlation coefficient (PCC) and Spearman rank-order correlation coefficient (SROCC)

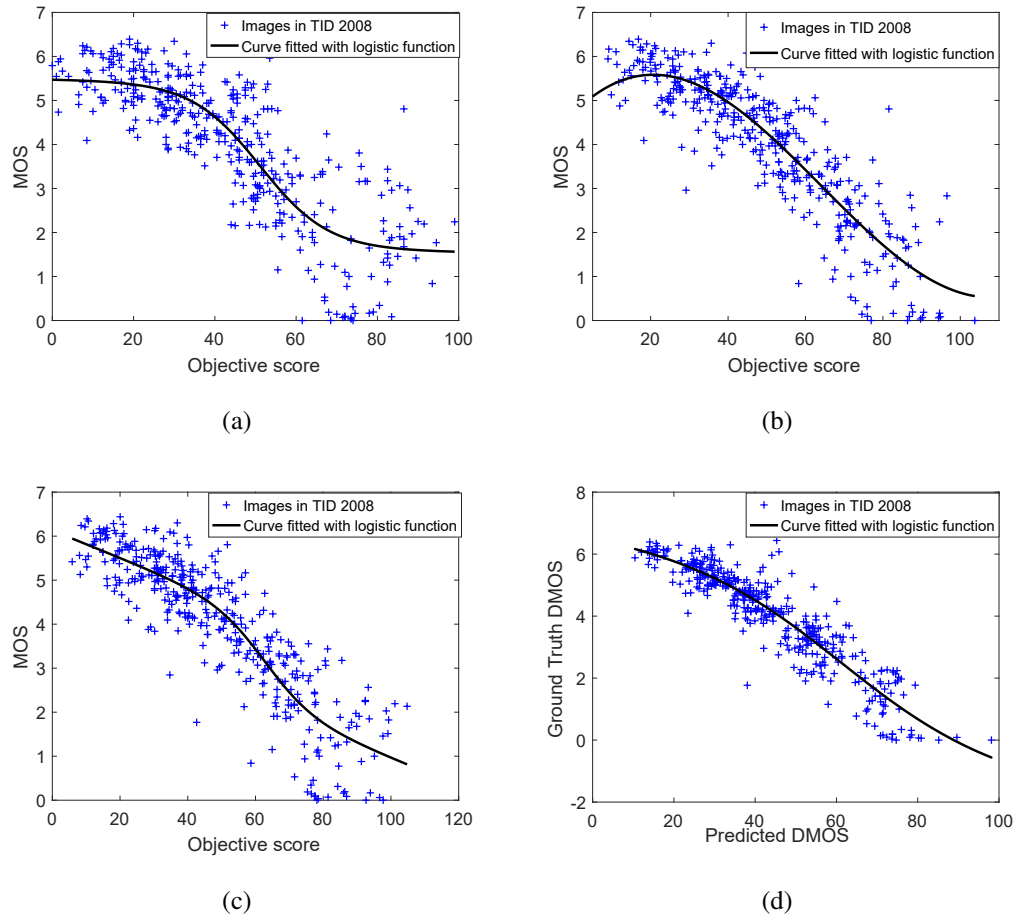


Fig. 2.16 Scatter plots of actual DMOS with (a) BLIINDS-II [12], (b) BRISQUE [13], (c) CORNIA [14], (d) FFIQA for cross-database validation.

Table 2.12 Comparison results trained on LIVE and tested on TID2008.

Works	PCC	SROCC
BLIINDS-II [12]	0.844	0.854
BRISQUE [13]	0.882	0.896
CORNIA [14]	0.890	0.880
Pan [21]	0.922	0.916
FFIQP	0.925	0.915

are used to give quantitative measures about the correlation with ground truth subjective quality scores.

2.5.3.1 Accuracy Evaluation

The comparison result on SIQAD dataset is shown as Table 2.13. In terms of category, “General images-oriented methods” are corresponding to the methods aiming at general images; “SCIs-oriented methods” denote the quality prediction methods aiming at SCIs; “Both” refers to the methods that can work with both general images and SCIs. The results of PSNR, SSIM, FSIM and VSI are obtained from our experiments by using their public code from [4] [6] [7]. The other results are from their original results in their papers. We calculate the mean value of all the patches as the whole image score, i.e. $[L, U] = [0, 1]$. For fairness, I also use the average patch score results in [22]. According to Table 2.13, our proposed method achieve the highest PCC and SROCC among existing SCIs-oriented methods.

Table 2.13 Comparison results on the SIQAD database.

Category	Works	PCC	SROCC
General images-oriented methods	PSNR	0.5869	0.5604
	SSIM, TIP [4]	0.7154	0.7112
	FSIM, TIP [6]	0.5906	0.5824
	VSI, TIP [7]	0.7514	0.7549
	Min, ICME [15]	0.7625	0.7689
Both	Zhou, TIP [31]	0.8280	0.7880
SCIs-oriented methods	Gu, TMM [28]	0.8872	0.8803
	Ni, ICIP [29]	0.8648	0.8504
	Fang, ICME [30]	0.8366	0.8063
	Zuo, ICIP [22]	0.9531	0.9508
	Yang, TIP [37]	0.8584	0.8416
	Proposal	0.9574	0.9557

Beside, the comparison results for each distortion type is summarized in Table 2.14. Seven distortion types are contained in SIQAD database, but as indicated in [14], contrast change (CC) is a highly subjective and non-monotonic task related with reference images for human, so it is hard for no-reference training and I exclude the distorted images with CC. We highlight the top results in bold. Among these methods, only Zuo’s work [22] used the CNNs and other works use the conventional image processing methods. The difference between Zuo’s work and our CNN architecture is that Zuo used the multi-channel CNN framework and I did the experiments to

Table 2.14 Comparison results for each distortion type on the SIQAD database.

Criterion	Works	GN	GB	MB	JPEG	JP2K	LSC	ALL
PCC	SSIM, TIP [4]	0.8006	0.8535	0.7501	0.6005	0.7252	0.6427	0.7154
	FSIM, TIP [6]	0.8850	0.8210	0.7289	0.5955	0.6544	0.7042	0.5906
	VSI, TIP [7]	0.8836	0.8504	0.6620	0.6526	0.7001	0.6717	0.7514
	Ni, ICIP [29]	0.8889	0.9156	0.8753	0.7904	0.7850	0.7747	0.8648
	Zuo, ICIP [22]	0.9796	0.9727	0.9393	0.9392	0.9343	0.8085	0.9531
	Proposal	0.9591	0.9577	0.9503	0.9329	0.9417	0.9120	0.9574
SROCC	SSIM, TIP [4]	0.8483	0.8455	0.7489	0.6090	0.7118	0.6288	0.7112
	FSIM, TIP [6]	0.8705	0.8221	0.7241	0.6640	0.6860	0.7058	0.5824
	VSI, TIP [7]	0.8655	0.8495	0.7658	0.7193	0.7299	0.7419	0.7549
	Ni, ICIP [29]	0.8745	0.9154	0.8788	0.7871	0.7762	0.7798	0.8504
	Zuo, ICIP [22]	0.9795	0.9700	0.9381	0.9298	0.9409	0.8155	0.9508
	Proposal	0.9391	0.9501	0.9487	0.8837	0.9490	0.8810	0.9557

analyze the best kernel size and the best number of features in the convolution layers as Section 2.1. It is worth noting that I can also combine our CNN with the weight assignment of Zuo's work to further improve the accuracy. It can be observed that the CNN based SCIs quality prediction outperforms the conventional SCIs quality prediction methods significantly. Especially, our method achieves the best accuracy performance for blurring and compression distortions (MB, JP2K and LSC). Some scatter plots between ground truth DMOS and predicted scores are shown in the Fig. 2.17. It can be observed that our method achieves higher linearity with subjective scores than conventional methods.

2.5.3.2 Complexity Evaluation

In this section, the complexity performance comparison with different quality prediction methods is discussed. Our platform is a PC with 4.20 GHz CPU, 16 GB RAM and GeForce GTX 1080 GPU. The experiments are implemented using MATLAB script in a MATLAB R2016b environment.

The running time comparisons for different methods are presented in Table 2.15. The time for SSIM [4], FSIM [6] and VSI [7] are reimplemented with our PC using their public source code. It can be observed that our method achieves moderate and flexible complexity performance with higher accuracy performance among related methods.

Table 2.15 Complexity comparison results.

Works	PCC	Running Time (s)
SSIM, TIP [4]	0.7154	0.0432
FSIM, TIP [6]	0.5904	0.1802
VSI, TIP [7]	0.7514	0.1124
Proposal	0.923 - 0.959	0.062 - 0.210

2.6 Chapter Summary

This chapter mainly discuss the proposed image quality assessment based on convolutional neural networks.

First, to address the high complexity issue of convolutional neural networks, I propose a pre-saliency map based blind IQA method via CNN. Firstly, I design a CNN to predict the quality score of each patch. Then I analyze the relationship between image saliency information and CNN prediction error. Secondly, I present a pre-saliency map based quality aggregation algorithm. This algorithm sets a threshold to adaptively apply CNN computation and assigns weights for patches. Experimental results demonstrate that our proposed method can achieve high accuracy (PCC = 0.978) with subjective quality scores on LIVE dataset, which outperforms existing IQA methods. Besides, our method achieves 52.7% time reduction compared to the IQA without saliency map.

Second, a distortion clustering strategy is proposed by utilizing the intermediate-layer results in the CNN to generate the distortion group. Each CNN associated with each group is trained to accurately predict the local quality scores for patches. By using these strategies, the algorithm can achieve better accuracy (PCC = 0.935) on TID2008 dataset than existing methods. Besides, our approach achieves competitive performance on many different types of distortion artifacts.

Third, I have proposed a fast SCIs-oriented image quality prediction method using CNNs. First, a well-designed CNN architecture is present to predict the quality scores of small patches. Second, I propose to select representative patches to accelerate the quality score aggregation for the whole SCIs. Experimental results demonstrate that the proposed method can achieve a high accuracy (PCC = 0.9574) in terms of subjective quality scores, outperforming existing methods. Furthermore, by selecting different groups of patches, our method processes one image from SIQAD database in 0.062-0.210 second to achieve the flexible and low complexity.

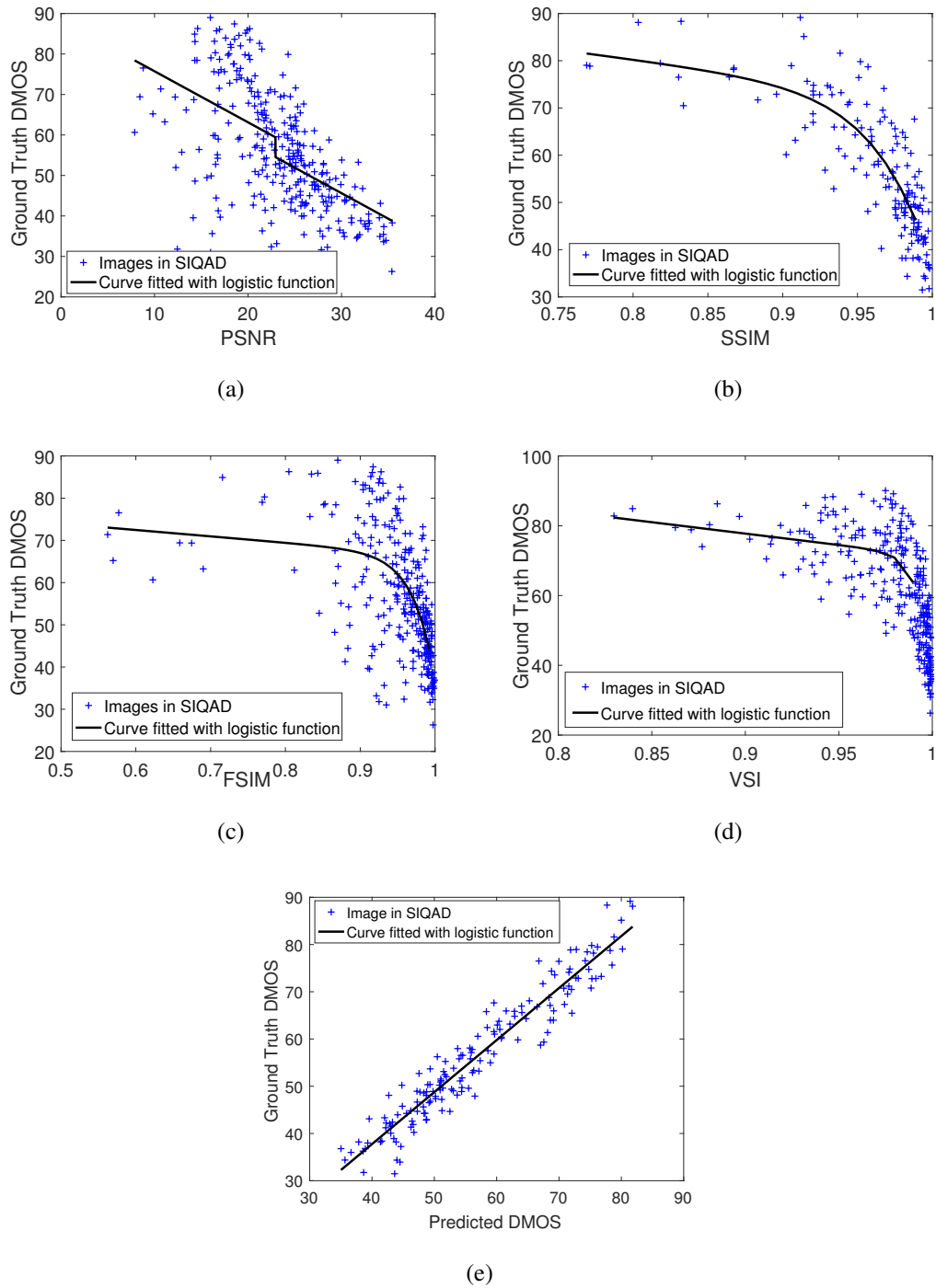


Fig. 2.17 Scatter plots of actual DMOS with (a) PSNR, (b) SSIM [4], (c) FSIM [6] (d) VSI [7] and (e) proposed method on the SIQAD database.

Chapter 3

Learning-based Image Compression through Convolutional Autoencoder

In this chapter, I propose a lossy image compression solution using convolutional autoencoder. Firstly, I compare the performance of CAE, GAN, and super resolution for image compression, and propose deep residual learning architecture. The results are submitted to CVPR workshop CLIC 2018 and 2019. Secondly, I design a novel CAE structure with multiple downsampling and upsampling units to generate feature maps. I optimize this CAE using an approximated rate-distortion loss function. To generate a more energy-compact representation, a principal components analysis (PCA)-based rotation is applied to generate more zeros in the feature maps. Thirdly, a mathematical analysis on the energy compaction property based on CAE is provided to define a normalized coding gain, which is a measure of compression capability. Based on the above analysis, a regularizer is given to be incorporated into a loss function to train the CAE models to achieve a higher coding gain. Experimental results demonstrate that our method outperforms the image compression standard HEVC-intra in terms of MS-SSIM. At last, a thorough and rigorous perceptual quality study on different compression algorithms is conducted based on my proposed approach.*

3.1 Related Work

Image compression has been a fundamental and significant research topic in the field of image processing for several decades. Traditional image compression algorithms, such as JPEG [44], JPEG2000 [45] and BPG (Better Portable Graphics) [46], which is the HEVC [47] with intra

*This chapter is adapted from the work published in [39], [40], [41], [42], [43].

profile, are module-based encoder and decoder (codec) pipeline designs. They use fixed transforms, quantization and the entropy coder to reduce spatial redundancy for images. However, they are not optimal image compression solutions for all types of image contents and formats. Deep learning approaches has the potential to enhance compression performance. Recently, various approaches have been studied to improve coding performance of learned image compression.

A brief review of recent approaches is provided in two categories. The first category is hand-crafted conventional image compression algorithms. The second one is end-to-end deep learning based compression, which has emerged as a research topic in recent years.

3.1.1 Hand-crafted Image Compression

Existing image compression standards rely on hand-crafted module design separately to achieve image compression. The modules include features such as color space conversion, intra prediction, transform, quantization, and entropy coder. Specifically, if the image consists of RGB data, the encoder will first convert it into YUV space as a preprocessing step. Subsequently, JPEG [44] applies 8×8 DCT, uniform quantization, and a variant of Huffman encoding to raw images to obtain the bitstream. JPEG2000 [45] uses multi-scale DWT, uniform quantization with deadzone and an adaptive arithmetic coder. JPEG2000 also introduces embedded block coding with optimal truncation, which is called the post rate-distortion optimization. Moreover, HEVC [47] utilizes flexible sizes (up to 32×32) for DCT, reconstruction quantization control and a content adaptive binary arithmetic coder (CABAC). Furthermore, HEVC introduces intra prediction with five types of sizes and 35 modes to remove the spatial correlation. However, there is still room to improve the performance of image compression based on HEVC. For example, the saliency information was incorporated into image compression optimization for HEVC-MSP in [48], and new color image compression implementations were proposed in [49] based on the YUV420 chroma format.

During the development of next-generation compression algorithms, some hybrid methods have been proposed to improve the performance of individual modules, by taking advantage of both conventional compression algorithms and latest machine learning approaches. For example, in [42] the downsampling of the input images was applied before BPG and used super-resolution neural networks as a post filter. In [50], a hybrid image coder based on a CNN-optimized in-loop filter and mode coding is proposed. In [51] two CNNs were proposed as the pre- and post-processing steps, along with existing compression algorithms. However, these are still not end-to-end approaches, resulting in performance limitations.

3.1.2 Learning-based Image Compression

Recently, machine learning and deep learning have been successfully utilized to different kinds of tasks and can potentially enhance the performance of image compression, owing to several reasons. First, the encoder-decoder pipeline in conventional compression standards resembles an autoencoder for learning compressed representation. In earlier years, the autoencoder was widely applied to dimensionality reduction, the compact representations of images, and generative learning models [52]. Therefore, it can achieve better compression performance. Most recent learning based compression approaches, including convolutional neural networks [53]- [60], recurrent neural networks [62]- [64] and generative adversarial networks [66]- [67], have all adopted the autoencoder architecture. The second advantage of learning based compression is its end-to-end learning architecture. Instead of designing the algorithms of each module like conventional codecs, the learning approaches have the potential to adapt quicker to fast development of new image formats, because all the parameters can be learned automatically regardless of image datasets. Therefore, deep learning based image compression is expected to become more generalized and more efficient. To summarize current developments, I will discuss all these related works in the following.

CAEs have also attracted great interest from researchers. Theis et al. [53] and Ballé et al. [54] first present end-to-end architecture of convolutional autoencoders for learned image compression. They incorporated differentiable approximations of round-based quantization and fully factorized the entropy model for end-to-end training with gradient backpropagation using an autoencoder. Moreover, to model an adaptive context model, Ballé et al. further proposed a hyperprior entropy model by adding a hierarchical autoencoder to the variational autoencoders reported in [55]. Ballé et al. [56] also reported a nonlinear transform for learned compression. Moreover, [57] proposed a joint autoregressive and hyperprior approach, denoted as Joint. The only difference is to append a masked 5×5 convolution after quantization and to concatenate the output of auxiliary autoencoder and masked convolution together to learn the entropy model. In [58] the round-based quantization was generalized to a soft vector quantization, and a soft histogram estimation method was proposed to estimate the rate. Following this approach, [59] further appended an auxiliary network to model the conditional probability of the latent representation to achieve promising compression performance. To realize the energy compaction, [39] proposed the use of principle component analysis to de-correlate different channels with regard to latent representation. To describe the region of interest for a human vision system, [60] considered an importance map to realize content-aware bit rate allocation.

RNN architectures can be used to predict the residual information between the raw image

and the reconstructed images in several iterations. Toderici et al. [62] first used a long short-term memory (LSTM) recurrent network to compress small thumbnail images (32×32), and also used a binarization layer to realize the quantization. This approach was further extended in [63] to compress full-resolution images. Priming and spatially adaptive bit rate were further considered in [64] to achieve higher compression performance. An RNN can be considered as a scalable coding system to generate multiple reconstructed images with different quality levels. However, it entails a complex composition.

GANs were used for image compression in [66]- [67] for high subjective reconstruction quality. Ripple et al. [66] first proposed to use multiscale adversarial training to obtain the reconstructed images. Santurkar et al. [65] used the DCGAN architecture to produce visually pleasing reconstructed images and videos. Agustsson et al. [67] proposed to fully synthesize the unimportant regions in the decoded image from a semantic label map to achieve extremely low bit rates. Typically, GANs are able to learn the distribution of images with a few information. The promising subjective quality results at low bit rate of these GAN-based approaches have validated the effectiveness. However, they are not designed to reach the transparent quality level, while high quality reconstruction is also important for image compression task. On the contrary, autoencoders are capable to compress data at all the bit rates, according to the abovementioned studies.

Based on the review of previous works, therefore, in this thesis I mainly discuss a convolutional autoencoder for image compression. Section 3.2 discuss different architectures. I not only compare the performance of CAE, GAN, and super resolution for image compression, but also propose deep residual learning architecture. The results are submitted to CVPRW 2018 and 2019. Section 3.3 propose a lossy image compression via CAE and Principle Component Analysis (PCA) from the viewpoint of spatial redundancy reduction. Based on this work, section 3.4 consider to add a regularizer of energy compaction term by formulating the coding gain of neural networks, which further reduce the spatial redundancy. Section 3.5 discusses the performance. Section 3.6 present a thorough and rigorous perceptual quality study through subjective tests. Section 3.7 gives a summary for this chapter.

3.2 Architecture Discussion

3.2.1 Performance Comparison of CAEs, GANs and SR-based approaches

In this subsection, I develop three overall compression architectures based on convolutional autoencoders (CAEs), generative adversarial networks (GANs) as well as super-resolution (SR), and present a comprehensive performance comparison. According to experimental results, CAEs achieve better coding efficiency than JPEG by extracting compact features. GANs show potential advantages on large compression ratio and high subjective quality reconstruction. Super-resolution achieves the best rate-distortion (RD) performance among them, which is comparable to BPG.

3.2.1.1 Convolutional Autoencoders for Compression

Generally, an autoencoder can be regarded as an encoder function, $y = f_{\theta}(x)$, and a decoder function, $\hat{x} = g_{\phi}(y)$, where x , \hat{x} , and y are original images, reconstructed images, and compressed data, respectively. θ and ϕ are optimized parameters in the encoder and the decoder function.

We propose a CAE network to replace conventional transforms, such as DCT and wavelet transform. The overall architecture is shown in Figure 3.1. Consecutive downsampling operations destroy the quality of reconstructed images. Therefore, I use a pair of convolution/deconvolution filters for one upsampling or downsampling operation. The CAE network structure is shown in Figure 3.2. As for the activation function after each convolutional layer, I utilize the Parametric Rectified Linear Unit (PReLU) function, instead of ReLU, which is commonly used in related works, because I find that PReLU can improve the quality of reconstructed images compared to ReLU, especially with high bit rate. Inspired by the RD cost function in traditional codecs, the loss function is defined as

$$\begin{aligned} J(\theta, \phi; x) &= \|x - \hat{x}\|^2 + \lambda \cdot \|y\|^2 \\ &= \|x - g_{\phi}(f_{\theta}(x) + \mu)\|^2 + \lambda \cdot \|f_{\theta}(x)\|^2 \end{aligned} \quad (3.1)$$

where $\|x - \hat{x}\|^2$ denotes the mean square error (MSE) distortion between original images x and reconstructed images \hat{x} . μ denotes uniform noises. λ is a parameter to measure the rate-distortion tradeoff. $\|f_{\theta}(x)\|^2$ denotes the amplitude of compressed data y , which reflects the number of bits used to encode compressed data. We use a subset of ImageNet database [76] consisting of 5500 images to train the CAE network. We used the Adam optimizer [70] and a batch size of 16 to train the model up to 8×10^5 iterations. The learning rate was kept at a fixed value of 0.0001, and

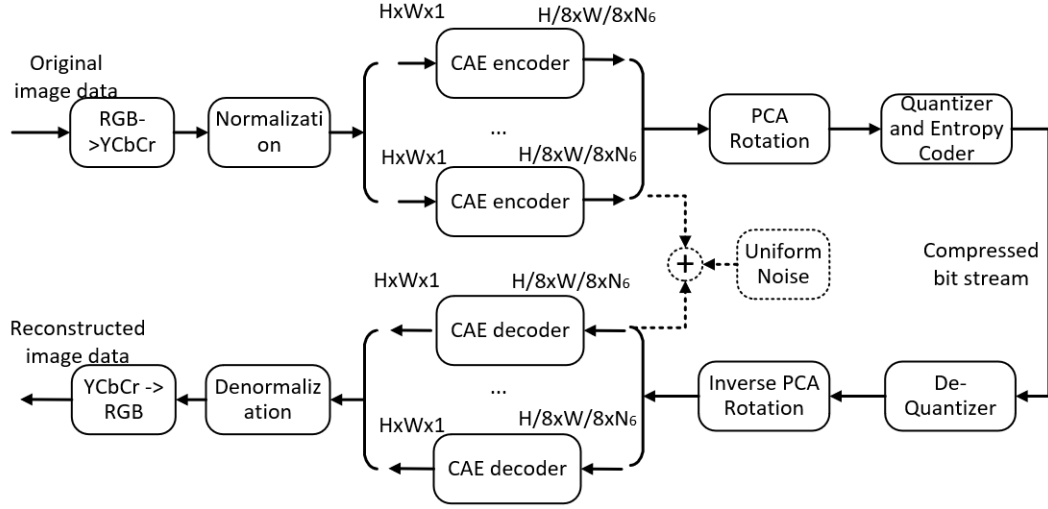


Fig. 3.1 Block diagram of CAE based image compression.

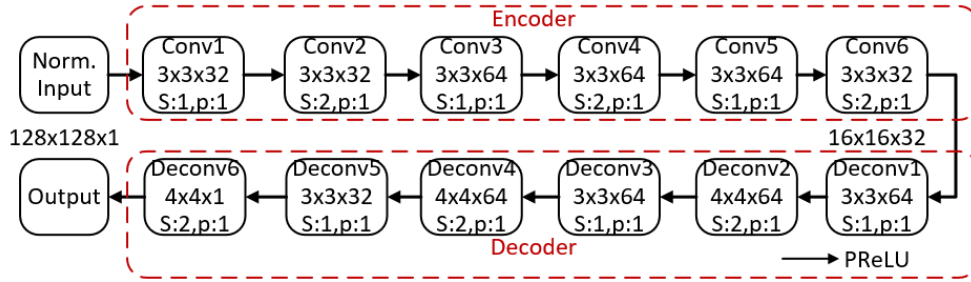


Fig. 3.2 The CAE network structure.

the momentum β_1 was set as 0.9. Then I apply the principle component analysis (PCA), uniform quantization and the JPEG2000 entropy coder to generate a bit stream.

The feature maps generated by CAEs are not energy-compact, thus, I further decorrelate feature maps using the PCA. The detailed discussions refer to the paper [39], which will be described in Section 3.3.

3.2.1.2 Generative Adversarial Networks for Compression

For the GAN based image compression, I add one convolutional layer to make the input size as 128×128 , based on the architecture of DCGAN [65] [84]. The activation function is kept the same as DCGAN. Because DCGAN only includes the generator as the decoder function, I add an encoder function, which has the same structure as the discriminator. To implement the end-to-end training, the loss function of the generator is defined as

$$J_G(x) = \|x - \hat{x}\|^2 + \beta \sum_{i \in [0,4]} \|D_{hi}(x) - D_{hi}(\hat{x})\|^2 \quad (3.2)$$

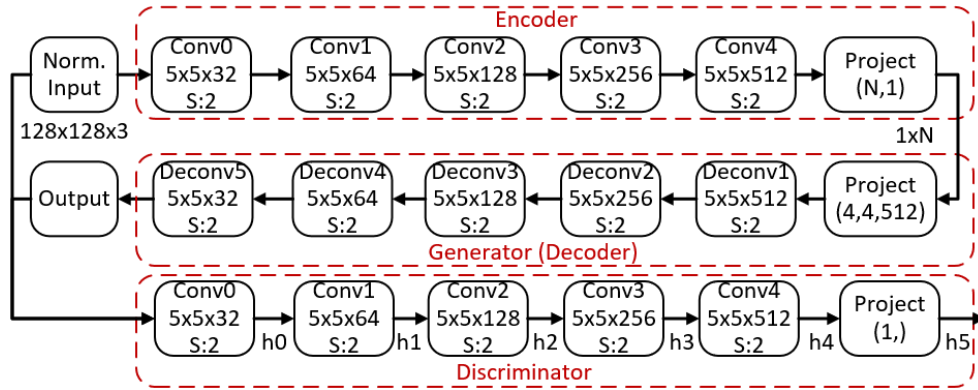


Fig. 3.3 The GAN structure.

where $\|x - \hat{x}\|^2$ denotes the mean square error between x and \hat{x} . Adding the discriminator network benefits the high quality reconstruction [66], so I add the second distortion term in Eq.(2). $D_{hi}(x)$ and $D_{hi}(\hat{x})$ are the outputs of the i -th convolutional layer in discriminator network for inputs x and \hat{x} , respectively. β is set as 0.01 in our experiments. The loss function of the discriminator is kept the same as DCGAN.

We use the training set of CLIC. The Adam optimizer [70] with a batch size of 128 was used for training. The learning rate was kept at a fixed value of 0.0001. The model is trained up to 25 epoches. The GAN structure is shown in Figure 3.3. The GAN based architecture has three differences from the CAE based architecture. First, the input has RGB components, so color space conversion from RGB to YCbCr is not applied. Second, I do not add uniform noises during the training process since GAN inherently reconstructs images from noises. Third, I use the range coder [87], instead of the JPEG2000 entropy coder.

I conduct some experiments on the CLIC validation dataset to discuss the performance. First, the effect of input sizes for one image is listed in Table 3.1, where $64 \times 64 \times 3 \rightarrow 1024$ denotes that the input size is $64 \times 64 \times 3$ and the code size N is 1024. It is observed that the input size 128×128 obtains the best PSNR because the tested image size is around 1080p, resulting that 128×128 is a proper size for semantics reconstruction of GANs. Second, the effect of code sizes and interpolation sizes is given in Table 3.2. Code size is the length of generated compressed code N . We set the input size as 128×128 . Along with the increase of code sizes, PSNR and MS-SSIM increases. The GAN with fixed code size cannot provide good performance for all the images with different textures, so an adaptively switchable encoder for GANs with different code sizes will be studied in the future. To obtain variable bit rates, I add one bicubic interpolation filter with different scales as the preprocessing. From Table 3.2, by interpolating the size from (W, H) to (2W, 2H), PSNR increases by about 2.2dB, MS-SSIM increases by around 0.10. Meanwhile, the

rate increases up to almost 4 times. The effect of different quantization bits is shown in Table 3.3. We set the code size as 256 with (4W, 4H) interpolation. Too few bits, e.g. 5 bit, will destroy the reconstruction quality significantly. Similar to CAE-based method, I also apply PCA to further reduce the code size. For the CLIC challenge, the entry Gcode uses the architecture of $128 \times 128 \times 3 \rightarrow 128$ with (3W, 3H) interpolation, 8-bit quantization and PCA rotation.

Table 3.1 The effect of different input sizes.

Input size	PSNR(dB)	MS-SSIM	Rate(bpp)
$64 \times 64 \times 3 \rightarrow 1024$	22.73	0.745	0.781
$128 \times 128 \times 3 \rightarrow 1024$	23.95	0.897	0.225
$256 \times 256 \times 3 \rightarrow 1024$	17.18	0.699	0.050

Table 3.2 The effect of different code sizes and interpolation sizes.

Code size	Interp. Size	PSNR(dB)	MS-SSIM	Rate(bpp)
64	(W,H)	22.195	0.753	0.024
	(2W,2H)	24.213	0.856	0.086
	(4W,4H)	26.451	0.928	0.329
128	(W,H)	23.126	0.791	0.042
	(2W,2H)	25.335	0.901	0.162
	(4W,4H)	27.801	0.941	0.389
256	(W,H)	23.962	0.831	0.071
	(2W,2H)	26.308	0.924	0.274
	(4W,4H)	29.262	0.957	0.792
1024	(W,H)	25.121	0.896	0.261
	(2W,2H)	27.474	0.947	0.981

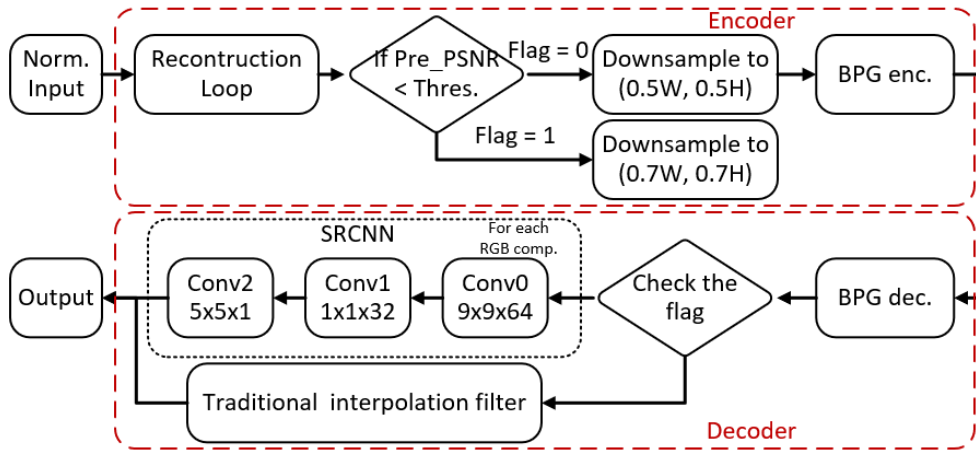
3.2.1.3 Super-Resolution for Compression

Using super-resolution as a post filter is an intuitive method for compression. We present a SR based compression architecture in Figure 3.4. We use a SRCNN architecture in [86] with three convolutional layers. The kernel sizes are set as 9, 1, 5, and the numbers of convolutional filters are

Table 3.3 The effect of different quantization bits.

Quan. bit	PSNR (dB)	MS-SSIM	Rate (bpp)
8 bit	27.932	0.952	0.764
7 bit	27.788	0.942	0.472
6 bit	27.313	0.901	0.352
5 bit	25.179	0.784	0.233

set as 64, 32, 1. We retrain this SRCNN model with the scale of 2 using the CLIC training dataset. The loss function and training parameters are kept the same as [86].

**Fig. 3.4** Block diagram of super-resolution based compression.

However, for images with complex textures or with small resolution, SR will become the bottleneck of high quality reconstruction. Thus, I propose an adaptive strategy by building a reconstruction loop in the encoder. This loop calculates the distortion only caused by SR, i.e. Pre_PSNR in Figure 3.4. When Pre_PSNR is larger than a pre-defined threshold, images are downsampled to (0.5W, 0.5H) and a SRCNN filter is conducted after decoding. Otherwise, images are downsampled to (0.7W, 0.7H) and a lanczos filter is alternatively applied for interpolation. The effect of adaptive strategy is listed in Table 3.4. The threshold is set as 33.0 dB in our experiments and about 30% of images are selected to use SRCNN filters. For the CLIC challenge, the entry Kattolab uses adaptive SR-based architecture.

To measure the coding efficiency, the rate is measured by bit per pixel (bpp). PSNR (dB) and MS-SSIM are used to measure objective and subjective qualities, respectively.

Table 3.4 The effect of adaptive strategy for super-resolution.

Case	PSNR (dB)	MS-SSIM	Rate (bpp)
qp=32, Non-adaptive	29.418	0.949	0.151
qp=35, Adaptive	30.002	0.945	0.156

3.2.1.4 Comparison Results

In this section, I use the CLIC validation dataset for a fair evaluation. The RD curves with MS-SSIM and PSNR are shown in Figure 3.5. RD curves for super-resolution is short because it is conducted by changing the threshold in the adaptive strategy with the fixed quantization parameter (QP) value in BPG codec. By changing the QP, super-resolution can also achieve a wide range of RD curves. Several observations are summarized from RD curves. 1) CAEs are better than JPEG in case of lossy compression due to the inherent property of autoencoder. Autoencoders can reduce the dimension to extract the compressed presentation from images, so CAEs outperform JPEG and JPEG2000. 2) GANs perform better with low bit rate than that with high bit rate, so GANs tend to achieve large compression ratio. Meanwhile, GANs have better performance on MS-SSIM than PSNR, because the reconstruction of GANs is based on the distribution of the image data, which is friendly to human visual system. Especially for MS-SSIM, GANs have stable performance from 0.2bpp to 0.8bpp. 3) SR achieves the best performance among these three methods, because it takes the advantages of both emerging algorithms BPG and machine learning based super-resolution filters. Promising results can be expected to outperform BPG by adding better super-resolution filters, if more computational resources can be provided.

The comparison for three methods with the rate constraint of 0.15bpp is shown in Table 3.5. It is observed that SR-based method is quite close to BPG. GAN and CAE based architectures are better than JPEG. Especially, GANs and CAEs have the similar PSNR, but GANs are much better than CAEs in terms of relatively subjective MS-SSIM.

3.2.2 Deep Residual Learning for Learned Image Compression

The network architectures that we used as anchors are illustrated in Fig. 3.6. This architecture is referred from the work [55] and the work [57], which has achieved the state-of-the-art compression efficiency. The network consists of two autoencoders. The main autoencoder controls the rate-distortion optimization for image compression, whose loss is formulated by

$$J = \lambda d(\mathbf{x}, \hat{\mathbf{x}}) + R(\hat{\mathbf{y}}) \quad (3.3)$$

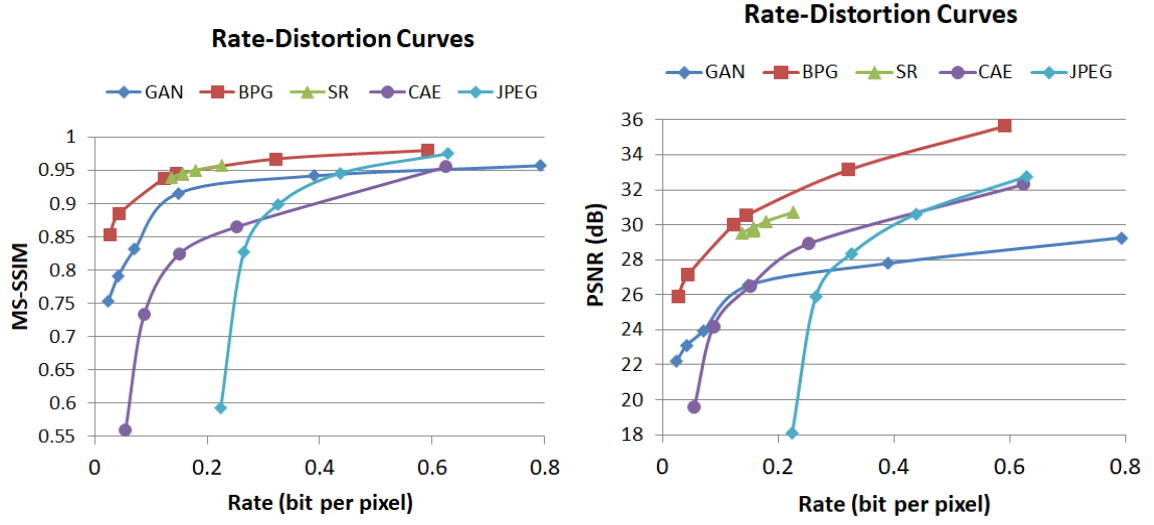


Fig. 3.5 RD curves of three methods.

Table 3.5 Performance comparison with 0.15bpp constraint.

Codecs	PSNR (dB)	MS-SSIM	Rate (bpp)
JPEG	25.82	0.853	0.133
CAE	26.48	0.825	0.151
GAN	26.53	0.915	0.148
SR	30.00	0.947	0.143
BPG	30.85	0.948	0.149

where λ is a parameter to measure the rate-distortion tradeoff. The auxiliary autoencoder is used to encode the side information to model the distribution of compressed information. Gaussian scale mixture is used to estimate an image-dependent and adaptive entropy model, where scale parameters are conditioned on a hyperprior. Moreover, [57] proposed a joint autoregressive and hyperprior approach, denoted as Joint. The only difference is to append a masked 5×5 convolution after quantization and to concatenate the output of auxiliary autoencoder and masked convolution together to learn the entropy model.

3.2.2.1 From Small Kernel Size to Large Kernel Size

In classical image compression algorithms, transform filter sizes are quite important to improve the coding efficiency, especially for UHD videos. From the smallest transform size 4×4 , larger and larger transform size is gradually used into video coding algorithms. Specifically, up to

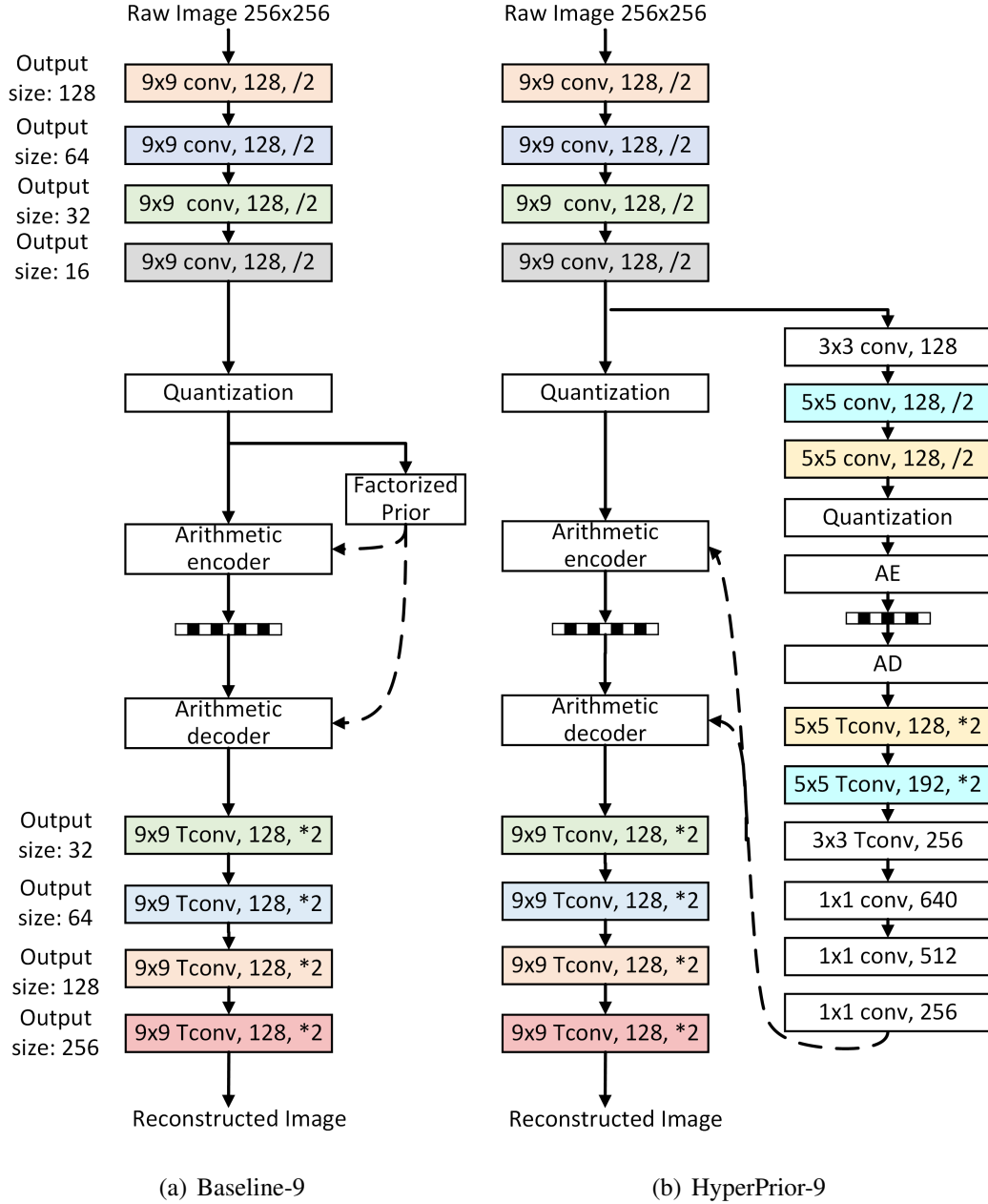


Fig. 3.6 The network structure of anchors we used.

32×32 DCT coefficients have been incorporated into HEVC [47]. Large kernel size brings benefits on capturing the spatial correlation and semantic information. Motivated from this, we conduct some experiments using Kodak dataset [77] with different filter sizes in the main and auxiliary autoencoders respectively to explore the effect of larger kernel size on coding efficiency. Table 3.6 shows for the Baseline architecture, along with the increasing of kernel sizes, the rate-distortion performance are becoming better. Table 3.7 demonstrate similar results for HyperPrior architectures. Table 3.8 shows large kernel in the auxiliary autoencoder cannot bring

Table 3.6 The effect of kernel size on Baseline on Kodak, optimized by MSE with $\lambda = 0.015$.

Method	PSNR	MS-SSIM	Rate
Baseline-3	32.160	0.9742	0.671
Baseline-5	32.859	0.9766	0.641
Baseline-9	32.911	0.9776	0.633

Table 3.7 The effect of kernel size on HyperPrior on Kodak, optimized by MSE with $\lambda = 0.015$.

Method	PSNR	MS-SSIM	Rate
HyperPrior-3	32.488	0.9742	0.543
HyperPrior-5	32.976	0.9757	0.518
HyperPrior-9	33.005	0.9765	0.512

any benefits on RD performance and even gets worse, because the compressed codes y has small size, so 5×5 are large enough. Too many learnable parameters instead increase the difficulty to learn. It is worth noting for Joint architecture [57], a sequential decoding is inevitable, which is extremely time-consuming when the test image becomes larger. Therefore, we exclude the masked convolution in this challenge, but keep the 1×1 conv as they are, for HyperPrior architecture. An ablation on the effect of 1×1 conv will be conducted in the future.

3.2.2.2 From Shallow Network to Deep Residual Network

With respect to the receptive field, the stack of four 3×3 kernels capture the same receptive field as one 9×9 kernel with fewer parameters. We have tried to replace one large kernel with several 3×3 filters, however, experiment shows the stack of 3×3 kernels cannot converge. Motivated from [69], we add the shortcut connection for neighboring 3×3 kernels. Our proposed deep residual network for image compression is shown in Fig. 3.7. Fig. 3.7(a) is denoted as $3 \times 3(3)$, where the stack of three 3×3 kernels reaches the same receptive field as 7×7 . The architecture of Fig. 3.7(b) is ResNet- $3 \times 3(4)$, where the stack of four 3×3 kernels reaches the same receptive field as 9×9 . As for the activation functions, to prevent more parameters overhead, we only use GDN/IGDN [54] for one time in each residual unit when the output size changes. For other convolutional layers, we use parameter-free Leaky ReLU as activation function to add the non-linearity in the networks.

Table 3.8 The effect of kernel size in the auxiliary autoencoder on Kodak, optimized by MS-SSIM with $\lambda = 5$.

Method	PSNR	MS-SSIM	Rate
HyperPrior-9-Aux-5	26.266	0.9591	0.169
HyperPrior-9-Aux-9	26.236	0.9590	0.171

Table 3.9 Comparison of residual networks and upsampling operations on Kodak, optimized by MS-SSIM with $\lambda = 5$.

Method	PSNR	MS-SSIM	Rate
Hyperprior-9	26.266	0.9591	0.1690
ResNet-3×3(3)	26.378	0.9605	0.1704
ResNet-3×3(4)-TConv	26.457	0.9611	0.1693
ResNet-3×3(4)-SubPixel	26.498	0.9622	0.1700

The shortcut projection is shown in Fig. 3.8. As shown in Table 3.9, ResNet-3×3(4) is better than ResNet-3×3(3) and Hyperprior-9.

3.2.2.3 Upsampling Operations at Decoder Side

The encoder-decoder pipeline is a symmetric architecture. The down-sampling operations at the encoder side are intuitively implemented using convolution filters with stride, however, up-sampling operations at the decoder side have various ways, including bicubic interpolation [71], transposed convolution [72], sub-pixel convolution [73]. Typically, almost all the previous works use the transposed convolution (TConv), except for the work [53] use sub-pixel convolution at the decoder side. Considering for fast end-to-end learning, we exclude bicubic interpolation and compare two popular up-sampling operations, i.e. TConv and Sub-pixel Conv. For sub-pixel conv, we increase the number of channels by 4 times and then use `tf.depth_to_space` function in Tensorflow. Results in Table. 3.9 show sub-pixel convolution filters bring some improvement on PSNR and MS-SSIM than transposed convolution filters.

3.2.2.4 Implementation Details

For training, we use 256×256 patches cropped from ILSVRC validation dataset (ImageNet [76]). Batch size is 8, and up to 2M iterations are conducted to reach stable results. The model was

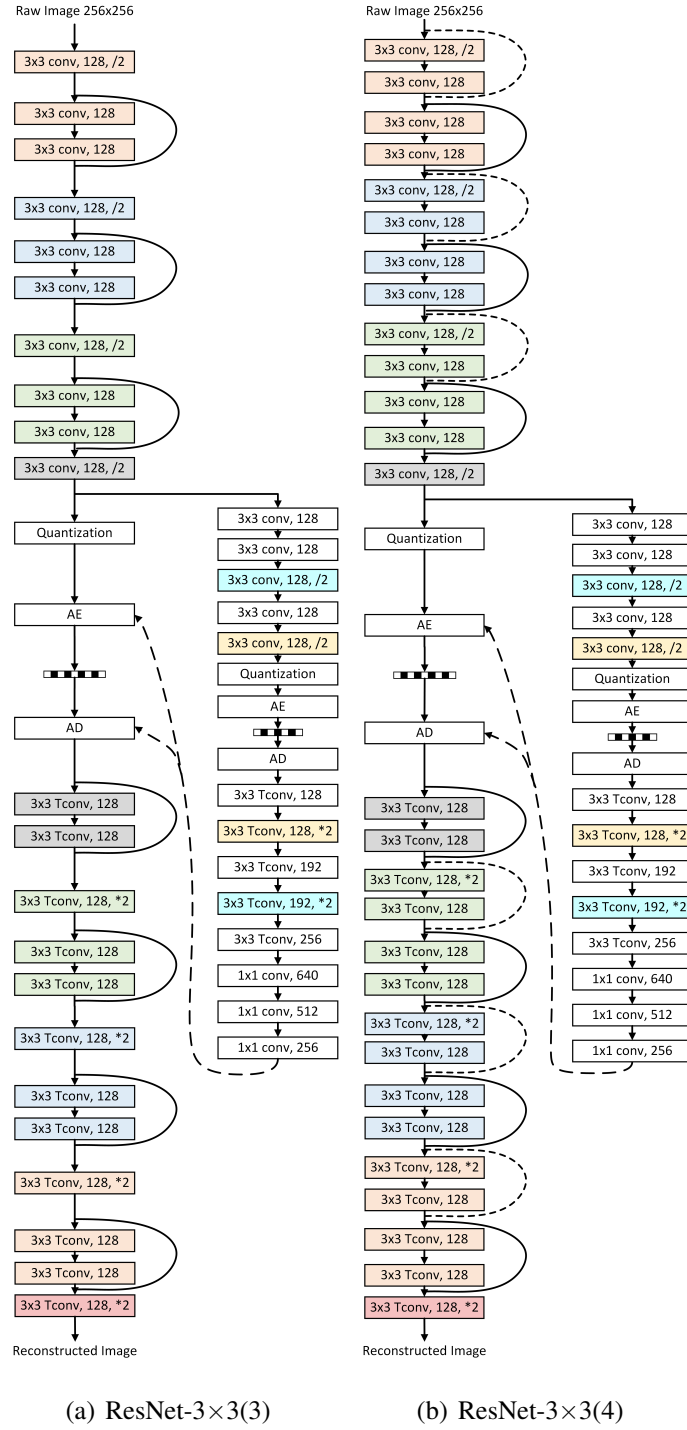


Fig. 3.7 Network structure of proposed deep residual learning, where the solid and dotted lines denote the shortcut connection without and with size change, respectively.

optimized using Adam [70], and the learning rate was maintained at a fixed value of 1×10^{-4} and was reduced to 1×10^{-5} for the last 80K iterations.

We also use two strategies for CLIC2019. One is **Wide Bottleneck**. More filters can increase

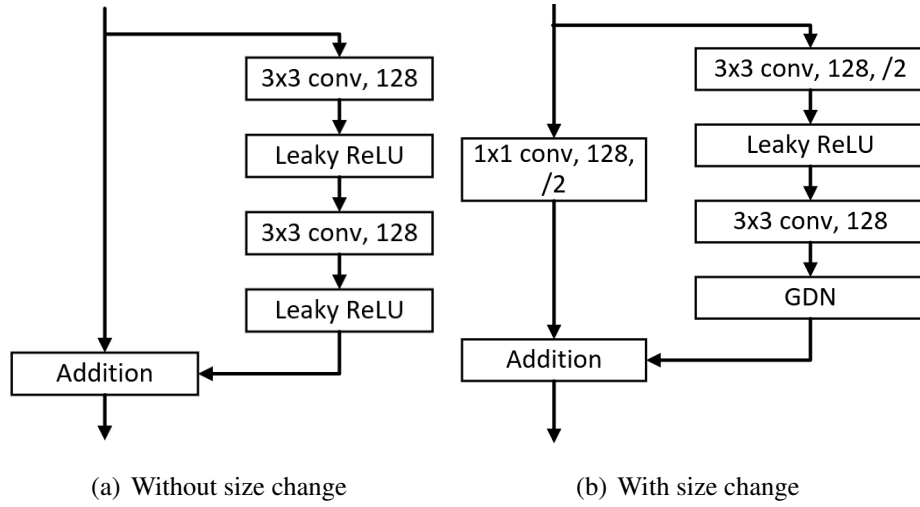


Fig. 3.8 The network structure of one residual unit.

Table 3.10 The effect of wide bottleneck on Kodak dataset.

Method	PSNR	MS-SSIM	Rate
ResNet-3x3(4)-Bottleneck128	26.498	0.9622	0.1700
ResNet-3x3(4)-Bottleneck192	26.317	0.9619	0.1667

the model capacity. Regarding that increasing the number of filters for large feature maps will significantly increase FLOPs, we only increase the number of filters in the last layer of encoder from 128 to 192, so that FLOPs are only increased a little, from 2.50×10^{10} to 2.56×10^{10} . Results are compared in Table. 3.10. Bottleneck192 reduces the bitrate a lot, but also degrades quality compared to Bottleneck128.

The other is **Rate Control**. For the low-rate track, 0.15bpp is the hard threshold. We train two models at different bit rates by adjusting λ , where the averaged rate with $\lambda = 5$ is less than 0.15bpp for the validation dataset, and the averaged rate with $\lambda = 8$ is larger than 0.15bpp. Results are shown in Table. 3.11. Then we can encode all the test images twice and select adaptive to push

Table 3.11 Rate control on CLIC validation dataset [85].

Method	λ	PSNR	MS-SSIM	Rate
ResNet-3x3(4)-Bottleneck192	5	29.708	0.9697	0.1369
ResNet-3x3(4)-Bottleneck192	10	30.710	0.9765	0.1816

Table 3.12 Results on CLIC validation dataset [85].

Entry	Description	PSNR	MS-SSIM	Rate
Kattolab	HyperPrior-9	28.902	0.9674	0.134
Kattolab	HyperPrior-9 + Rate Control	29.102	0.9701	0.150
Kattolab	ResNet-3×3(4)-TConv + Rate Control	29.315	0.9716	0.150
Kattolabv2	ResNet-3×3(4)-SubPixel+ Rate Control	29.300	0.9720	0.150
KattolabSSIM	ResNet-3×3(4)-SubPixel + Wide Bottleneck + Rate Control	29.211	0.9724	0.150

the rate to 0.15bpp with the maximized MS-SSIM. One bit should be added into the bitstream to specify which model is used for decoding, which will not increase the complexity of the decoder.

3.2.2.5 Compression Results

The compression results of our approaches on CLIC validation dataset are summarized in Table 3.12.

Although deep residual network brings the coding gain, the model size grows significantly. In this section, we will analyze the number of parameters and the model complexity with respect to floating point operations per second (FLOPs) for all kinds of architectures. Specifically, take the architecture HyperPrior-9 as an example, the layer-wise model size analysis is illustrated in Table 3.13. The number of parameters and FLOPs are calculated by

$$\begin{aligned} \text{Para} &= (h \times w \times C_{in} + 1) \times C_{out} \\ \text{FLOPs} &= \text{Para} \times H' \times W' \end{aligned} \quad (3.4)$$

where $h \times w$ is the kernel size, $H' \times W'$ is the output size. C_{in} and C_{out} are the number of channels before or after one operation. If no bias is applied, the $+1$ are removed, such as conv4. Quantization and leaky-ReLU are parameter-free. GDN [56] only run across different channels, but not across different spatial positions, the number of parameters of GDN is only $(C_{in} + 1) \times C_{out}$. FLOPs of the total GDN and inverse GDN calculation is only 7.10×10^8 . This paper mainly focus on the backbone of convolutional layers, so we omit the FLOPs of GDN, inverse GDN and factorized prior. The comparison is listed in Table 3.14, where the last column is relative value of FLOPs using Baseline-5 [54] as a baseline model. ResNet-3×3(4) also denotes ResNet-3×3(4)-TConv. Our models achieve better coding performance with low complexity.

In conclusion, I have described the proposed deep residual learning and sub-pixel convolution for image compression. This is the basis of our submitted entries Kattolab, Kattolabv2 and KattolabSSIM. Results have shown our approaches achieve 0.972 in MS-SSIM at the rate of

Table 3.13 The model size analysis of HyperPrior-9.

Layer	Kernel		Channel		Output		Para	FLOPs
	h	w	C_{in}	C_{out}	H'	W'		
conv1	9	9	3	128	128	128	31232	5.12×10^8
conv2	9	9	128	128	64	64	1327232	5.44×10^9
conv3	9	9	128	128	32	32	1327232	1.36×10^9
conv4	9	9	128	128	16	16	1327104	3.40×10^8
GDN/IGDN							99072	-
Hconv1	3	3	128	128	16	16	147584	3.78×10^7
Hconv2	5	5	128	128	8	8	409728	2.62×10^7
Hconv3	5	5	128	128	4	4	409728	6.56×10^6
FactorizedPrior							5888	-
HTconv1	5	5	128	128	8	8	409728	2.62×10^7
HTconv2	5	5	128	192	16	16	614592	1.57×10^8
HTconv3	3	3	192	256	16	16	442624	1.13×10^8
layer1	1	1	256	640	16	16	164480	4.21×10^7
layer2	1	1	640	512	16	16	328192	8.40×10^7
layer3	1	1	512	256	16	16	131072	3.36×10^7
Tconv1	9	9	128	128	32	32	1327232	1.36×10^9
Tconv2	9	9	128	128	64	64	1327232	5.44×10^9
Tconv3	9	9	128	128	128	128	1327232	2.17×10^{10}
Tconv4	9	9	128	128	256	256	31107	2.04×10^9
Total							11188291	3.88×10^{10}

0.15bpp with moderate complexity during the validation phase.

3.3 Proposed Learned Image Compression through Principle Component Analysis (PCA)

In this section, I present a convolutional autoencoder (CAE) based lossy image compression architecture. The main contributions are twofold.

- 1) To implement the transform coding, I design a symmetric CAE structure with multiple

Table 3.14 The model complexity of different architectures.

Method	Para	FLOPs	Relative
Baseline-3	997379	4.25×10^9	0.36
Baseline-5	2582531	1.18×10^{10}	1.00
Baseline-9	8130563	3.82×10^{10}	3.24
HyperPrior-3	4055107	4.78×10^9	0.40
HyperPrior-5	5640259	1.23×10^{10}	1.04
HyperPrior-9	11188291	3.88×10^{10}	3.28
ResNet-3×3(3)	5716355	1.75×10^{10}	1.48
ResNet-3×3(4)	6684931	2.43×10^{10}	2.06
ResNet-3×3(4)-SubPixel	8172172	2.50×10^{10}	2.12
ResNet-3×3(4)-SubPixel-Bottleneck192	11627916	2.56×10^{10}	2.17

cascaded downsampling and upsampling units to generate feature maps with low dimensions. We optimize this CAE using an approximated rate-distortion loss function.

- 2) To generate a more energy-compact representation, I propose a principal components analysis (PCA)-based rotation to generate more zeros in the feature maps. Then, the quantization and entropy coder are followed to compress feature maps further.

Experimental results show my method outperforms JPEG and JPEG2000 in terms of PSNR, and achieves a 13.7% BD-rate decrement compared to JPEG2000 with the popular Kodak database images. In addition, our method is computationally more appealing compared to other autoencoder based image compression methods.

3.3.1 Proposed Convolutional Autoencoder (CAE) Network

The block diagram of the proposed image compression based on CAE is illustrated in Fig.3.9. The encoder part includes the pre-processing steps, CAE computation, PCA rotation, quantization, and entropy coder. The decoder part mirrors the architecture of the encoder.

To build an effective codec for image compression, I train this approach in two stages. First, a symmetric CAE network is designed using convolution and deconvolution filters. Then, I train this CAE greedily using an RD loss function with an added uniform noise, which is used to imitate the quantization noises during the optimizing process. Second, by analyzing the produced feature maps from the pre-trained CAE, I utilize the PCA rotation to produce more zeros for improving the coding efficiency further. Subsequently, quantization and entropy coder are used to compress

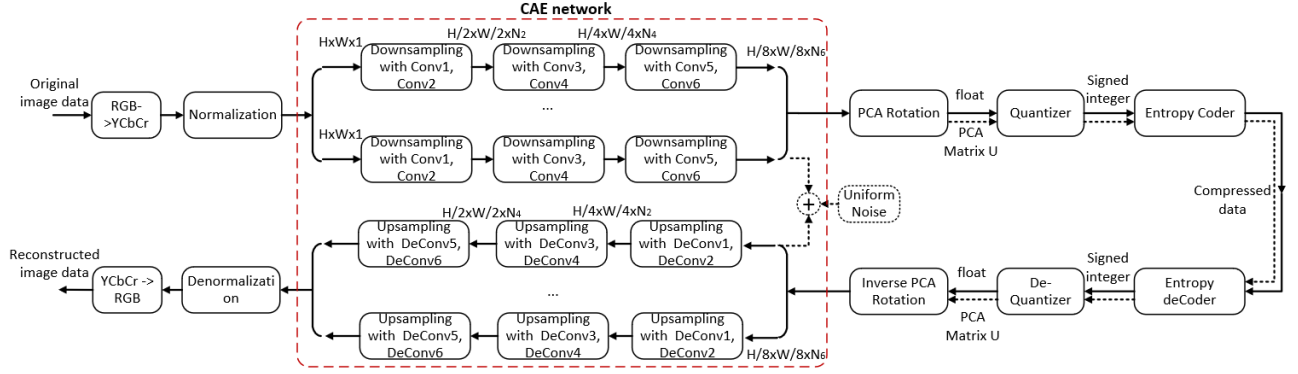


Fig. 3.9 Block diagram of the proposed CAE based image compression. (The detailed block for downsampling/upsampling is shown in Fig. 3.10)

the rotated feature maps and the side information for PCA (matrix U) to generate the compressed bitstream. Each of these components will be discussed in detail in the following.

As the pre-processing steps before the CAE design, the raw RGB image is mapped to YCbCr images and normalized to $[0,1]$. For general purposes, I design the CAE for each luma or chroma component; therefore, the CAE network handles inputs of size $H \times W \times 1$. When the size of raw image is larger than $H \times W$, the image will be split into non-overlapping $H \times W$ patches, which can be compressed independently.

The CAE network can be regarded as an analysis transform with the encoder function, $y = f_{\theta}(x)$, and a synthesis transform with the decoder function, $\hat{x} = g_{\phi}(y)$, where x , \hat{x} , and y are the original images, reconstructed images, and the compressed data, respectively. θ and ϕ are optimized parameters inside them.

To obtain the compressed representation of the input images, downsampling/upsampling operations are required in the encoding/decoding process of CAE. However, consecutive downsampling operations will reduce the quality of the reconstructed images. In the work [53], it points out that the super resolution is achieved more efficiently by first convolving images and then upsampling them. Therefore, I propose a pair of convolution/deconvolution filters for upsampling or downsampling, as shown in Fig. 3.10, where N_i denotes the number of filters in the convolution or deconvolution block. By setting the stride as 2, I can get downsampled feature maps. The padding size is set as one to maintain the same size as the input. Unlike the work [53], I do not use residual networks and sub-pixel convolutions, instead, I apply deconvolution filters to achieve a symmetric and simple CAE network.

In traditional codecs, the quantization is usually implemented using the round function (denoted as $[\cdot]$), and the derivative of the round function is almost zero except at the integers. Due to the non-differentiable property of rounding function, the quantizer cannot be directly

incorporated into the gradient-based optimization process of CAE. Thus, some smooth approximations are proposed in related works. Theis et al. [53] proposed to replace the derivative in the backward pass of back propagation as $\frac{d}{dy}([y]) \approx 1$. Balle et al. [54] replaced the quantization by an additive uniform noise as $[y] \approx y + \mu$. Toderici et al. [62] used a stochastic binarization function as $b(y) = -1$ when $y < 0$, and $b(y) = 1$ otherwise. In our method, I use the simple uniform noises intuitively to imitate the quantization noises during the CAE training. After CAE training, I apply the real round-based quantization in the final image compression. The network architecture of CAE is shown in Fig. 3.9, in which N_i denotes the number of filters in each convolution layer and determines the number of generated feature maps.

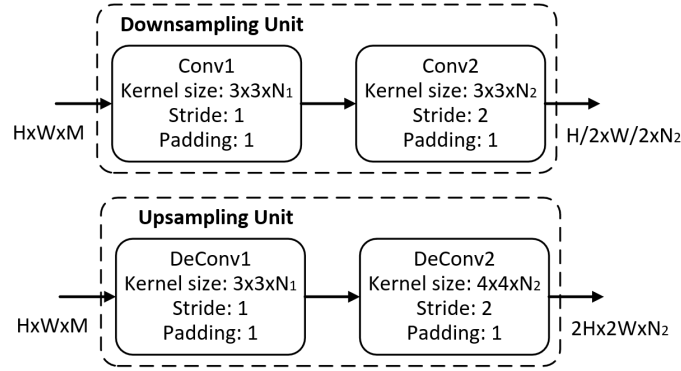


Fig. 3.10 Downsampling/Upsampling Units with two (De)Convolution Filters.

As for the activation function in each convolution layer, I utilize the Parametric Rectified Linear Unit (PReLU) function [68], instead of the ReLU which is commonly used in the related works. The performance with ReLU and PReLU functions are shown in Fig. 3.11. Compared to ReLU, PReLU can improve the quality of the reconstructed images, especially for high bit rate.

Inspired by the rate-distortion cost function in the traditional codecs, I optimize this CAE using the loss as

$$\begin{aligned} J(\theta, \phi; x) &= \|x - \hat{x}\|^2 + \lambda \cdot \|y\|^2 \\ &= \|x - g_\phi(f_\theta(x) + \mu)\|^2 + \lambda \cdot \|f_\theta(x)\|^2 \end{aligned} \quad (3.5)$$

where μ is uniform noise. λ is a parameter to measure the tradeoff. $\|f_\theta(x)\|^2$ denotes the amplitude of the compressed data y , which reflects the number of bits used to encode the compressed data. In this work, the CAE model was optimized using Adam [70], and was applied to images with the size of $H \times W$. We used a batch size of 16 and trained the model up to 8×10^5 iterations, but the model reached convergence much earlier. The learning rate was kept at a fixed value of 0.0001, and the momentum was set as 0.9 during the training process.

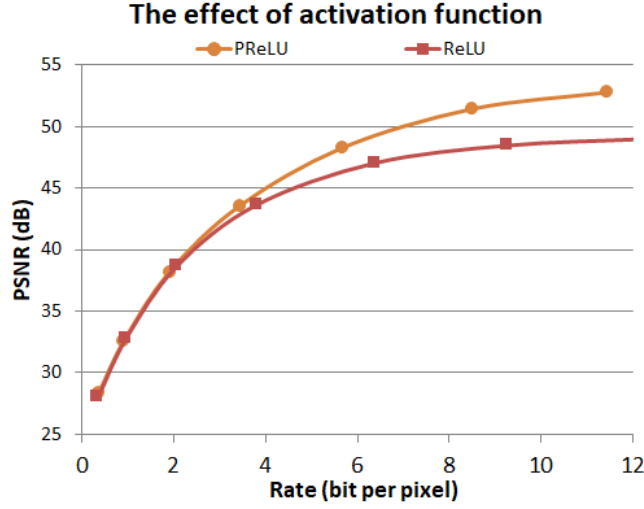


Fig. 3.11 The effect of activation function in CAE.

3.3.2 Principle Component Analysis (PCA) Rotation

After the CAE computation, an image representation with a size of $\frac{H}{8} \times \frac{W}{8} \times N_6$ is obtained for each $H \times W \times 1$ input, where N_6 denotes the number of filters in the sixth convolution layer of the encoder part. Three examples of the feature maps for the 512×512 images cropped from Kodak databases [77] are demonstrated in the second column of Fig. 3.12. It can be observed that each feature map can be regarded as one high-level representation of the raw images.

To obtain a more energy-compact representation, I decorrelate each feature map by utilizing the principle component analysis (PCA), because PCA is an unsupervised dimensionality reduction algorithm and is suitable for learning the reduced features as a supplementary of CAE. The generated feature maps are denoted as $y = \frac{H}{8} \times \frac{W}{8} \times N_6$, and y is reshaped as N_6 -dimensional data. PCA is performed using the following steps. The first step is to compute the covariance matrix of z as follows:

$$\Sigma = \frac{1}{m} \sum_{i=1}^m (y)(y)^T \quad (3.6)$$

where m is the number of samples for y . The second step is to compute the eigenvectors of Σ and stack the eigenvectors in columns to form the matrix U . Here, the first column is the principal eigenvector corresponding to the largest eigenvalue, the second column is the second eigenvector, and so on. The third step is to rotate the N_6 -dimensional data y by computing

$$y_{rot} = U^T y \quad (3.7)$$

By computing y_{rot} , I can ensure that the first feature maps have the largest value, and the features maps are sorted in descending order. Experimental results demonstrate that the

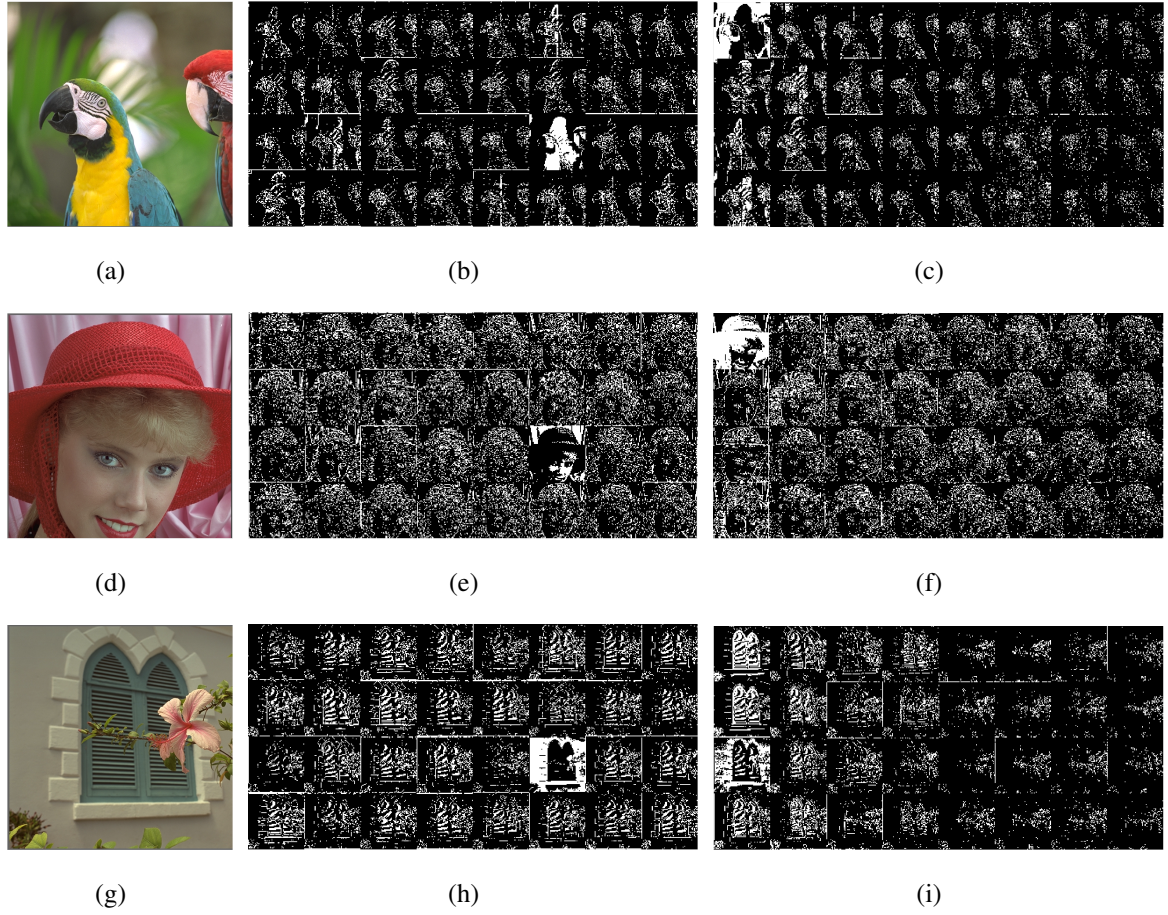


Fig. 3.12 Examples of three images and their corresponding feature maps arranged in raster-scan order ($N_6 = 32$): (a)(d)(g) Raw images, (b)(e)(h) Generated 32 feature maps for Y-component by CAE, and the size of each feature map is $\frac{H}{8} \times \frac{W}{8}$, (c)(f)(i) Rotated Y feature maps by PCA, arranged in vertical scan order.

vertical-scan order for the feature maps works a little better than diagonal scan and horizontal scan; therefore, I arrange the feature maps in vertical scan as shown in the third column of Fig. 3.12. It can be observed that more zeros are generated in the bottom-right corner and large values are centered in the top-left corner in the rotated feature maps, which can benefit the entropy coder to achieve large compression ratio.

After the PCA rotation, the quantization is performed as

$$y' = \lceil 2^{B-1} \cdot y_{rot} \rceil \quad (3.8)$$

where B denotes the number of bits for the desired precision, which is set as 12 in our model.

As for the entropy coder, I use the JPEG2000 entropy coder to decompose y' into bitplanes and apply the adaptive binary arithmetic coder. It is noted that JPEG2000 entropy coder applies EBCOT (Embedded block coding with optimized truncation) algorithm to achieve a desired rate

R , which is also referred to as post-compression RD optimization. In our method, the feature maps rotated by PCA have many zeros; therefore, assigning the target bits R can further improve the coding efficiency.

In the decoder part, de-quantization is performed as

$$\tilde{y} = \frac{y'}{2^{B-1}} \quad (3.9)$$

After obtaining the float-point number \tilde{y} from the bitstream, I recover the feature maps from the rotated data by using

$$\hat{y} = U\tilde{y} \quad (3.10)$$

Then, the CAE decoder network will reconstruct the images using $\hat{x} = g_{\phi}(\hat{y})$. The side information of PCA rotation is the matrix U with a dimension of $N_6 \times N_6$ for each image. We also quantize U and encode it. The bits for U is added to the final rate as the side information in the experimental results.

3.4 Proposed Learned Image Compression through Energy Compaction

In this section, I first describe the proposed convolutional autoencoder (CAE) architecture as a baseline, as shown in Fig. 3.13. Secondly, I present the mathematical analysis of the energy compaction property and propose a normalized coding gain metric. Thirdly, based on the proposed coding gain metric, I propose an energy compaction-based training strategy to help the baseline CAE network achieve better results.

3.4.1 Convolutional Autoencoder (CAE) Architecture

According to the transform coding theory, the compression system can be considered as an analysis transform $\mathbf{y} = f_{\theta}(\mathbf{x})$, and a synthesis transform $\hat{\mathbf{x}} = g_{\phi}(\hat{\mathbf{y}})$, where \mathbf{x} , $\hat{\mathbf{x}}$, \mathbf{y} , and $\hat{\mathbf{y}}$ are the original images, reconstructed images, compressed data (also called latent presentation) before quantization, and quantized compressed data, respectively. Finally, θ and ϕ are optimized parameters in the analysis and synthesis transforms, respectively.

The block diagram of the proposed image compression based on CAE is illustrated in Fig. 3.13. The only pre-processing steps before the CAE network consist of normalizing the raw RGB image to $[-1, 1]$ by calculating $(\frac{\mathbf{x}}{127.5} - 1.0)$. The size of the input is denoted as $H \times W \times C$, where C

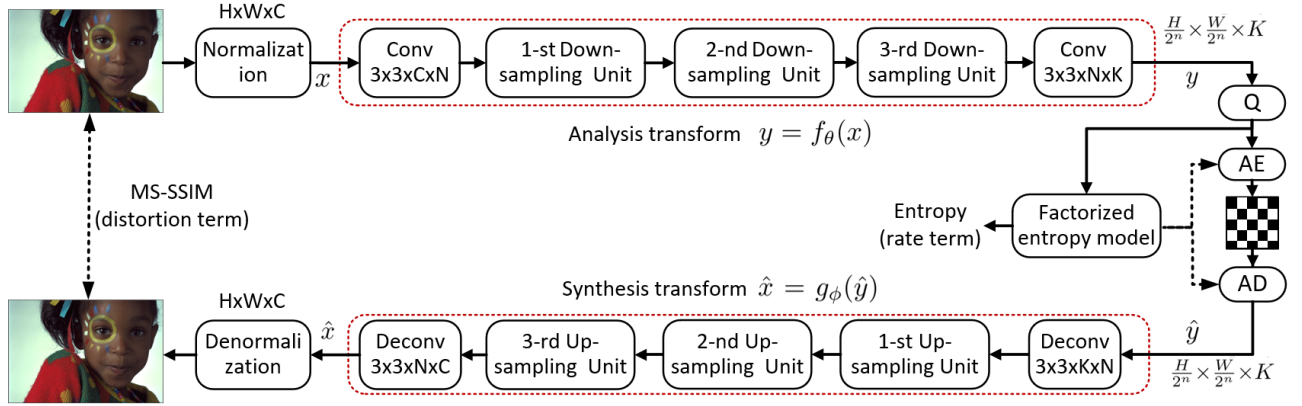


Fig. 3.13 Overview of our proposed CAE image compression architecture. Conv and Deconv use 3×3 the kernel size with a stride of 1. The block for the downsampling/upsampling unit is shown in Fig. 3.14, where Q represents quantization, and AE and AD represent the arithmetic encoder and arithmetic decoder. Here, the factorized entropy model produces a context model that generates an estimated entropy and serves for AE and AD. During the test, I use the JPEG2000 entropy coder. Let us suppose that the number of down(up)sampling units is n and that the number of channels before Q is K , and the latent representation y has the dimension of $\frac{H}{2^n} \times \frac{W}{2^n} \times K$. Additionally, N denotes the number of filters in the downsampling unit. In our experiments, $n = 3$, $K = 48$, $N = 128$, and $C = 3$ for the RGB image.

is set as 3 for the RGB images. The analysis transform and synthesis transform have symmetric networks, apart from using convolutional and de-convolutional filters, respectively.

To obtain a high-level representation with a smaller size, I decomposed the analysis and synthesis transforms into a sequence of consecutive down(up)sampling operations. To leverage the number of filters in the down(up)sampling operations as the constant N , I added one convolutional operation with N filters before the downsampling operations to increase the number of channels to N at the encoder side. Additionally, I added one more convolutional operation with K filters after several downsampling operations to control the number of channels for latent presentation. After one downsampling unit with N filters, the size of the feature maps becomes $\frac{H}{2} \times \frac{W}{2} \times N$. Let us assume that I have n downsampling units, the latent presentation y will have a size of $\frac{H}{2^n} \times \frac{W}{2^n} \times K$ as shown in Fig.3.13. The size of the feature maps at the decoder side mirrors the corresponding structure at the encoder side.

Generally, one downsampling operation can be achieved by using one convolutional filter with a stride of 2. However, consecutive downsampling operations will significantly reduce the quality of the reconstructed images. In [73], the authors pointed out that super resolution is achieved more efficiently by first convolving the images and then upsampling, instead of first upsampling and then convolving. Therefore, I propose two downsampling and upsampling operation designs,

i.e., plain unit and residual unit. Plain units use two convolutional filters, while a residual unit replaces one convolutional filter with a residual block [69]. The network structure is shown in Fig. 3.14. Comparison of different types of downsampling and upsampling units is presented in Section IV.B.

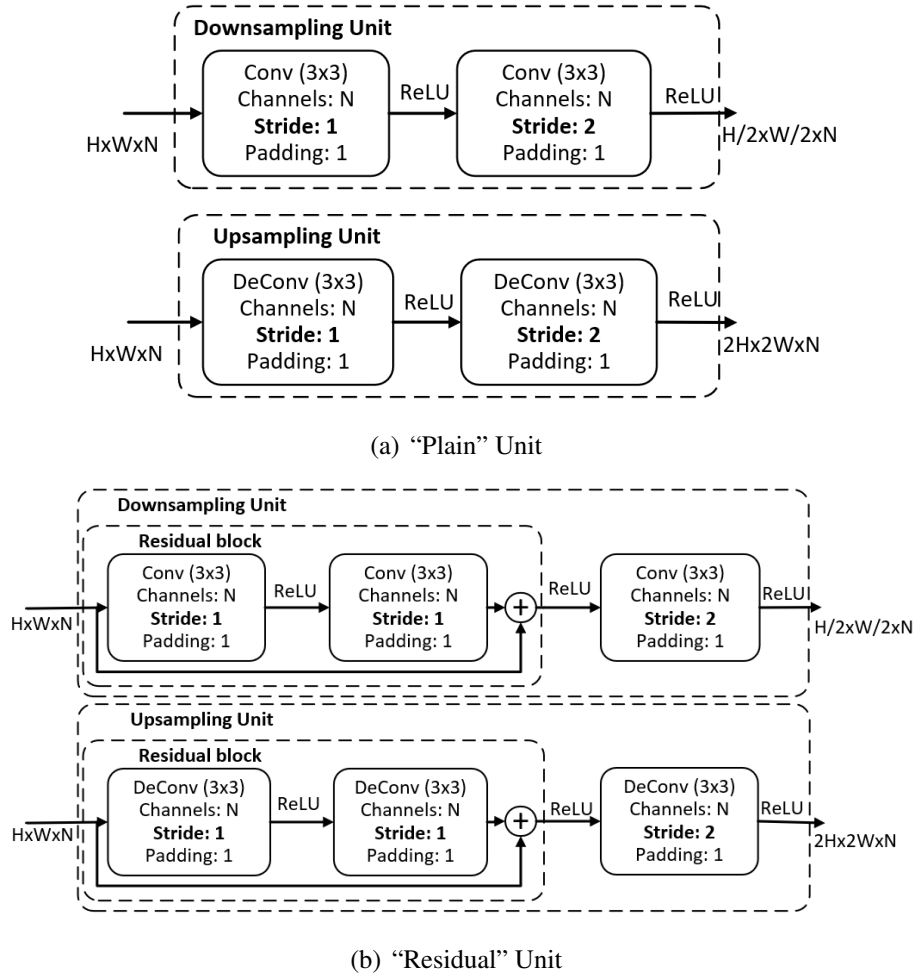


Fig. 3.14 Two Types of Downsampling/Upsampling Units.

Quantization is a necessary component in image compression. In traditional codecs, quantization is implemented by using a round function (denoted as $Q[\cdot]$), and its derivative is almost zero except at the integers. Therefore, it cannot be directly incorporated into the gradient-based optimization process. Several quantization approximations have been proposed, such as uniform noise approximation [54] and soft histogram [58]. In other studies [53] [59], the derivation was replaced in the back propagation only, but it was guaranteed that the quantized value was correct in the forward propagation. Comparison of different quantization approximations is presented in experimental results. For simplicity, I used additive uniform noise

approximation.

Based on the rate-distortion cost function in traditional codecs, I optimize this CAE using loss:

$$J(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \lambda D(\mathbf{x}, \hat{\mathbf{x}}) + R(\hat{\mathbf{y}}) \quad (3.11)$$

All the symbols have been already explained above. R represents number of bits required to encode the quantized compressed data $\hat{\mathbf{y}}$. According to the Shannon theory [93], the rate is lower-bounded by the entropy of the discrete probability distribution of the quantized codes, as follows:

$$R = \mathbb{E}_{\tilde{\mathbf{y}} \sim q} [-\log_2(p_{\tilde{\mathbf{y}}}(\tilde{\mathbf{y}}))] \quad (3.12)$$

where q is the actual distribution of the compressed code $\tilde{\mathbf{y}}$ and $p_{\tilde{\mathbf{y}}}(\tilde{\mathbf{y}})$ is the entropy model. Thus, several entropy estimation methods have been introduced by related studies, including the soft histogram based entropy estimation [58], non-parametric factorized entropy model [55], 3D-CNN based conditional probability model [59], and hyperprior based entropy model [55]. To prevent an excessive network structure overhead to be caused by entropy estimation, I used the univariate non-parametric density model to represent the fully factorized prior [55] to calculate the entropy as a rate.

3.4.2 Proposed Coding Gain Metric based on Mathematical Analysis of Energy Compaction Property

Existing deep learning based image compression studies have already achieved promising performance, as has been reported in [59] [66], by focusing on the selection of neural network architectures and the design of differentiable quantization and entropy estimation models, which are critical components for end-to-end training. More related studies are discussed in Section II. However, because the autoencoder is trained in an unsupervised manner, it is difficult to guarantee that the compressed bits will be allocated in a completely identical manner to achieve the same compression performance every time, even though I train a network with the same structure and the same number of iterations. Few previous studies have discussed bit allocation methods. Therefore, the problem of optimum bit allocation must be solved in the learning based compression approaches. Traditional digital coding theories have revealed that optimum bit allocation is highly relevant to the energy compaction property. A good energy compaction property can lead to a high compression ratio. Therefore, in this paper, I propose an energy compaction-based image compression using a convolutional autoencoder (CAE).

Table 3.15 Notations.

Notation	Defination
k	The index of channels for the last layer, $k \in [0, K - 1]$
\mathbf{y}_k	The compressed data for the k -th channel
\mathbf{q}_k	The quantization error for the k -th channel, $\mathbf{q}_k = \hat{\mathbf{y}}_k - \mathbf{y}_k$
\mathbf{r}	The reconstruction error for the image, $\mathbf{r} = \hat{\mathbf{x}} - \mathbf{x}$
$\sigma_{\mathbf{x}}^2$	The variance of the raw image \mathbf{x}
$\sigma_{\mathbf{y}_k}^2$	The variance of the compressed data \mathbf{y}_k for the k -th channel
$\sigma_{\mathbf{q}_k}^2$	The variance of \mathbf{q}_k for the k -th channel
$\sigma_{\mathbf{r}}^2$	The variance of the reconstruction error \mathbf{r}

More importantly, the problem of optimum bit allocation for learning compression methods has not been investigated yet. Moreover, few studies exist on evaluating the coding performance of neural networks from the viewpoint of the energy compaction property.

To address the abovementioned issues, I analyze the energy property of each generated channel in our CAE architecture and formulate the bit allocation problem to propose a metric and evaluate the coding gain. Based on this analysis, I propose to add a regularizer to train the neural network to learn the latent representation more efficiently and reconstruct the image with higher quality.

Based on the CAE, I will provide a mathematical analysis for the energy compaction property and present a normalized coding gain metric. For ease of searching, Table 3.15 summarizes the notations in the following analysis.

Good energy compaction property is critical for high coding efficiency performance. In traditional digital coding systems, energy compaction implies the production of more zeros for the quantized coefficients, which can achieve a higher compression ratio. Thus, many energy compaction-based coding tools have been developed. For example, DCT and DWT exhibit excellent energy compaction by using a few low frequency coefficients to represent the majority of the energy in the signals and generate many zeros for high frequency coefficients. Previous digital coding theories [74] have provided theoretical solutions to the allocation of bits from the viewpoint linear system energy. In a linear sub-band coding system, for any arbitrary analysis transform (which is not required to be non-orthogonal), for the k -th channel I can obtain the

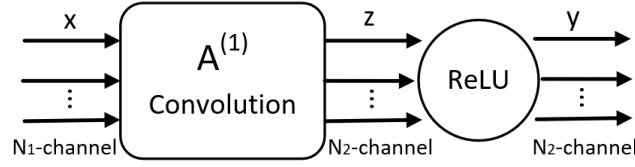


Fig. 3.15 Diagram of a single neuron.

equations from [75],

$$\begin{aligned}\sigma_{y_k}^2 &= A_k \sigma_x^2 \\ \sigma_r^2 &= \sum_{k=0}^{K-1} B_k \sigma_{q_k}^2\end{aligned}\tag{3.13}$$

Then I define two vectors \mathbf{A}_k and \mathbf{B}_k , where $\mathbf{A}_k = \{A_0, A_1, \dots, A_{K-1}\}^T$, determined by input \mathbf{x} and the parameter $\boldsymbol{\theta}$ during the analysis transform; $\mathbf{B}_k = \{B_0, B_1, \dots, B_{K-1}\}^T$, determined by both the quantization errors and the parameter $\boldsymbol{\phi}$ during the synthesis transform. Both \mathbf{A}_k and \mathbf{B}_k have the dimension of $K \times 1$.

However, the CAE network is a non-linear system. Thus, I need to prove that the above equations can be extended to non-linear neural networks. In our CAE network, the compressed data \mathbf{y} consists of multiple channels, and each channel is encoded independently, which is similar to a band in a sub-band coding system. However, owing to the non-linear activation function, the energy compaction property of the CAE system becomes complicated.

First, I analyze a single neuron, as shown in Fig. 3.15. Then, I generalize the energy representation from variance to sum-of-squares. For a linear convolutional operation, this satisfies the following:

$$\mathbb{E}[\mathbf{z}^2] = \mathbf{A}^{(1)} \mathbb{E}[\mathbf{x}^2]\tag{3.14}$$

which is equivalent to $\sigma_z^2 = \mathbf{A}^{(1)} \sigma_x^2$ across the channels when the mean tends to zero. Let us assume that the number of channels in \mathbf{x} and \mathbf{z} are N_0 and N_1 , respectively. $\mathbf{A}^{(1)}$ is a matrix with a dimension of $N_1 \times N_0$, which is determined by the convolutional filter banks. Even if the convolutional filter has the stride of 2, it only brings a downsampling factor, similar to a subband coding system, which does not affect the linearity.

I applied a rectified linear unit (ReLU) $\mathbf{y} = \max\{0, \mathbf{z}\}$ and obtained the following equation:

$$\begin{aligned}\mathbb{E}[\mathbf{y}^2] &= \int_{-\infty}^{+\infty} \max\{0, \mathbf{z}\}^2 p(\mathbf{z}) d\mathbf{z} = \int_0^{+\infty} \mathbf{z}^2 p(\mathbf{z}) d\mathbf{z} \\ &= \frac{1}{2} \int_{-\infty}^{+\infty} \mathbf{z}^2 p(\mathbf{z}) d\mathbf{z} = \frac{1}{2} \mathbb{E}[\mathbf{z}^2]\end{aligned}\tag{3.15}$$

where $p(\mathbf{z})$ is also assumed to be symmetrical in the vicinity of zero. By substituting Eq.(3.14) into

Eq.(3.15), I obtain the following equation:

$$\mathbb{E}[\mathbf{y}^2] = \frac{1}{2} \mathbf{A}_{(1)} \mathbb{E}[\mathbf{x}^2] \quad (3.16)$$

The CAE is considered as a sequence of consecutive neurons, which have convolutional operations and corresponding activation functions. Similar analysis can be applied, and then the parameter $\mathbf{A}^{(1,2,\dots,M)}$ become as follows:

$$\mathbf{A}^{(1,2,\dots,M)} = \left(\frac{1}{2}\right)^{M-1} \mathbf{A}^{(1)} \times \mathbf{A}^{(2)} \dots \mathbf{A}^{(M)} \quad (3.17)$$

where $\mathbf{A}^{(1)} \times \mathbf{A}^{(2)}$ represents matrix multiplication, the number of convolutional layers is M , and the last layer will typically have no activation function. Let us assume that the last layer has K channels; then, $\mathbf{A}^{(1,2,\dots,M)}$ will have the dimension of $K \times N_0$. The input image x generally has three RGB channels or one gray channel. However, I calculate the mean energy for the RGB images. Then, N_0 will become 1 and $\mathbf{A}^{(1,2,\dots,M)}$ has a dimension of $K \times 1$. Therefore, I can define \mathbf{A}_k simply by calculating the sum-of-squares of \mathbf{x} and \mathbf{y}_k , as follows:

$$\mathbf{A}_k \triangleq \mathbf{A}^{(1,2,\dots,M)} = \frac{\mathbb{E}[\mathbf{y}_k^2]}{\mathbb{E}[\mathbf{x}^2]} \quad (3.18)$$

If I train the neural network with a sufficiently large number of images, the mean of \mathbf{y}_k and \mathbf{x} can be regarded as zero based on the statistics. According to the definition of $\sigma_{\mathbf{x}}^2 = \mathbb{E}[(\mathbf{x} - \mu)^2]$, I can obtain the following equation:

$$\mathbf{A}_k = \frac{\sigma_{\mathbf{y}_k}^2}{\sigma_{\mathbf{x}}^2} \quad (3.19)$$

which implies that, as an energy compaction constraint, CAE will have a similar property as Eq.(3.13) in an analysis transform with a sufficiently large dataset.

Second, I determined how to calculate \mathbf{B}_k in the synthesis transform. During the training stages, the quantization error was assumed as uniform noise; therefore, the quantization errors are not correlated with each other, which results in \mathbf{B}_k only being related to $\{g_\phi\}$. In a linear system, \mathbf{B}_k can be computed by the square-sum of the filter bank coefficients [75]. However, it is difficult for CAE to calculate the square-sum of filter banks with numerous convolutional operations. Thus, I constructed the fake code \mathbf{c}_k by setting the k -th channel as 1 and the other channels as 0, and I feed \mathbf{c}_k to a given pre-trained synthesis transform. Then, I estimate vector $\mathbf{B}_k = \{B_0, B_1, \dots, B_{K-1}\}$ as follows:

$$\mathbf{B}_k = \{\sigma_{\hat{\mathbf{x}}}^2 | (\hat{\mathbf{y}}_k \triangleq \mathbf{c}_k)\}, k \in [0, K-1] \quad (3.20)$$

By constructing the fake codes $\{\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_{K-1}\}$, I can obtain a $K \times 1$ vector \mathbf{B}_k . Note that I construct the fake codes \mathbf{c}_k because \mathbf{B}_k measures the impact of the quantization's degree of error

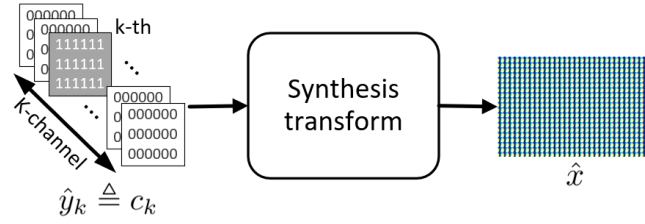


Fig. 3.16 Construction of fake codes to calculate B_k .

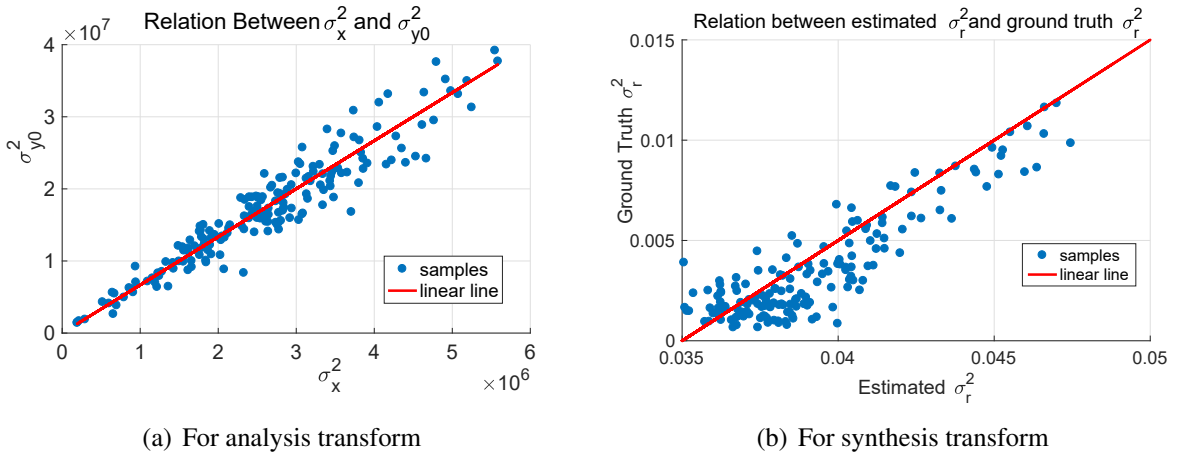


Fig. 3.17 Statistical observations on energy property.

on the final reconstructed errors for a specified channel. When all of the other channels are set as 0, except for the k -th channel, the reconstructed signal \hat{x} is the magnitude of B_k for the k -th channel.

Statistical observations have validated the approximate linearity. Given a pre-trained analysis transform (in this experiments, it has been trained for 694000 iterations), I compress the images from CLIC2018 test dataset [85] to collect samples of σ_x^2 and $\sigma_{y_k}^2$. The scatter plot of σ_x^2 and $\sigma_{y_k}^2$ is shown in Fig. 3.17(a), which shows a good energy linearity for analysis transform. Given a pre-trained synthesis transform, (in this study training had been carried out for 336500 iterations), B_k is calculated using Eq.(3.20). Then, I compress the images in the CLIC test dataset [85] to collect samples of $\{\sigma_r^2, \sigma_{q_k}^2\}$. The estimated σ_r^2 is calculated using Eq.(3.13), and the scatter plots of the estimated and ground-truth σ_r^2 are shown in Fig. 3.17(b). It can be observed that the estimated σ_r^2 and ground-truth σ_r^2 demonstrate a type of linearity, although the bias in the convolutional filter affects the offset. Thus, I can assume that CAE still approximately satisfies the linearity constraint of the reconstruction error with quantization errors, as follows:

$$\sigma_r^2 \propto \sum_{k=0}^{K-1} B_k \sigma_{q_k}^2 \quad (3.21)$$

Based on the above analysis, I can formulate the optimum bit allocation problem. Let $\alpha_k = \frac{N_k}{N}$,

where N_k is the number of samples in \mathbf{y}_k and N is the number of samples in \mathbf{x} . According to [75], the optimum bit allocation problem is described next. Under the following constant bit rate constraint:

$$\sum_{k=0}^{K-1} \alpha_k R_k = R(\text{const}) \quad (3.22)$$

, minimize

$$\sigma_r^2 = \sum_{k=0}^{K-1} B_k \sigma_{q_k}^2 \quad (3.23)$$

where R_k is the bit rate for the k -th channel. By substituting the approximating relationship [75]

$$\sigma_{q_k}^2 \simeq \varepsilon^2 2^{-2R_k} \sigma_{y_k}^2 \quad (3.24)$$

where ε is a constant depending on the input images. Using the Lagrangian multiplier method, the minimum value of the reconstruction error variance is expressed as follows:

$$\min\{\sigma_r^2\} = \prod_{k=0}^{K-1} \left(\frac{A_k B_k}{\alpha_k} \right)^{\alpha_k} \cdot \varepsilon^2 2^{-2R} \sigma_x^2 \quad (3.25)$$

To provide a measurement of coding gain, I define a metric as follows. In Eq.(3.25), ε^2 and σ_x^2 are constants for a given image. In our CAE architecture, α_k is a constant because all of the channels have the same size; therefore, I can ignore them. Then, given a constraint R , the coding gain is inversely proportional to σ_r^2 ; that is, the smaller σ_r^2 is, the larger the coding gain becomes. Then, I can define the normalized coding gain as follows:

$$G \propto \frac{1}{(\prod_{k=0}^{K-1} A_k B_k)} \quad (3.26)$$

For simplicity, I focus only on the energy distribution on each channel, not on the absolute value of energy. Therefore, \mathbf{A}_k and \mathbf{B}_k calculated by Eq.(3.19)(3.20) are normalized by dividing their individual sum. Subsequently, \mathbf{A}_k and \mathbf{B}_k become the distribution of energy. So I convert them from the product to the sum with a logarithmic scale. Thus, I define a normalized coding gain as follows:

$$G = -\log_{10} \left(\sum_{k=0}^{K-1} A_k B_k \right) = -\log_{10} (\mathbf{A}_k \cdot \mathbf{B}_k) \quad (3.27)$$

The above analysis proves that the CAE has an energy compaction property that is similar to the linear coding system's equation expressed by Eq.(3.13).

3.4.3 Proposed Energy Compaction-based Bit Allocation Method

Based on the above analysis, the physical meaning of G is to describe the compression capability of neural networks. By realizing a large G , I can achieve the highest reconstruction

Algorithm 3 Proposed Energy Compaction-based Bit Allocation Method

```

for number of training iterations  $I_A$  do
  while  $\max(\mathbf{A}_k) \leq p$  do
    Loss function of CAE becomes as follows:
     $J(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \lambda \cdot D(\mathbf{x}, \hat{\mathbf{x}}) + R(\hat{\mathbf{y}}) + \beta \mathbb{E}[-\log_2 \mathbf{A}_k]$ 
    Update  $\boldsymbol{\theta}, \boldsymbol{\phi}$  by descending its stochastic gradient
  end while
end for

for number of training iterations  $I_B$  do
  Find the channel  $e$  with the largest energy;
  while  $B_e \geq \xi$  do
    Loss function of CAE becomes as follows:
     $J(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \lambda \cdot D(\mathbf{x}, \hat{\mathbf{x}}) + R(\hat{\mathbf{y}}) + \beta B_e$ 
    Update  $\boldsymbol{\theta}, \boldsymbol{\phi}$  by descending its stochastic gradient
  end while
end for

for number of training iterations  $I$  do
  Loss function is  $J(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \lambda \cdot D(\mathbf{x}, \hat{\mathbf{x}}) + R(\hat{\mathbf{y}})$ 
  Update  $\boldsymbol{\theta}, \boldsymbol{\phi}$  by descending its stochastic gradient
end for

```

quality (smallest σ_r^2) with a given rate constraint. \mathbf{A}_k describes the energy distribution of channels in the analysis transform; \mathbf{B}_k measures the extent of the quantization error's impact on the reconstructed error for a specified channel in the synthesis transform. If a large G is desired, $(\mathbf{A}_k \cdot \mathbf{B}_k)$ should be minimized. This implies that the quantization error on a large energy channel (i.e., A_k is large) should have small impact on the final reconstruction quality; that is, B_k should be small for this channel. Moreover, in an almost-all-zero channel (A_k is small), the quantization error can have large influence on the reconstruction quality, i.e., B_k can be either large or small. Thus, I propose to add a regularizer for \mathbf{A}_k and \mathbf{B}_k to the loss function during the training stage. This is the motivation of our energy-compaction based bit allocation method.

First, I need to center the energy in a few channels as much as possible. Because \mathbf{A}_k is already normalized, therefore, \mathbf{A}_k measures the energy distribution for the compressed code \mathbf{y}_k . This means that if $A_e = 0.8$ for the e -th channel, 80% of the energy will be distributed in the e -th channel. Then, I construct a penalty term by using the entropy of the energy distribution as follows:

$$P(\mathbf{A}_k) = \mathbb{E}[-\log_2 \mathbf{A}_k] = \sum_{k=0}^{K-1} -A_k \log_2 A_k \quad (3.28)$$

We add this penalty to the loss function. After several iterations, I find that most of the energy is centered only in one channel, while the other channels have little energy. We denote the channel with the largest energy as e .

Next, I need to minimize B_k for the channel with the largest energy, and change the penalty term as follows:

$$P(\mathbf{B}_k) = B_e \quad (3.29)$$

By minimizing the B_e , I can easily get smaller $(\mathbf{A}_k \cdot \mathbf{B}_k)$, which leads to higher coding gain.

Finally, I also introduce a weight β as a penalty term for flexible adjustment:

$$J(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \lambda \cdot D(\mathbf{x}, \hat{\mathbf{x}}) + R(\hat{\mathbf{y}}) + \beta P(\mathbf{A}_k, \mathbf{B}_k) \quad (3.30)$$

where β controls the influence of the penalty term on the loss function. In our experiments, β was set to 0.001. The proposed energy compaction-based bit allocation method during the training strategy is summarized by Algorithm 3. Empirically, p is about 0.7 and ξ is 0.001.

In our experiments, I_A and I_B have several 10^5 iterations, and I has several 10^6 iterations. I use the penalty terms only during the pre-training, and from the experiments I have observed the following training will not destroy the good energy compaction property. The latent reason is the formulation of energy-compaction property comes from the rate-distortion optimization as Eq.(12)-(15), which is consistent with the Lagrangian multiplier-based rate-distortion loss $J(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \lambda \cdot D(\mathbf{x}, \hat{\mathbf{x}})$.

3.5 Experimental Results

3.5.1 Experimental Results of Image Compression through PCA

We use a subset of the ImageNet database [76] consisting of 5500 images to train the CAE network. In our experiments, H and W are set as 128; therefore, the images that are input to the CAE are split to a size of 128×128 patches. The numbers of filters, i.e. $N_i, i \in [1, 6]$ in convolutional layers are set as $\{32, 32, 64, 64, 64, 32\}$, respectively. The decoder part mirrors the encoder part. The luma component is used to train the CAE network. Mean square error is used in the loss function during the training process in order to measure the distortion between the reconstructed images and original images. For testing, I use the commonly used Kodak lossless

image database [77] with 24 uncompressed 768×512 or 512×768 images. In our CAE training process, λ is set as one and the uniform noise μ is set as $[-\frac{1}{2^{10}}, \frac{1}{2^{10}}]$.

In order to measure the coding efficiency of the proposed CAE-based image compression method, the rate is measured in terms of bit per pixel (bpp). The distortion is measured using metrics PSNR and MS-SSIM [78], which measure the objective quality and perceived quality, respectively.

3.5.1.1 Coding Efficiency Performance

We compare our CAE-based image compression with JPEG and JPEG2000. The color space in this experiment is YUV444. Since the human visual system is more sensitive to the luma component than chroma components, it is common to assign the weights $\frac{6}{8}$, $\frac{1}{8}$, and $\frac{1}{8}$ to the Y, Cb, and Cr components, respectively. The RD curves for the images red door and a girl are shown in Fig. 3.18. The coding efficiency of CAE is better than those of both JPEG2000 and JPEG in terms of PSNR. In terms of MS-SSIM, CAE is better than JPEG and comparable with JPEG2000, because optimizing MSE in CAE training leads to better PSNR characteristic, but not MS-SSIM. Besides, CAE handles a fixed input size of 128×128 ; therefore, block boundary artifacts appear in some images. It is expected that adding perceptual quality matrices into the loss function will improve the MS-SSIM performance, which will be carried out in our future work. Examples of reconstructed patches are shown in Fig. 3.19. We can observe that the subjective quality of the reconstructed images for CAE is better than JPEG and comparable with that of JPEG2000.

The rate-distortion performance can be evaluated quantitatively in terms of the average coding efficiency differences, BD-rate (%) [79]. While calculating the BD-rate, the rate is varied from 0.12bpp to 2.4bpp and the quality is evaluated by using PSNR. With JPEG2000 as the benchmark, the BD-rate results for 24 images in the Kodak database are listed in Fig. 3.20. On average, for the 24 images in the Kodak database, our method achieves 13.7% BD-rate saving compared to JPEG2000.

We also compare our proposed CAE-based method with Balle's work, which released the source code for gray images [54]. For a fair comparison, I give the comparison results for gray images. For Balle's work, the rate is estimated by the entropy of the discrete probability distribution of the quantized vector, which is the lower bound of the rate. In our work, the rate is calculated by the real file size (kb) divided by the resolution of the tested images. Two examples of RD curves are shown in Fig. 3.21. Our method exhibits better RD curves than Balle's work for some test images, such as Fig. 3.21(a), but exhibits slightly worse RD performance for some images, such as Fig. 3.21(b). On average, the performance of our proposed method CAE is

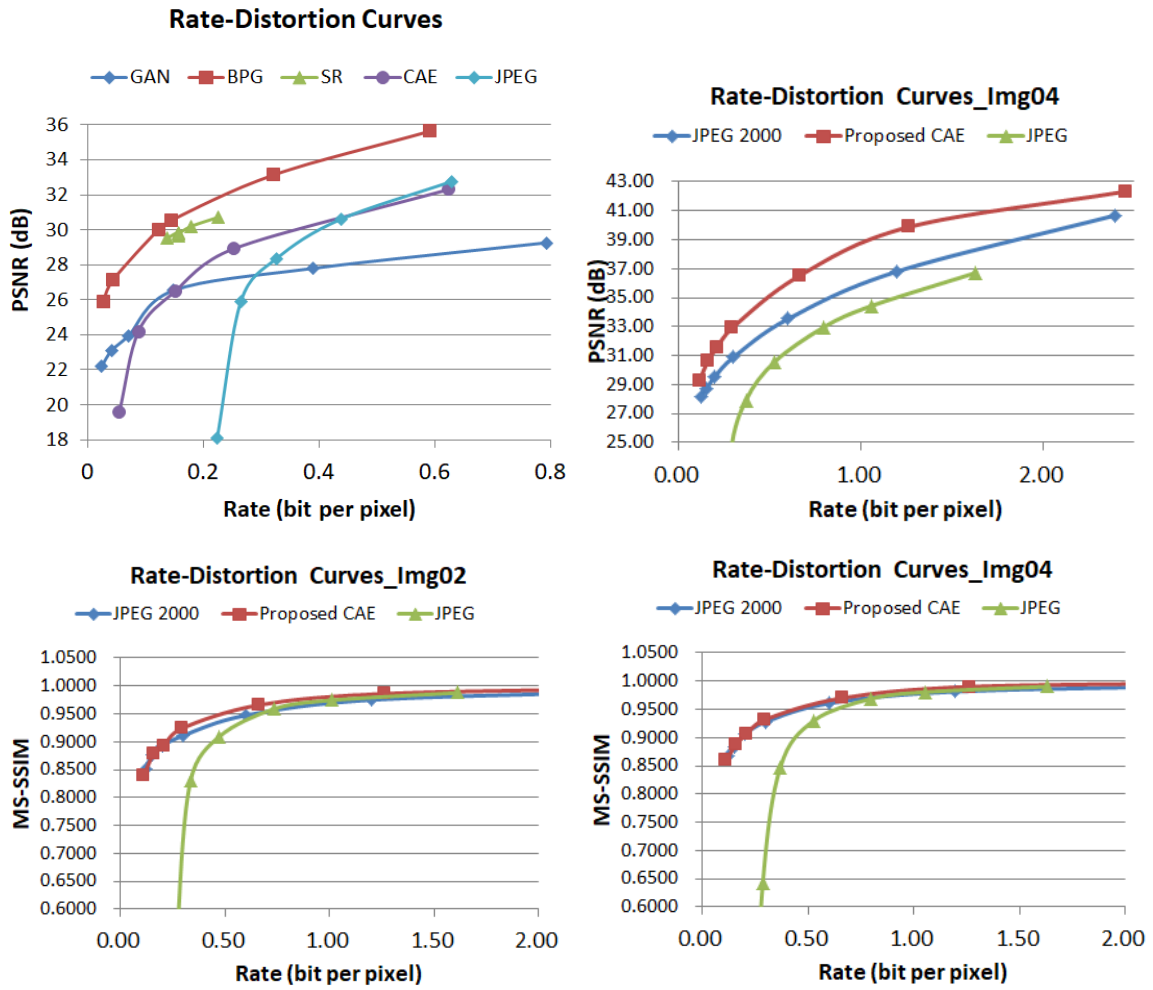


Fig. 3.18 RD curves of color images for the proposed CAE, JPEG, and JPEG2000

comparable with Balle's work, even though the CAE used an actual entropy coder against the ideal entropy of Balle's work.

3.5.1.2 Complexity Performance

The experiments are conducted with Intel Core i7-7700K CPU 4.20 GHz, 16GB RAM and GeForce GTX 1080 GPU. The pre-processing steps for the images and Balle's codec [54] are implemented using Matlab script in Matlab R2016b environment. The codecs of JPEG and JPEG2000 can be found from [82] and [83], implemented with CPU. Balle released only their CPU implementation. Running time refers to one complete encoder and decoder process for one color image with a resolution of 768×512 , while Balle's time refers to the gray image. The running time comparison for each image for different image compression methods is listed in Table 3.16. It can be observed that our CAE-based method achieves lower complexity than

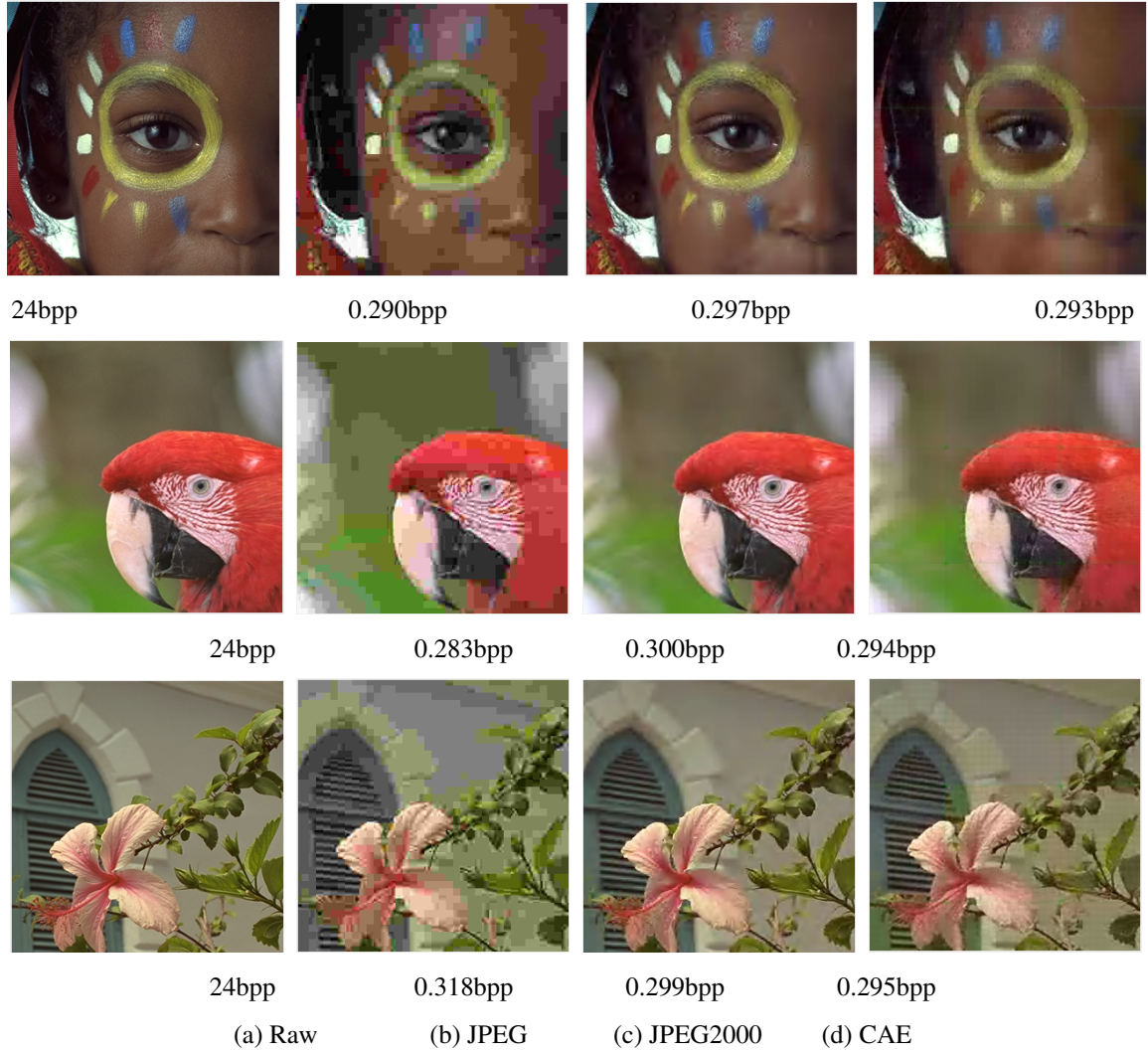


Fig. 3.19 Examples of raw image (a) and reconstructed images (300×300) cropped from Kodak images using (b)JPEG, (c)JPEG2000 and (d)CAE.

Balle's method [54] when it is run by the CPU, because I have designed a relatively simple CAE architecture. Besides, with GPU implementation, our method could achieve comparable complexity with those of JPEG and JPEG2000, which are implemented by C language. Thus, it proves that our method has relatively low complexity.

3.5.2 Experimental Results of Image Compression through Energy Compaction

We used a subset of the ImageNet database [76], which consists of 5,500 images and cropped them into millions of 128×128 samples to train the CAE network. Additionally, H and W were set to 128. The number of filters N was set to 128 except the last one in the encoder and decoder

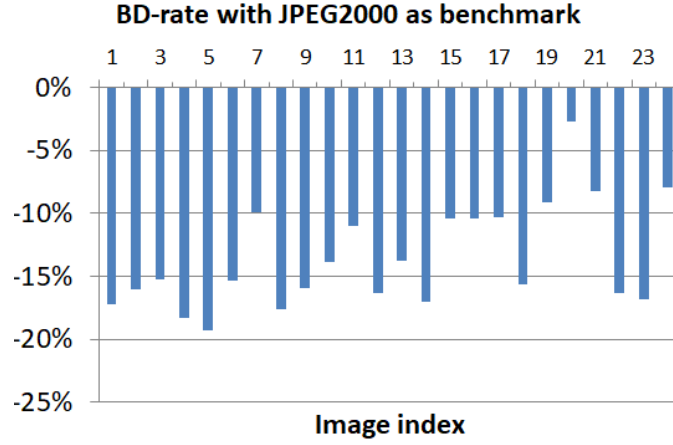


Fig. 3.20 BD-rate of the proposed CAE with JPEG2000 as the benchmark.

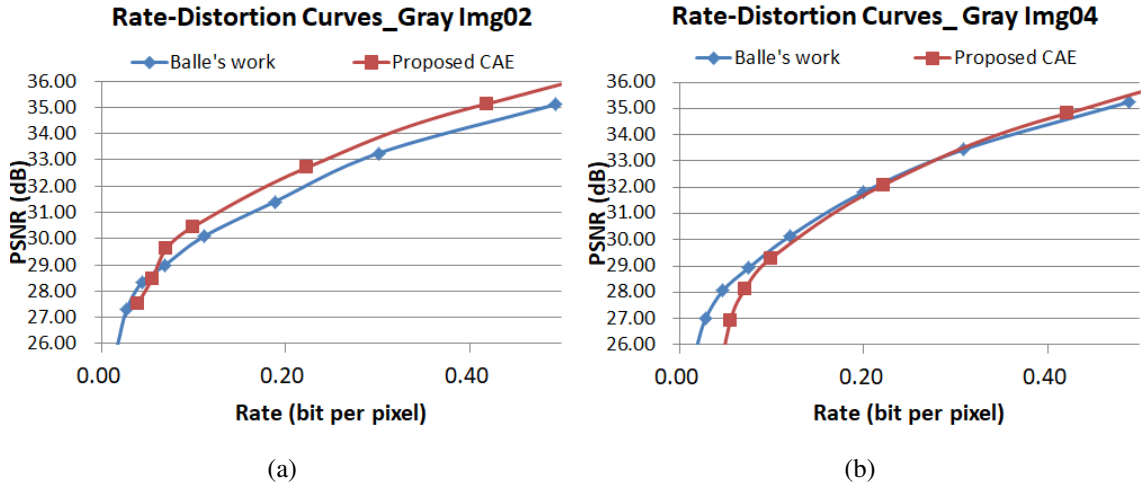


Fig. 3.21 RD curves of gray images for our proposed CAE and Balle's work.

sides. To achieve high subjective quality in the final experiments, I used a popular MS-SSIM [78] as the distortion term defined by $1 - \text{MS-SSIM}(x, \hat{x})$. To train the CAE, the model was optimized using Adam [70] with a batch size of 16. The learning rate was maintained at a fixed value of 1×10^{-4} during the training process. We trained the model up to a few 10^6 iterations for each λ . By introducing different λ to finetune a pre-trained autoencoder, I can obtain variable bit rates. Here, I obtained six models with λ in the set $\{2, 4, 8, 16, 32, 64\}$.

For testing, I used the commonly used Kodak lossless image database [77] with 24 uncompressed 768×512 or 512×768 images. To show the effectiveness of my method, I also tested the proposed method on the CVPR workshop CLIC validation dataset [85] with large resolutions up to about 2048×1370 .

To measure the coding efficiency of the proposed CAE-based image compression method, the

Table 3.16 Average running time comparison.

Codec	Time (s)
JPEG	0.39
JPEG2000	0.59
Balle's work [54] with CPU	7.39
Propose CAE with CPU	2.29
Propose CAE with GPU	0.67

rate is measured in bits per pixel (bpp), and the rate-distortion (RD) curves are drawn to demonstrate their coding efficiency.

3.5.2.1 Network Architecture

To compare the performance of different types of downsampling units, I optimize the models using MSE and set the λ as 0.005 in the loss function for this preliminary experiment. The loss curves of residual unit and plain unit are shown in Fig. 3.22. Fig. 3.22(a) shows the loss of two downsampling units with the architecture of $16 \times 16 \times 48$. The residual unit exhibited an advantage than the plain unit and converges a little faster than the plain unit. The fast convergence of residual unit comes from the shortcut connection to learn the residual mapping easily. Meanwhile, residual unit covers the receptive field as 7×7 , while plain unit can only reach 5×5 receptive field. Large receptive field can contribute to capturing spatial correlation, which leads to better coding performance. After 1×10^6 iterations, the loss almost keeps stable. The rate-distortion performance at 1×10^6 iterations is listed in Table 3.17. The residual unit achieves better results than plain unit, while the number of parameters for the residual unit is about 1.5 times of that for plain unit. This is the tradeoff between model complexity and coding performance. In this paper, considering the model's complexity, I used the plain unit.

Table 3.17 Influence of different types of downsampling units

Variant	PSNR (dB)	MS-SSIM	Rate (bpp)
Plain Unit	33.541	0.981	0.679
Residual Unit	33.872	0.981	0.734

To compare the variants with different number of downsampling units, I conduct the

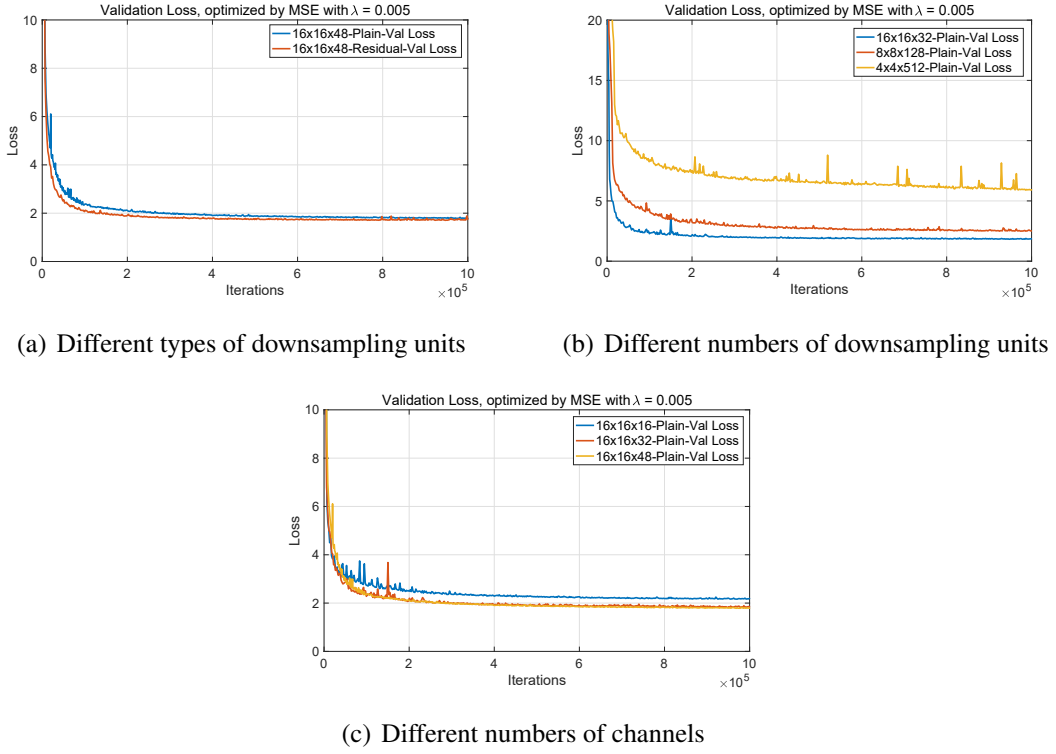


Fig. 3.22 Loss curves of different variants.

experiments with the same experimental setting as the previous part. The loss curves of 3, 4, 5 downsampling units are visualized in Fig. 3.22(b). We changed the number of channels to maintain the amount of compressed codes constant. The rate-distortion performance at 1×10^6 iterations is listed in Table 3.18. It can be observed that $16 \times 16 \times 32$, that is, 3 downsampling unit, achieves the smallest loss among them, because more downsampling units contribute to higher compression ratio, but severely destroyed the reconstructed image quality. Therefore, given an input size of 128, I used three downsampling units.

Table 3.18 Influence of different number of downsampling units

Variant	PSNR (dB)	MS-SSIM	Rate (bpp)
$16 \times 16 \times 32$	33.329	0.980	0.683
$8 \times 8 \times 128$	30.523	0.970	0.503
$4 \times 4 \times 512$	25.969	0.903	0.231

Given three downsampling units for the input size of 128×128 , I conducted experiments with $\lambda = 0.005$ to validate the influence of different number of channels on the coding efficiency. The

loss curves of 16, 32, 48 channels are shown in Fig. 3.22(c). The rate-distortion performance at 1×10^6 iterations is listed in Table 3.19. It can be observed that along with the increasing of number of channels, the loss gets smaller. When the number of channels increases to 48, the model almost saturated. We select 48 channels to have a high capacity.

Table 3.19 Influence of different number of channels

Variant	PSNR (dB)	MS-SSIM	Rate (bpp)
$16 \times 16 \times 48$	33.541	0.981	0.679
$16 \times 16 \times 32$	33.329	0.980	0.683
$16 \times 16 \times 16$	31.477	0.976	0.513

Several differentiable quantization approximations are present. One intuitive approach consists of approximating the quantization error. Subsequently, the error can be modeled as uniform noise. Assuming that the quantization errors are not correlated with each other, Ballé et al. [54] replaced the quantization by an additive uniform noise u as $\tilde{y} \approx y + \mu$, where $u \in U(-\frac{1}{2}, \frac{1}{2})$. The other method is the soft vector quantization reported in [58]. Let us assume that I have L centers vectors $C = \{c_1, \dots, c_L\}$. Then, the soft assignment of y to C is expressed as $\phi(y) = \text{softmax}(-\sigma(\|y - c_1\|^2, \dots, \|y - c_L\|^2))$. Then, the soft quantization is defined as $\hat{y} = \sum_{j=1}^L c_j \phi(y)$. The comparisons between the round-based quantization, additive uniform noise, and soft vector quantization are shown in Fig. 3.23. Uniform noises generate random errors. Soft vector quantization is more accurate when σ is large, although a small σ is friendly to gradient backpropagation. By conducting experiments, I found that the different quantization methods do not affect the compression performance.

3.5.2.2 Different Bit Allocation Methods

To ensure a fair comparison, I used the same network architecture with $n = 3, K = 48$ and the same training dataset to investigate the performance of different bit allocation methods. Before that, I will briefly introduce two related studies for comparison.

(1) Principle Component Analysis (PCA) based Bit Allocation [39]

To decorrelate each channel, principle component analysis (PCA) was used to obtain a more energy-compact representation. y represents the compressed data with a dimension of $[\frac{H}{2^n}, \frac{W}{2^n}, K]$, and that y is reshaped as the 2-dimensional data $[\frac{H}{2^n} \times \frac{W}{2^n}, K]$. PCA is applied to each channel by executing the following steps:

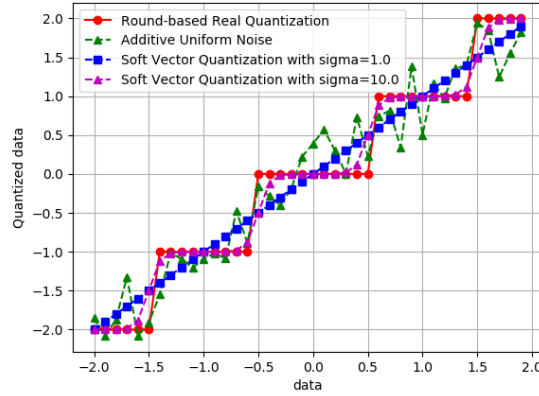


Fig. 3.23 Visualization of different quantization methods.

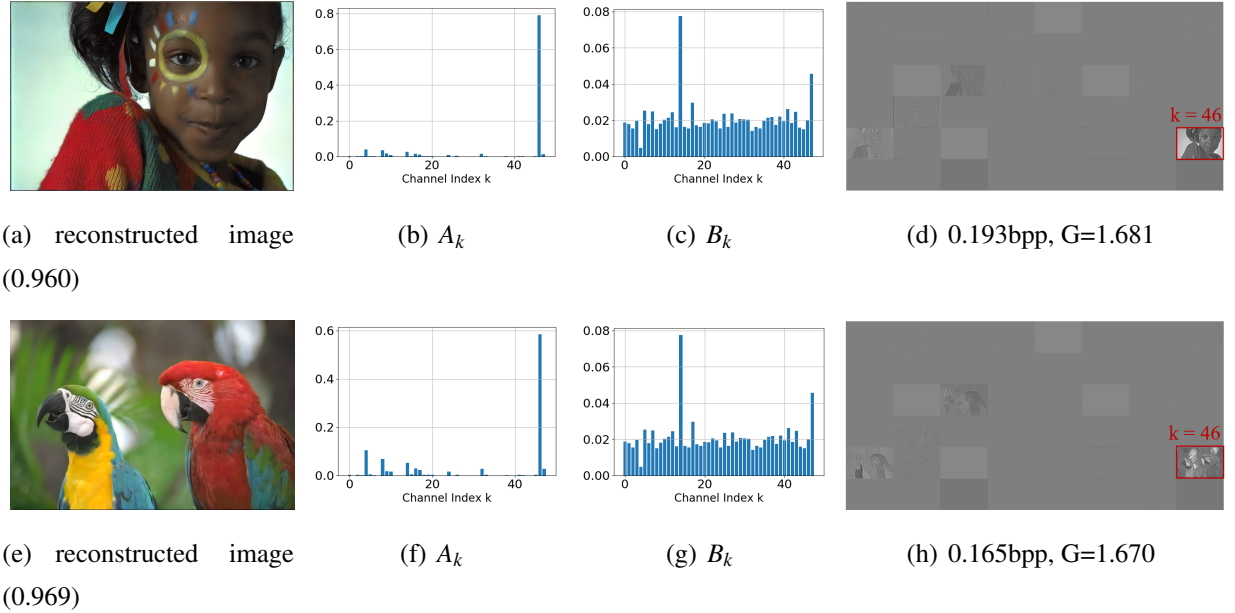


Fig. 3.24 Visualization Examples for Baseline.

- Compute the covariance matrix as $\Sigma = \frac{1}{m} \sum_1^m (y)(y)^T$, where $m = \frac{H}{2^n} \times \frac{W}{2^n}$.
- Compute the eigenvectors of Σ and stack the eigenvectors in columns to form the matrix U in descending order.
- Reduce the data by computing $y_{pca} = U^T y$.

At the decoder side, the compressed codes can be recovered by $y' = U y_{pca}$, where U has the dimension of $[K, K]$ and must be transmitted to the decoder as side information.

(2) Importance Map based Bit Allocation [59]

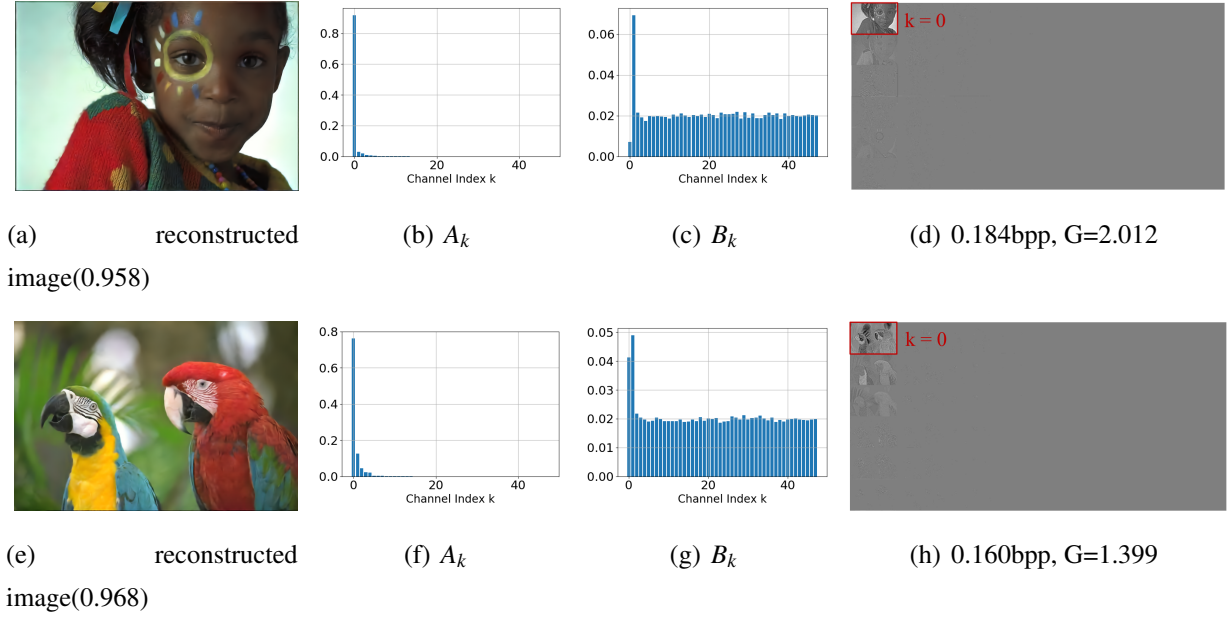


Fig. 3.25 Visualization Examples for PCA based bit allocation.

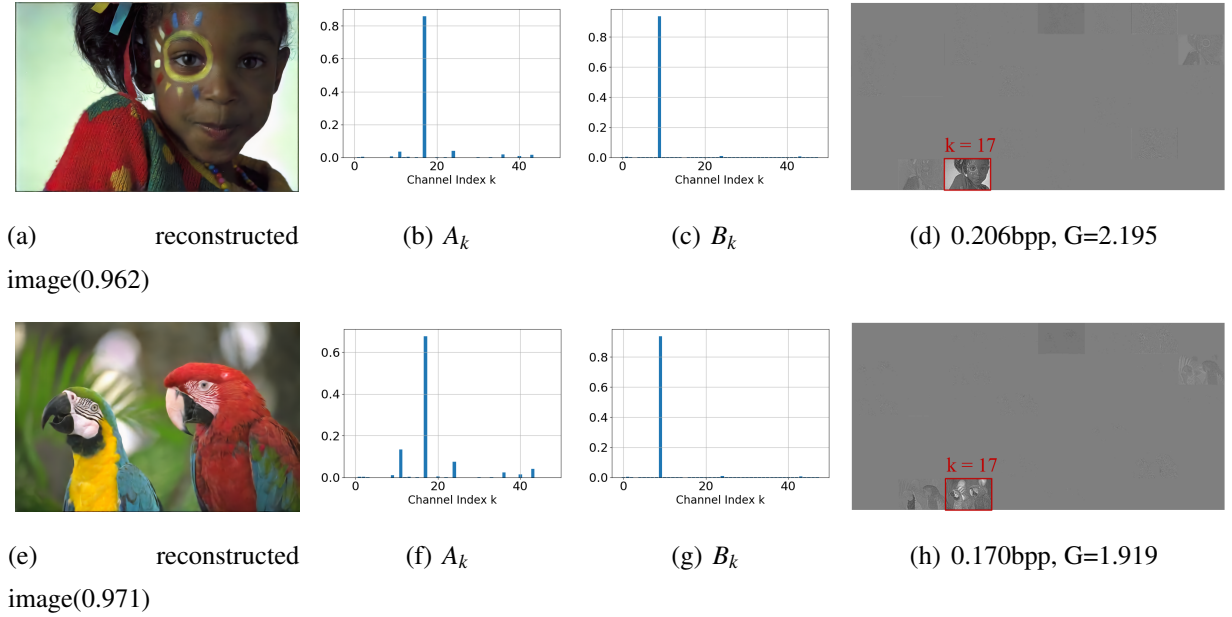


Fig. 3.26 Visualization Examples for MAP based bit allocation.

An importance map was added as a spatial bit allocation method. The last layer of the encoder was taken to add the second single-channel output $m \in \mathbb{R}^{\frac{H}{2^n} \times \frac{W}{2^n} \times 1}$. This single channel m was further expanded into a mask $M \in \mathbb{R}^{\frac{H}{2^n} \times \frac{W}{2^n} \times K}$ with the same dimensionality as y , and the value of the mask was clipped into the range of $[0, 1]$ as follows:

$$M_{i,j,k} = \text{clip}(m_{i,j} - k, 0, 1) \quad (3.31)$$

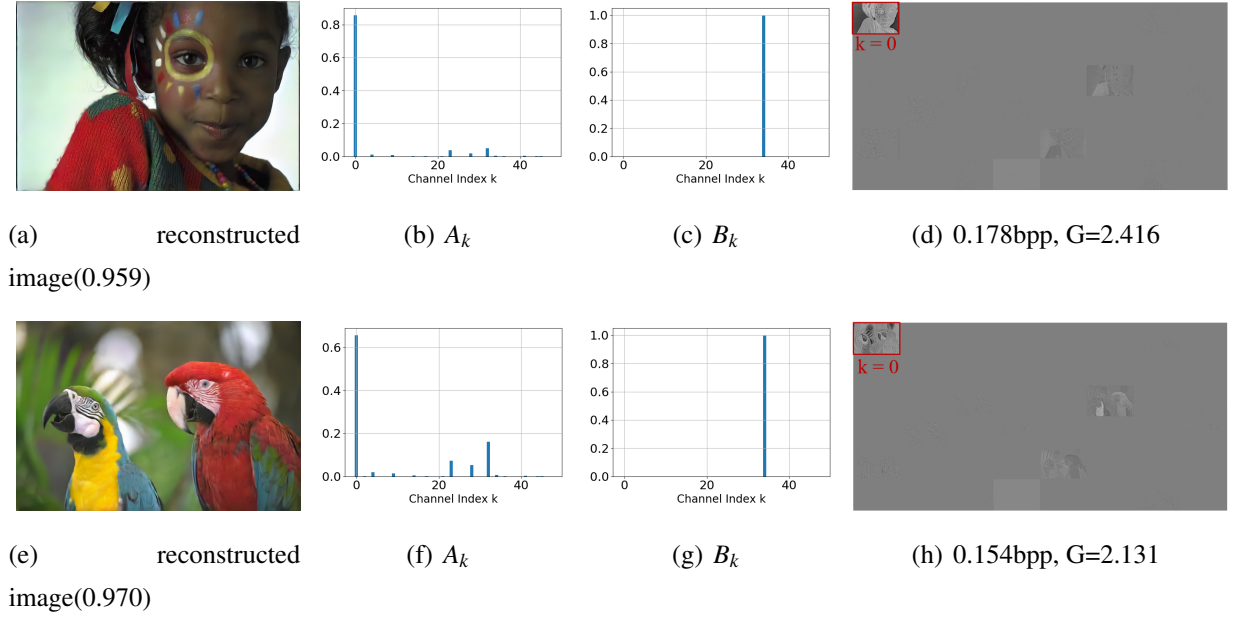


Fig. 3.27 Visualization Examples for Our proposed energy compaction based bit allocation.

The mask inherently makes the first several channels have larger energy than the last few channels. As the channels increase, the mask decreases. Then, y is masked by pointwise multiplication (i.e., $y = y \odot M$).

We compare four cases, i.e., without any bit allocation (we denote it as baseline), principle component analysis (PCA), importance map (MAP), and our proposed energy compaction-based bit allocation method.

First, I present A_k and B_k without any bit allocation methods for the baseline shown in Fig. 3.24. The first column shows the reconstructed images, and the one below is the corresponding MS-SSIM value. The second column shows the normalized A_k for 48 channels, which measures how much energy is allocated in each channel. The third column is the normalized B_k calculated by faking codes such that they could pass through the synthesis transform. Thus, B_k was kept the same for all of the input images. The fourth column is the visualization of y by stacking each channel in a vertical-scan order and scaling to the range of $[0, 255]$ with a mean of 127.5. The one below is the rate required for the quantized \hat{y} and the coding gain metric G . It can be observed that the 46-th channel achieved the largest energy according to A_k , and auto-learned through the CAE training. There was no constraint for B_k ; therefore, B_k was normally distributed on each channel. The coding gain G was approximately 1.681 or 1.670.

For a PCA-based bit allocation method, A_k, B_k , and the reconstructed images are presented in Fig. 3.25. The computation of y_{pca} can ensure that the first feature maps have the largest value,

and that the features maps are sorted in descending order. Moreover, it can be observed that a large amount of energy was centered at the top-left corner of the rotated channels, and that A_k was arranged well in descending order. B_k can be determined both by U and the original decoder filter coefficient. U is related to the covariance matrix of y ; therefore, B_k was different for the different texture characteristics of the raw images. For some images, such as those shown in Fig. 3.25(a), the coding gain was largely improved to approximately 2.012. However, as shown in Fig. 3.25(e), the coding gain decreased to 1.399.

The results of the MAP-based bit allocation method are shown in Fig. 3.26, where it can be observed that the last few channels already had no energy left. Therefore, B_k could be learned as being distributed in the first several channels, but not in all channels. The coding gain was improved up to 2.195. However, I determined experimentally that the map for the first 40 channels almost became equal to one; therefore, it could not shape the energy as I had desired. The largest amount of energy was located in the 17-th channel.

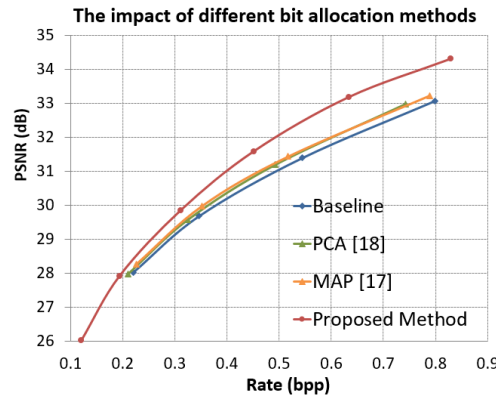


Fig. 3.28 Comparison between different bit allocation methods.

Table 3.20 Coding gain averaged on Kodak.

Method	Baseline	PCA [39]	MAP [59]	Proposed
Gain	1.703	1.714	1.823	2.230

The results of our proposed energy compaction-based bit allocation method are presented in Fig. 3.27, where both A_k and B_k exhibited significant sparsity, and particularly for B_k , the B_k for almost all of the channels is approximately zero, which implies that these channels have the ability to recover from quantization noises. Only for the 34-th channel, $B_k[34]$ was close to 1, which implies that this channel had significant impact on the reconstruction quality. This may be related

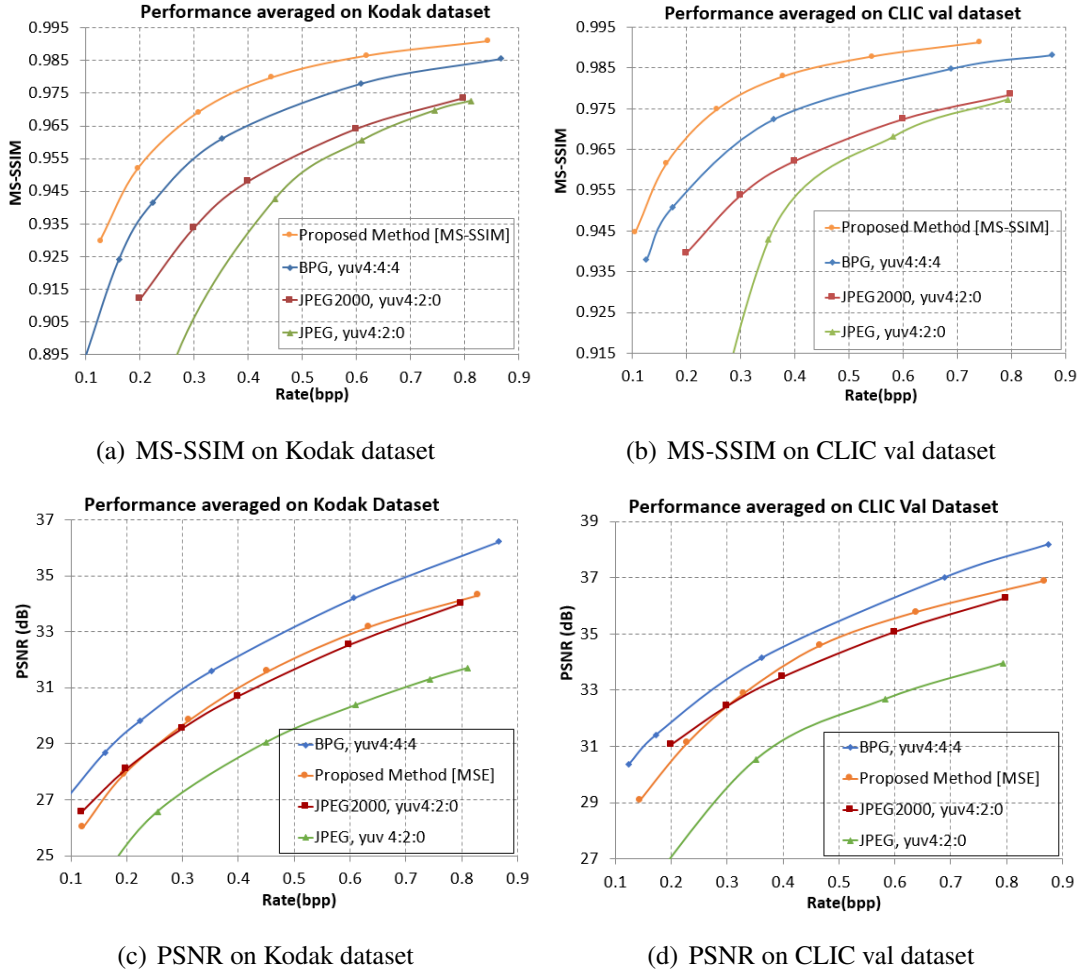


Fig. 3.29 Performance comparison with compression standards in terms of PSNR.

to the mean value of the reconstructed images. G was greatly improved up to 2.416.

The corresponding RD performances averaged on the Kodak dataset with different bit allocation methods are presented in Fig. 3.28, where MS-SSIM is converted to decibels ($-10\log_{10}(1-\text{MS-SSIM})$) to illustrate the difference clearly. The averaged coding gain for 24 images in the Kodak dataset is presented in Table 3.20. G describes the energy compaction property of each model. It can be observed the coding gain of our proposed method had the highest value of among all of the bit allocation methods, and the RD curves also demonstrate that our method achieved the best coding efficiency. To the best of our knowledge, our method is the first work to give a quality metric to measure the energy compaction property of neural networks, so it can help unsupervised autoencoder to train more efficiently.

3.5.2.3 Performance Comparison

In this section, I describe the comparison of the proposed method with related studies. First, I compared our method with well-known compression standards. For JPEG and JPEG2000, I used the official software libjpeg [82] and OpenJPEG [83], which uses the best default configuration YUV420 format. The state-of-the-art image compression standard was BPG [46], for which I used the non-default YUV444 format refer to [59] [66]. Fig. 3.29 shows the comparison between JPEG, JPEG2000, and BPG averaged on Kodak and the CLIC validation dataset. My method is comparable with JPEG2000 on PSNR. My method also outperforms JPEG, JPEG2000, and BPG significantly in terms of MS-SSIM.

Moreover, I have compared the performance of our method with that of recent neural network-based learned compression methods. Because the source codes of previous neural networks based approaches were not available, I carefully traced the points in the RD curves from the respective studies of Nick [64], Theis [53], Ballé [54], and Ripple [66]. The data in Mentzer [59] were obtained from their project page <https://github.com/fab-jul/imgcomp-cvpr>. It can be observed that our method significantly outperforms Nick [64], Theis [53], Ballé [54]. Furthermore, our methodology is better than the work of Mentzer [59] and Ripple [66] at high bit rate, because our proposed bit allocation method can allocate bits more efficiently, with higher bit budgets. Our method achieves comparable performance with Mentzer's work [59] and Ripple [66] at low bit rate. Currently I only use a factorized entropy model and our method does not depend on the design of entropy models. Thus, in future work, our bit allocation method can be combined with a more complicated context adaptive entropy model, such as [55], to yield a better result at low bit rates.

3.5.2.4 Visualization

We have visualized some reconstructed images as examples to demonstrate the subjective quality performance in Fig. 3.31 and Fig. 3.32. Fig. 3.31 shows examples kodim01 under approximately 0.24 bpp with a compression ratio of 100:1, because the raw images are a lossless PNG format with 24 bpp (8 bit for each color component). Our method denotes the model optimized by MS-SSIM, because MS-SSIM can represent subjective quality better than PSNR. Thus, it is observed that the latch on the door is maintained well in our reconstructed images. However, the images are blurry for the BPG, JPEG2000 and JPEG reconstructed images. Fig. 3.32 shows examples kodim21 with approximately 0.12 bpp and a compression ratio of 200:1. It can be observed that the cloud above the sea appear more natural in our reconstructed images using 0.115 bpp less bits than BPG,

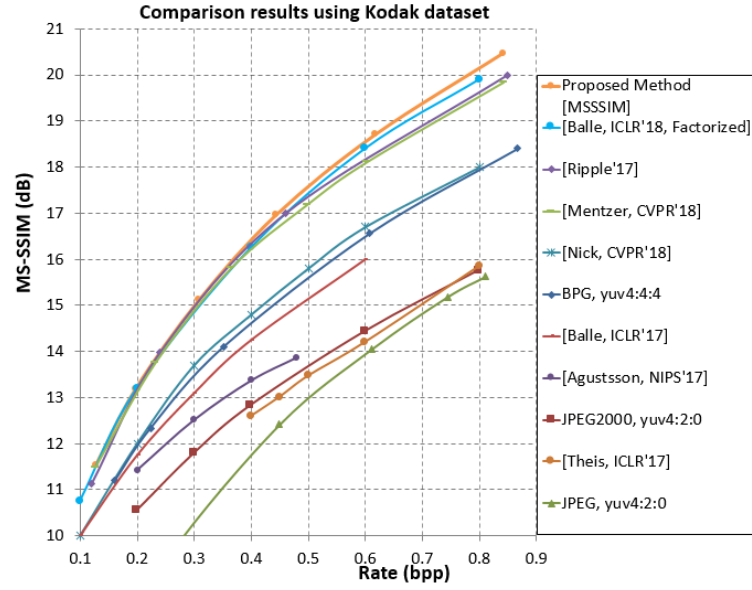


Fig. 3.30 Comparison with related studies using Kodak dataset.



Fig. 3.31 Example of one reconstruction image (kodim01) with approximately 100:1 compression ratio from Kodak dataset.

JPEG2000, and JPEG. Particularly, for the BPG-encoded images, blocking artifacts occur in the sky when a large compression ratio is applied.

Besides, ringing artifacts often occur in classical image compression standards. To check whether ringing artifacts exist in deep learning based methods, I show the zoom-in sub-image from Kodim19 in Fig. 3.33. Obvious ringing artifacts can be seen for JPEG and JPEG2000

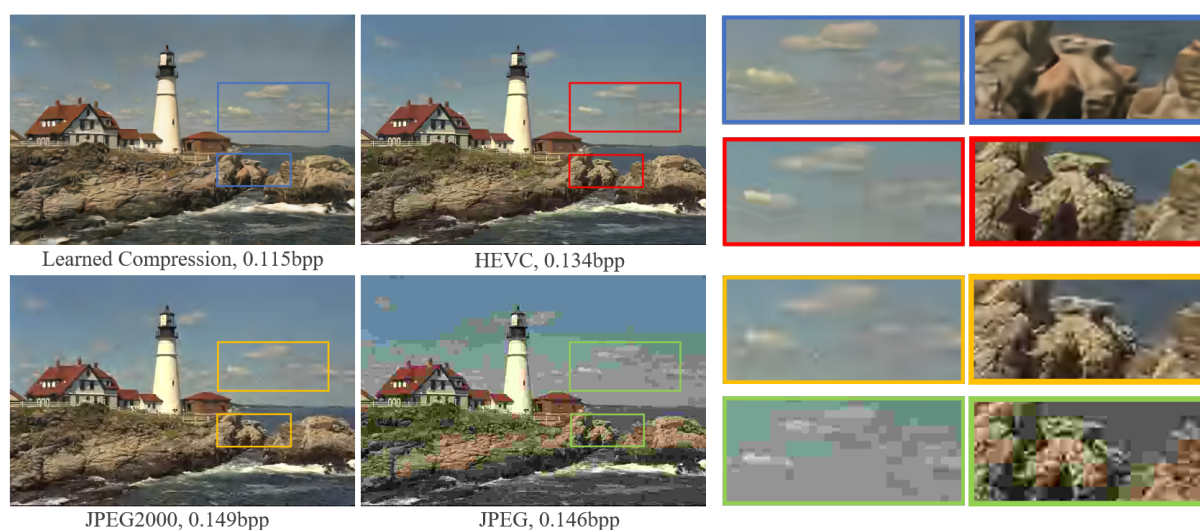


Fig. 3.32 Example of one reconstruction image (kodim21) with an approximate compression ratio of 200:1 from Kodak dataset.



Fig. 3.33 Visualization of ringing artifacts in compressed images with comparable bitrate about 0.6bpp.

compressed images. For BPG compressed image, only a little ringing artifact can be seen at the top of tower. Ringing artifacts are due to loss of high frequency components, transform by DCT or DWT. Classical image compression standards usually quantize the high-frequency coefficients to zero for compression, resulting to visible ringing artifacts. However, for deep learning based

image compression approaches, I did not observe the ringing effect for both models optimized by MSE or MS-SSIM. Interestingly, I think deep learning-based approaches brings new type of artifacts, such as deformation.

3.6 Perceptual Quality Study on Learning based Image Compression

Most techniques only report their performance in terms of objective quality metrics, e.g. PSNR or MS-SSIM [78]. Very few, however, have put emphasis on evaluating their performance based on subjective quality assessments. An example is the work in [61] which relies on small images (736×960) not necessarily fit for the current trend in high-resolution imaging applications. A particularly important observation is that learned compression brings new types of artifacts that differ from blocking or ringing artifacts created by traditional codecs, as illustrated in Fig. 3.32. One observes that shape of clouds in the above illustration tend to be well-preserved in learned compression while clear artifacts can be seen in traditional coding approaches. However, the rock distorted by learned compression looks unnatural while the rock reconstructed by traditional codecs look more realistic. The impact of artifacts produced by learned image compression on human perception are still unknown. Therefore, a study on perceptual quality of learned image compression is essential in order to achieve further progress in this direction.

Our contributions in this paper are two-fold. First, I carefully design a generic learned image compression approach. Different objective quality losses are used to optimize our models as in many prior efforts in the state of the art. Second, I conduct subjective quality assessments to compare the performance of six representative compression algorithms. Subjective quality evaluation results demonstrate that the learned compression algorithm optimized by MS-SSIM yields competitive results to reconstruct visually pleasant images. More importantly, I gain valuable insights on the future developments for learned compression.

3.6.1 Codec Architecture

We build a general learned compression based on a convolutional autoencoder (CAE) [39], shown in Fig. 4.1. The analysis and synthesis transforms can be decomposed into downsampling and upsampling units, where a downsampling unit is composed of two convolution filters. The upsampling unit has the same structure with the convolution filters replaced by deconvolution filters. Each filter has a kernel size of 3 and 128 output channels. The latent representation y has

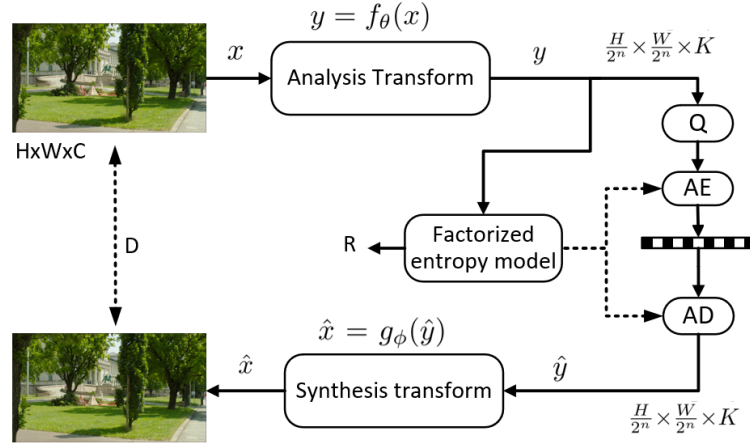


Fig. 3.34 Proposed learned image compression approach.

the dimension of $\frac{H}{2^n} \times \frac{W}{2^n} \times K$ where the n is the number of down(up)sampling units and K is the number of channels before quantization. In our experiments, I set $n = 3$, $K = 48$. We use factorized entropy model [55], which is proved to be efficient to model any arbitrary distribution. It produces a context model and generates an estimated entropy to serve for AE and AD. For testing, I use the JPEG 2000 entropy coder to generate compressed bitstreams.

The loss function is defined similar to rate-distortion optimization (RDO) in traditional codecs, defined by

$$J(\theta, \phi; x) = \lambda D(x, \hat{x}) + R(\hat{y}) \quad (3.32)$$

where R represents the number of bits to encode the quantized compressed data \hat{y} . θ and ϕ are optimized parameters at the encoder and decoder sides. D represents the distortion between original x and reconstructed image \hat{x} , and can be estimated by any objective quality metrics. The reconstruction quality of learned compression heavily relies on the quality metrics in the loss function. The two most popular ones are

$$D(x, \hat{x}) = (1 - \text{MS-SSIM}(x, \hat{x})) \quad (3.33)$$

or

$$D(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2 \quad (3.34)$$

The model was optimized using Adam [70] with a batch size of 16 up to 10^6 iterations. The learning rate was set at a fixed value of 1×10^{-4} . We have tested models trained using MSE and MS-SSIM loss to investigate the effect of quality metrics.

In order to cope with high-resolution images, they are split into tiles and each coded individually. After decoding, tiles are either stitched together as they do not overlap, or combined

by weighted averaging of their boundary regions that overlap. We used a 32 pixel overlap as a compromise between redundancy and reduction of blocking artifacts between tiles. Fig. 3.35 depicts the performance of the proposed autoencoder codec in terms of MS-SSIM when compared to other state-of-the-art codecs on Kodak dataset.

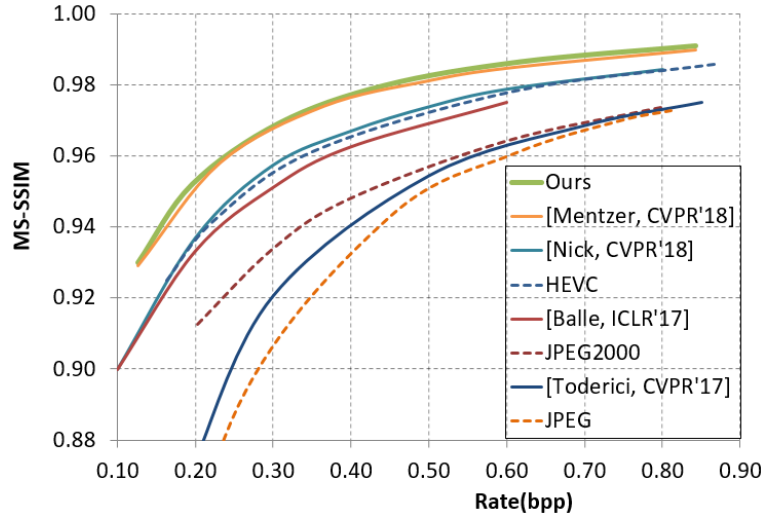


Fig. 3.35 Performance of recent works on Kodak dataset.

The most recent JPEG XL call for proposals [80] was used for anchor generation and resulted in a total of 6 codecs to be considered as in Table 3.21.

Table 3.21 Codecs considered in this paper.

Codec	Specification and reference software
CAE-MSE-ov	CAE optimized by MSE, stitched in an overlapped manner
CAE-MS-SSIM-nonov	CAE optimized by MS-SSIM, stitched in a non-overlapped manner
CAE-MS-SSIM-ov	CAE optimized by MS-SSIM, stitched in an overlapped manner
HEVC/H.265	ISO/IEC 23008-2 ITU-T Rec. H.265, Software: HM16.18+SCM-8.7, Intra [81]
JPEGXT	ISO/IEC 18477, Software: JPEG XT v1.53 [88]
JPEG2000	ISO/IEC 15444-1 ITU-T Rec. T.800, Software: Kakadu v7.10.2 [89]

3.6.2 Subjective Quality Evaluations

3.6.2.1 Dataset

For this study, I selected 7 uncompressed 8-bit RGB test images in high-resolutions (HD to UHD), following the latest JPEG XL Call for Proposals [80], as shown in Fig. 3.36. The codecs

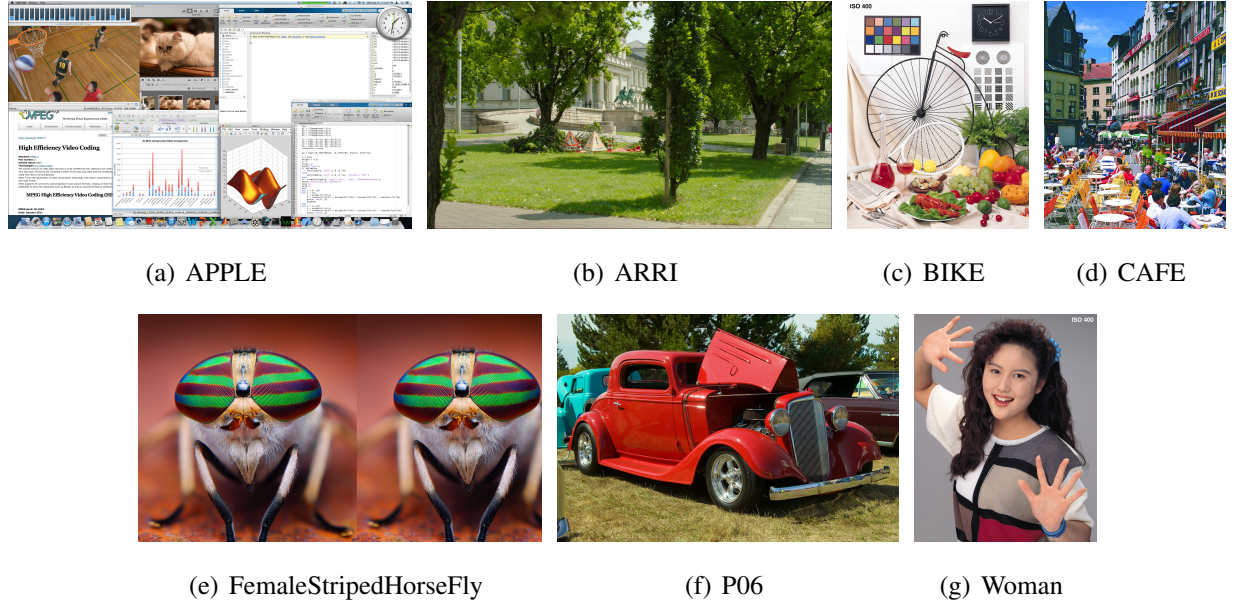


Fig. 3.36 Test images in this study.

were evaluated on four target bitrates R_1 to R_4 , corresponding to very low to high bitrates, which were determined during expert sessions as described in [80]. For image compression standards, I selected quantization parameters (QP) to match the target bitrates. For our proposed learned compression scheme, I adjust λ in Eq.(3.32) to train models and achieve target bitrates.

3.6.2.2 Test Methodology

The methodology is based on Absolute Category Rating with Hidden Reference (ACR-HR) [90] where only one image is displayed in the center of the screen at a time. Participants were required to rate the visual quality based on a five-level scale, i.e., Excellent (5), Good (4), Fair (3), Poor (2), Bad (1). The whole evaluation consisted of 168 stimuli, namely 6 codecs, 7 contents and 4 bitrates. The display order was randomized so that the same content was never displayed consecutively. The test was split into two sessions to avoid subjects fatigue. Prior to testing, a training was performed to familiarize subjects with the typical artifacts and the rating scale.

To avoid the involuntary influence of external factors and to ensure the reproducibility of results, a controlled environment with mid-gray background for subjective quality assessment was preferred according to [91]. An Eizo ColorEdge CG318-4K monitor with native resolution of 4096×2160 pixels was used for tests. The background of the display was set to mid grey [92]. The display brightness was calibrated at 120 cd/m² and background illumination was set to 15 lux. A total of 16 participants (10 males and 6 females) took part in experiments, with an age

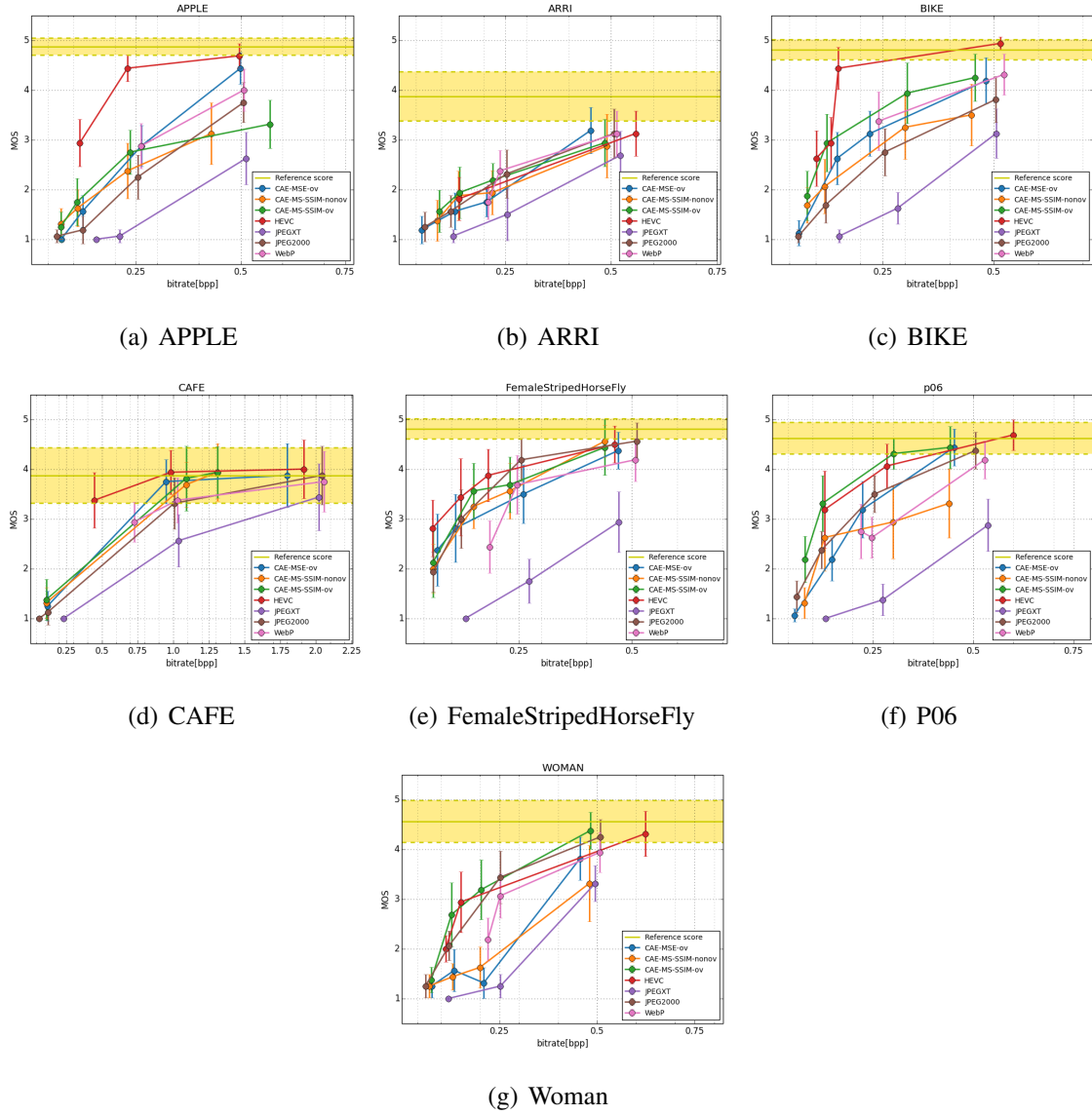


Fig. 3.37 Results of MOS vs. bitrate with corresponding confidence interval.

between 19 and 38 years old, with an average of 26.4 and a median of 27.

3.6.3 Results and Discussion

For the evaluation of perceived quality, outlier detection was performed using the approach in [91] and no outlier was detected. The mean opinion score (MOS) was computed for each stimulus as

$$MOS_j = \frac{1}{N} \sum_{i=1}^N m_{ij} \quad (3.35)$$

where m_{ij} is the score for stimulus j given by subject i and N is the total number of participants. 95% confidence intervals (CIs) were computed assuming a Student's t -distribution of the scores.

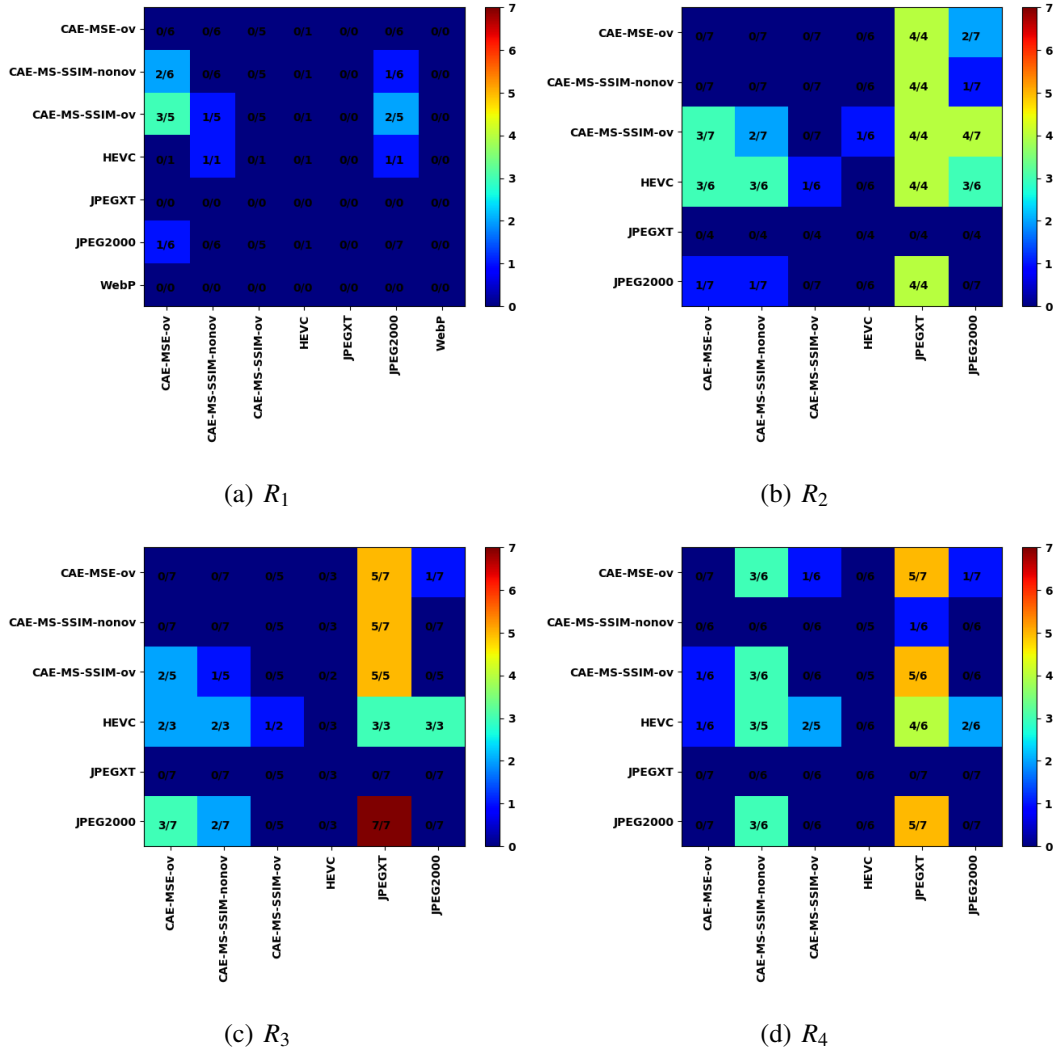


Fig. 3.38 Pairwise Comparison for each bitrate, where for “n/m” in each cell, m denotes how many contents are comparable for each pairwise comparison, and n denotes how many times the codec on y-axis outperforms the codec on x-axis.

Fig. 3.37 shows the MOS vs. bitrate curves for 4 typical contents. MOS of the hidden reference with corresponding CI are depicted with a yellow stripe, whereas all the codecs are plotted with a solid line. To determine whether the results yield statistical significance, a two-sided Welch test at 5% significance level was performed on the scores.

Fig. 3.38 shows how many contents the codec on the y-axis performs significantly better than the codec on the x-axis for each bitrate. The minimum value is 0 and the maximum value is 7, corresponding to the total number of test contents. Some of the codecs could not reach all target bitrates (especially the lowest) within reasonable deviation due to limited and integer quantization parameter. To ensure fair comparison, I excluded the lowest bitrates and conducted the pairwise

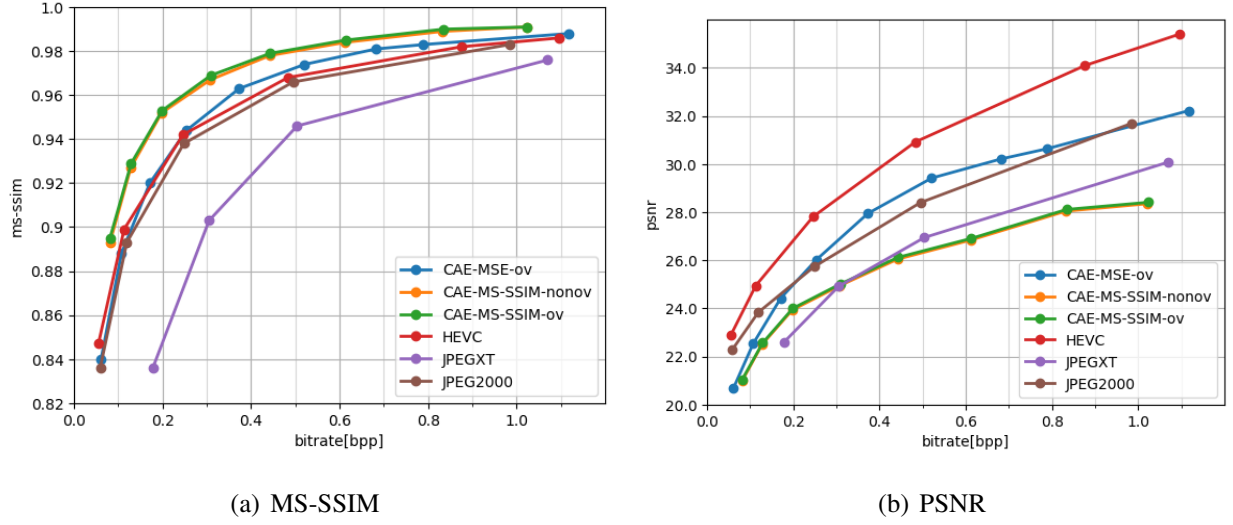


Fig. 3.39 Performance with respect to MS-SSIM and PSNR.

comparison only when the bitrate difference between actual bitrate and target bitrate was less than a predefined threshold specific to each targeted bitrate.

Fig. 3.38 shows that HEVC/H.265-intra outperforms all codecs for all bitrates and all contents except for content Woman at R_2 , where CAE-MS-SSIM-ov outperforms HEVC/H.265-intra. This implies a better reconstruction quality for learned image compression approach for face contents. This result is expected as optimizing MS-SSIM leads to better structural similarity and human visual system is sensitive to human faces. Overall, CAE-MS-SSIM-ov achieves the second best performance, as it is statistically comparable with HEVC/H.265-intra at R_2 , but a little worse at R_3 and R_4 .

I can also conclude some useful information for learned image compression. First, comparing CAE-MS-SSIM-ov and CAE-MS-SSIM-nonov indicates the overlapping stitching strategy achieves superior performance compared to the non-overlapping case. More interestingly, CAE-MS-SSIM-ov outperforms CAE-MSE-ov on 3, 2 and 1 out of 7 contents at R_2 , R_3 and R_4 , respectively. On the contrary, CAE-MSE-ov outperforms CAE-MS-SSIM-ov only on 1 out of 6 contents at R_4 . We have observed that to achieve higher subjective quality at low bitrates, MS-SSIM works better than MSE, while at high bitrates, MS-SSIM and MSE do not show significant differences.

Furthermore, I calculate the PSNR and MS-SSIM results to illustrate the difference between subjective and objective quality evaluations. To obtain a fair comparison, I use averaged PSNR on RGB components, defined by

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2 \times 3}{MSE_R + MSE_G + MSE_B} \right) \quad (3.36)$$

The MS-SSIM calculation refers to [78]. Fig. 3.39 shows the averaged results on 7 contents. It can be observed that CAE-MS-SSIM-ov achieves the best MS-SSIM performance, which outperforms HEVC/H.265-intra significantly. In terms of PSNR, CAE-MSE is comparable to JPEG 2000, but worse than HEVC/H.265-intra. Both results correlate poorly with subjective quality evaluation results, which illustrates the need for a better perceptual similarity metric to improve the performance of learned based compression.

3.7 Chapter Summary

In this chapter, I mainly discuss the deep learning based image compression. The main contribution includes several aspects.

First, I fully discuss the architecture selection in Section 3.2. I compare the performance of CAE, GAN, and super resolution for image compression. The results are submitted to CLIC 2018. Besides, I propose the deep residual learning architecture to increase the receptive field and submit it to CLIC 2019 to achieve 5-th rank among all the submission teams.

Second, I design a novel CAE structure with multiple downsampling and upsampling units to generate feature maps with low dimensions. We optimize this CAE using an approximated rate-distortion loss function. To generate a more energy-compact representation, I further propose a principal components analysis (PCA)-based rotation to generate more zeros in the feature maps. Experimental results show my method achieves superior performance than JPEG and JPEG2000 in terms of PSNR and achieves a 13.7% BD-rate decrement compared to JPEG2000 with the popular Kodak database images. In addition, our method is computationally more appealing compared to other autoencoder based image compression methods.

Third, I proposed an energy compaction-based image compression architecture using a CAE. Based on the CAE architecture, I provided a mathematical analysis with regard to the energy compaction property based on CAE to define a normalized coding gain, which is a measure of compression capability. Thirdly, based on the above analysis, I propose a regularizer and add it into a loss function to train the CAE to achieve a higher coding gain. The experimental results revealed that our method outperforms the image compression standard HEVC-intra in terms of MS-SSIM quality metric. Furthermore, I achieved better performance than existing learning based compression methods at high bit rate. This results has been accepted by IEEE Transactions on Multimedia.

Last, a thorough and rigorous perceptual quality study on different compression algorithms is conducted based on my proposed approach. A total of six compression algorithms were involved

in subjective quality assessment tests and high resolution images were selected carefully, in line with state-of-the-art codec comparisons. We then conducted the subjective tests in a controlled environment by adopting an ACR-HR methodology. Results demonstrate learned compression optimized by MS-SSIM achieved competitive results with state-of-the-art codecs and the advantages of optimization with respect to MS-SSIM at low bitrates when compared to PSNR for learned image compression.

Chapter 4

Learning-based Video Compression through Interpolation Network

In this chapter, a learning based video compression through spatial-temporal energy compaction is proposed. This proposal is an extension of spatial energy compaction in Chapter 3, by extending image compression to video compression. I generalize the learning-based video compression, by considering an interpolation loop and adaptive interpolation period selection based on entropy of temporal information. The residual information after interpolation is also transmitted to decoder side. This mechanism is similar to Bi-directional prediction frames (B-frames). The GoP size can be adaptively selected according to the motion characteristics of input sequences. Experimental results demonstrate that the proposed video compression approach outperform MPEG-4, and is competitive with commonly used H.264. The proposed video compression can achieve better visual quality than traditional standards.*

4.1 Related Work

Video Compression is a very important research topics for both academics and industries. According to Cisco Report [95], video content has consumed 70% of all the internet traffic in 2016 and is expected to grow threefold by 2021. Video compression will have a significant impact on the delivery and transmission of every video content. Through many efforts spanning several decades, many standards have been established. Similar to image compression, video compression standards are well engineered and compatible with many hardware and devices. However, along with the proliferation of high-resolution videos, as well as the development of

*This chapter is adapted from the work published in [94].

novel 3D video formats, existing standards are not always expected to be the optimal compression solution for all types of contents. Therefore, deep learning based video compression is desired and expected to achieve higher coding efficiency than traditional standards.

Next I will introduce the related works from two aspects, the hand-craft compression and recent deep learning based compression.

4.1.1 Hand-crafted Video Compression

Conventional well-known video coding algorithms, including H.261, MPEG-4 Part 2, H.264/AVC [97], HEVC/H.265 [47] and ongoing Versatile video coding (VVC) standards [98], have also achieved impressive performance through the efforts from JCT-VC experts. H.261 was developed in 1990s by ITU for data rates that are multiples of 64 Kbps, which supports two resolutions, i.e. Common Interface Format (CIF) with the size of 352×288 and quarter CIF (QCIF) with the size of 172×144 . H.261 has already incorporated the inter prediction, discrete cosine transform, quantization and entropy coding to present the basic encoder architecture. After that, MPEG-2 is defined for compression of audio and visual digital data in 1995, and MPEG-4 is defined in 1998 to improve the coding efficiency. Then the most representative video compression standard H.264/AVC was proposed in 2003, which achieved two times of coding efficiency compared to the previous standard and reduce the file size significantly. H.264 consider the well-designed intra prediction and in-loop filter to improve the coding efficiency, which has achieved a great success in many aspects, such as Blu-Ray disk storage, transmission, broadcasting and so on. In 2010, ITU-T Video Coding Experts Group (VCEG) and MPEG joined together to start a joint collaborative Team on Video Coding (JCT-VC) to start the standardization of HEVC/H.265. HEVC introduced more flexible quadtree structure together with more coding units, prediction units and transform units, and better intra prediction with 33 directions, and more accurate sub-pixel motion estimation and compensation. Besides, SAO filter and deblocking filter with better performance is also proposed. HEVC further reduce the file size by about 50% compared to H.264/AVC. Ongoing VVC will be the next generation compression standard expected by the end of 2020. It supports larger coding unit up to 128×128 , more prediction modes (65 directions for intra prediction), more transform types and other coding tools.

Different from image compression, inter prediction is a key technique in video compression standards, including motion estimation and motion compensation. Inter prediction utilize the temporal similarity of neighboring frames to reduce the transmitted information. As for the order of reference frames, both H.264 and HEVC/H.265 support two configurations, that is low delay and random access. Typically low delay P only use the previous frames as uni-directional

references, while random access allows bidirectional referencing in a hierarchical way. Random access can achieve higher coding efficiency than low delay P. The key technique in inter prediction is integer and fractional motion estimation using block matching and motion compensation. Therefore, usually the encoded frames in video compression standards can be categorized into three types, namely, I-frames with only intra prediction, P-frames with inter prediction using uni-directional reference frames, and B-frames with inter prediction through bi-directional reference frames. That is, P-frames are extrapolated from previous frames and B-frames are interpolated from previously transmitted frames in both the past and future.

4.1.2 Learning-based Video Compression

However, learning video compression has not yet been largely exploited. Although it is similar to learned image compression, the large size of video data and the high temporal-spatial redundancy make it difficult to design an end-to-end architecture, which would be very memory-consuming to train such networks.

Only a few related works [99] [100] have been proposed. Wu et al. [99] firstly proposed to use image interpolation network to predict frames except for key frames. The interpolation network is based on U-net architecture and it has the fixed interpolation period, such as 12 frames regardless of input sequences. The results are comparable with H.264/AVC. Chen et al. [100] relied on traditional block based architecture to use CNN nets for predictive and residual signals. It is a hybrid video compression approach, by combining the traditional standard and CNN reconstruction modules. Very recently, there are some new papers released from industries, such as Qualcomm and WaveOne. I believe to further exploit learning video compression algorithms is still highly desired and not fully investigated yet.

4.2 Proposed Interpolation Loop

4.2.1 Problem Formulation

Given an image, which has been explained in Chapter 3 as Fig. 4.1(a), a compression system can be formulated as,

$$\begin{aligned} y &= f_{\theta}(x) \\ \hat{x} &= g_{\phi}(\hat{y}) \end{aligned} \tag{4.1}$$

where x , \hat{x} , y , and \hat{y} are the original images, reconstructed images, compressed data (also called latent presentation) before quantization, and quantized compressed data, respectively. θ and ϕ are

optimized parameters in the analysis and synthesis transforms, respectively.

To obtain high-level features, the analysis and synthesis transforms can be composed into a sequence of consecutive down(up)sampling operations, which can be implemented by convolutional or transposed convolutional filter with a stride of 2. Our network architecture mainly refer to the autoencoder in [54], but according to [73], it is pointed that super resolution is achieved more efficiently by first convolving the images and then upsampling, instead of first upsampling and then convolving. Thus we use 2 convolutional filters as one down(up)sampling unit, and the network architecture is already given in the previous section. Assume we have n downsampling units and the number of convolutional filters in the last layer is K , the compressed data y will have the dimension of $\frac{H}{2^n} \times \frac{W}{2^n} \times K$. In practice, $n = 3$, $K = 48$, H and W are set as 128 due to memory limitation.

Based on the rate-distortion cost function in traditional codecs, the loss function is defined as follows:

$$J(\theta, \phi; x) = \lambda D(x, \hat{x}) + R(\hat{y}) \quad (4.2)$$

where D represents the distortion; R represents bits required to encode the quantized compressed data \hat{y} .

Considering that a video consists of consecutive frames, a video compression system can be simply extended from image compression system as

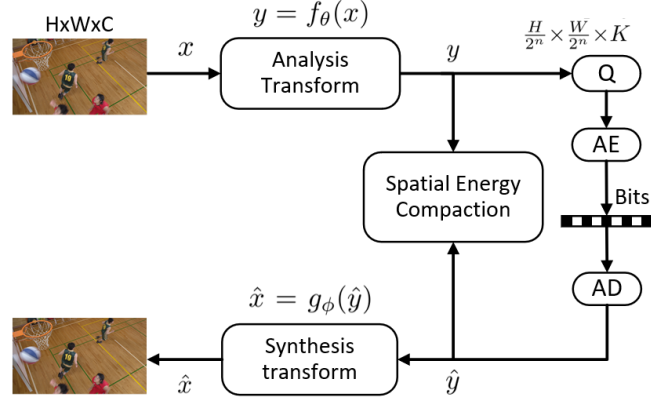
$$\begin{aligned} y^{(t)} &= f_{\theta}(x^{(t)}) \\ \hat{x}^{(t)} &= g_{\phi}(\hat{y}^{(t)}) \end{aligned} \quad (4.3)$$

where $x^{(t)}$ represents $\{x^{(0)}, x^{(1)}, \dots, x^{(T)}\}$, and similar notations are for $y^{(t)}, \hat{y}^{(t)}, \hat{x}^{(t)}, t \in [0, T)$. We define T as group of pictures (GOPs) as HEVC/H.265 [47], which can be encoded and decoded independently. Two consecutive groups share the same boundary frames.

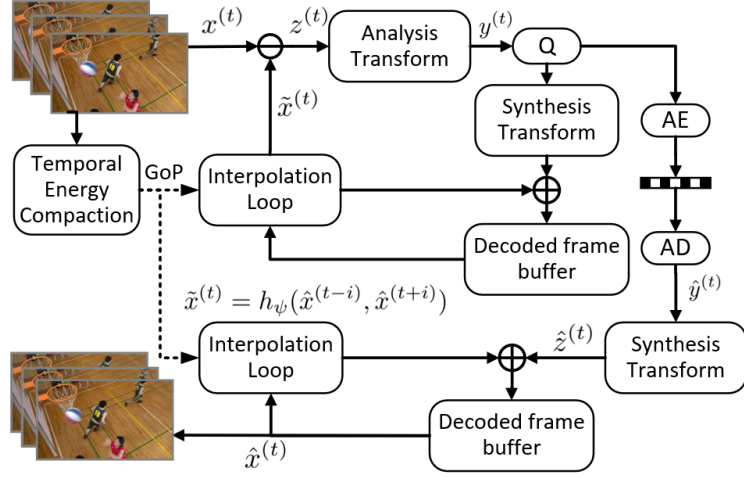
Due to the temporal similarity of neighboring frames, encoding residual information between two frames can gain more coding efficiency than encoding them separately. Thus, a more general form of video compression system is rewritten as

$$\begin{aligned} z^{(t)} &= x^{(t)} - \tilde{x}_E^{(t)} \\ y^{(t)} &= f_{\theta}(z^{(t)}) \\ \hat{z}^{(t)} &= g_{\phi}(\hat{y}^{(t)}) \\ \hat{x}^{(t)} &= \hat{z}^{(t)} + \tilde{x}_D^{(t)} \end{aligned} \quad (4.4)$$

where $\tilde{x}_E^{(t)}$ and $\tilde{x}_D^{(t)}$ are predicted frame using neighboring frames at encoder and decoder side, respectively. For the first frame, there is no previous information, i.e. $\tilde{x}_E^{(0)} = \tilde{x}_D^{(0)} = 0$, video



(a) Learning Image Compression



(b) Learning Video Compression

Fig. 4.1 Overview of our proposed learning video compression with spatial-temporal energy compaction.

compression reduces to a image compression, therefore, we call these key frames as I-frame. The block diagram of the proposed learning image compression is illustrated in Fig. 4.1(b).

4.2.2 Interpolation Loop

The closer generated predictive frame $\tilde{x}^{(t)}$ gets to raw frames $x^{(t)}$, the fewer information $z^{(t)}$ has. Therefore, a high-quality frame interpolation network is desired. We use the latest work [101] to formulate the interpolation as a convolution h over two neighboring frames as

$$\tilde{x}^{(t)} = h_{\psi}(x^{(t-i)}, x^{(t+i)}) \quad (4.5)$$

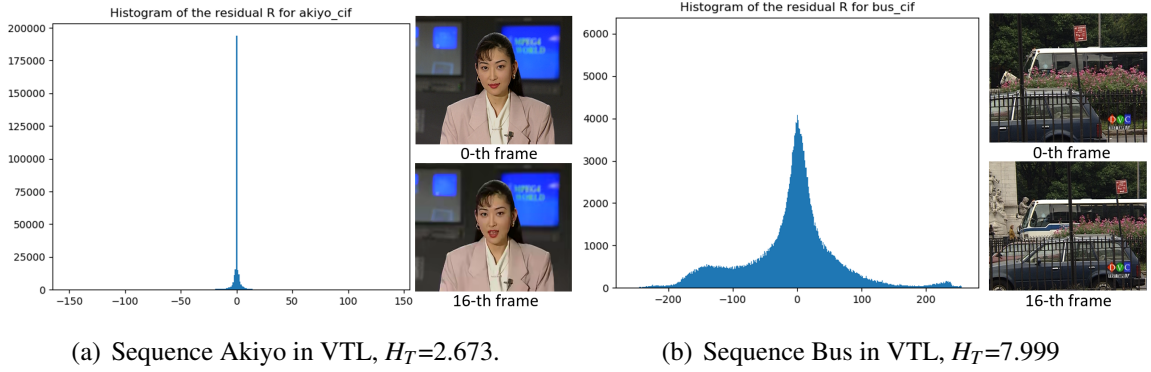


Fig. 4.2 Examples of Temporal Energy Histogram for R_T

where i is the distance between reference frames and generated frames. ψ are optimized parameters in interpolation network h . More importantly, according to Eq.(4.4), predicted frames should be kept the same between encoders and decoders to prevent quality gap,

$$\tilde{x}_E^{(t)} = \tilde{x}_D^{(t)} \quad (4.6)$$

Therefore, the encoder and decoder should see the same information equally. Then, the input for interpolation network should be reconstructed frames, not raw frames. Eq.(4.5) is rewritten as

$$\tilde{x}^{(t)} = h_{\psi}(\hat{x}^{(t-i)}, \hat{x}^{(t+i)}) \quad (4.7)$$

We use a local interpolation loop at the encoder side to store reconstructed frames in the buffer, as Fig. 4.1(b).

4.3 Proposed Spatial-Temporal Energy Compaction for Video Compression

To further reduce the temporal redundancy, inspired by HEVC random access [47] and the work [99], we use a hierarchy interpolation method, which can be illustrated as

$$\begin{aligned} z^{(0)} &= x^{(0)}, z^{(T)} = x^{(T)} \\ \hat{x}^{(\frac{T}{2})} &= h_{\psi}(\hat{x}^{(0)}, \hat{x}^{(T)}) \\ \hat{x}^{(\frac{T}{4})} &= h_{\psi}(\hat{x}^{(0)}, \hat{x}^{(\frac{T}{2})}), \hat{x}^{(\frac{3T}{4})} = h_{\psi}(\hat{x}^{(\frac{T}{2})}, \hat{x}^{(T)}) \end{aligned} \quad (4.8)$$

The hierarchy interpolation is recursively conducted until all the frames are reconstructed.

Each video contents has different motion textures, so the T should be adaptively selected to fit the motion characteristic of videos, so we propose an adaptive T determination strategy based

on temporal energy compaction. We define the temporal motion difference in two neighboring I-frames with a proper distance τ (in our experiments $\tau = 16$) as

$$R_T = x^{(\tau)} - x^{(0)} \quad (4.9)$$

then, we consider the distribution of R_T , and calculate the entropy of R_t as

$$H_T = \mathbb{E}[-P_{R_T} \log_2 P_{R_T}] \quad (4.10)$$

H_T describes the motion characteristic of video sequences, as shown in Fig. 4.2. Large H_T implies that this video has fast motion objects, while low motion videos has small H_T . Then we propose to select the T using

$$T = \begin{cases} 2, & U \leq H_T \\ 8, & L \leq H_T < U \\ 16, & H_T < L \end{cases} \quad (4.11)$$

where L, U are constants for lower and upper bounds. Low motion videos are assigned with $T = 16$, that is, intermediate $(T - 1)$ frames can be interpolated, without destroying the quality. In this case, $z^{(t)}$ is already small, so we send fewer I-frames to achieves temporal energy compaction. As for high motion videos, T is only set as 2, because I-frames do not provide enough information to interpolate a high-quality frame, so we remove the hierarchy interpolation to prevent the error propagation.

To visualize how temporal energy compact works, we conduct the experiments of learning video compression with different T , as shown in Fig. 4.3. Along with the increasing of T , coding efficiency gets improved, but the performance almost saturates between $T = 8$ and $T = 16$. Our approach can adaptively select T to achieve better rate-distortion optimization than the case with constant T .

4.4 Implementation Details

4.4.1 Dataset

To train our image compression models, we used a subset of ImageNet database [76], and cropped them into millions of 128×128 samples. For testing, we used commonly used Kodak lossless image database [77] with 24 uncompressed 768×512 images. To validate the robustness of our proposed method, we also tested the proposed method on the CVPR workshop CLIC validation

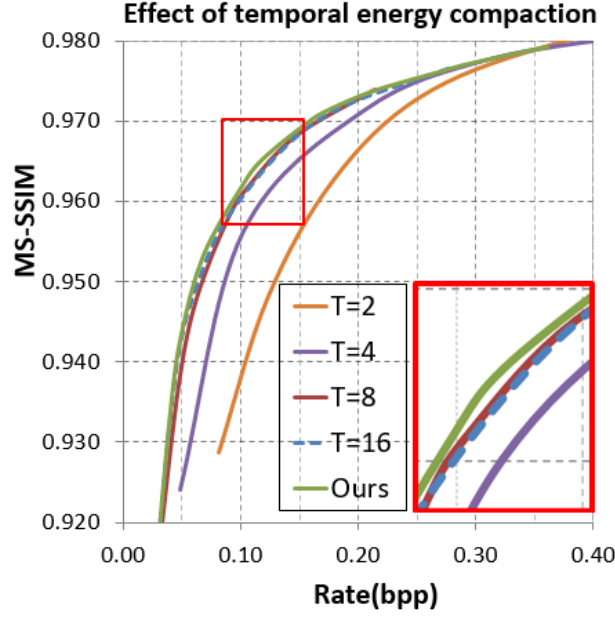


Fig. 4.3 Ablation Study.

dataset [85] with large and various resolutions up to about 2K. To test the performance of video compression, we use the widely used Video Trace Library (VTL) dataset [102], which includes 20 videos with the resolution of 352×288 and 8 test sequences with the resolution of 832×480 and 416×240 , which are commonly used by video coding standardization group with rich texture scenes and motion scenes.

4.4.2 Training Details

To train our image compression autoencoder, the model was optimized using Adam [70]. Batch size is 16. Learning rate was maintained at a fixed value of 1×10^{-4} during the training process. In our experiments, we add the energy compaction penalty to the model at high bit rate, and train for several 10^5 iterations, and then train the model up to several 10^6 iterations for each λ . By introducing different λ to fine-tune a pre-trained autoencoder, we can obtain variable bit rates. We have found that by changing λ with a pre-trained autoencoder, the energy distribution property will not be changed largely, as long as the initial state of the parameters in the models already has a good spatially energy compaction. Thus, we only consider the penalty for one λ trial when training the neural network. Here, we obtained six models with λ in the set $\{2, 4, 8, 16, 32, 64\}$.

In our video compression approach, we use the pre-trained models of [101] to build our reconstruction loop. The value of L , U determines the lower and upper bound for the entropy term to decide I-frame period T . At one extreme, too large L makes all the contents select $T = 16$, and

our performance degrades to the performance with $T = 16$ as shown in Fig. 4.3. At the other extreme, too small U makes all the contents select $T = 2$, and our performance degrades to the curve of $T = 2$. When the interval between L and U is too large, all the contents select $T = 8$, and our curve degrades to the curve $T = 8$. L and U is determined according to the textures of our used dataset. I conduct a preliminary experiment to observe the distribution of H_T , by using several sequences in VTL dataset. The histogram of H_T is shown in Fig. 4.4, and the mean of H_T is about 6.5. By checking the histogram of H_T using VTL dataset, we used $L = 6.0$, $U = 8.0$ in Eq.(4.11) to ensure majority of sequences to select proper T and averaged T is equal to 8.

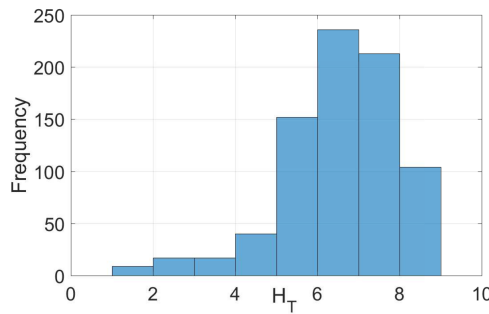


Fig. 4.4 The histogram of H_T in VTL dataset.

The encoding of $z^{(t)}$ can be compressed either the trained models by our image architectures or traditional codecs. In the following results, we use BPG to encode residual for simplicity because it can provide more quality levels.

4.4.3 Measurements

To achieve high subjective quality, we used a popular MS-SSIM [78] as distortion term defined by $D = 1 - \text{MS-SSIM}(x, \hat{x})$. To measure the coding efficiency, the rate is measured in bits per pixel (bpp), and the rate-distortion (RD) curves are drawn to demonstrate their coding efficiency.

4.5 Experimental Results

4.5.1 Frame-level Results

For low motion sequence, such as Akiyo, PSNR and MS-SSIM is almost stable. For high motion sequences, I depict the frame-level PSNR and MS-SSIM of the sequence Bus as shown in Fig. 4.5. T is equal to 8. So it can be observed that the quality fluctuates periodically. The maximal PSNR difference is about 1.5dB, which is not very large. Our approach generates stable reconstruction quality generally, but it has some ghosting artifacts due to interpolation error.

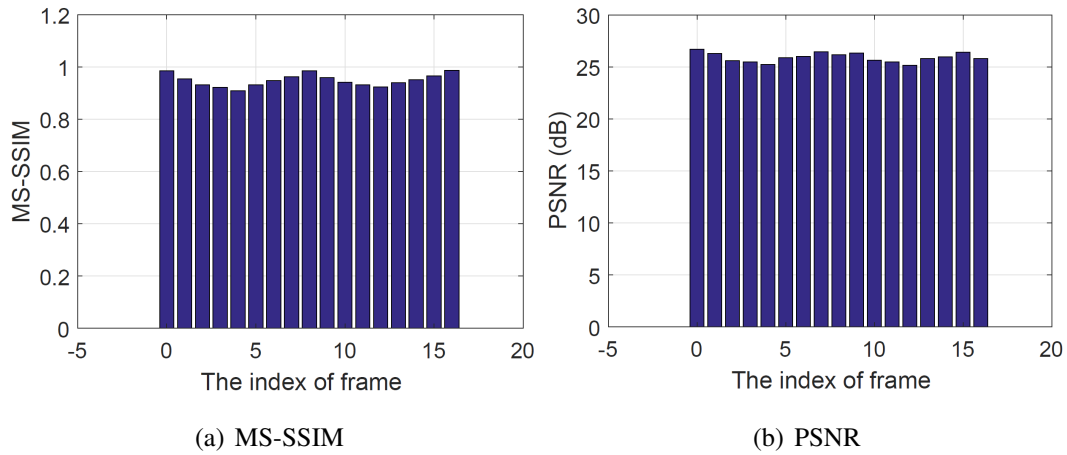


Fig. 4.5 Ablation Study.

4.5.2 Performance Comparison

I compare my method with traditional video compression algorithm and recent learning video compression methods [99]. To reach a fair comparison, I use the the averaged results on each video sequences as [99].

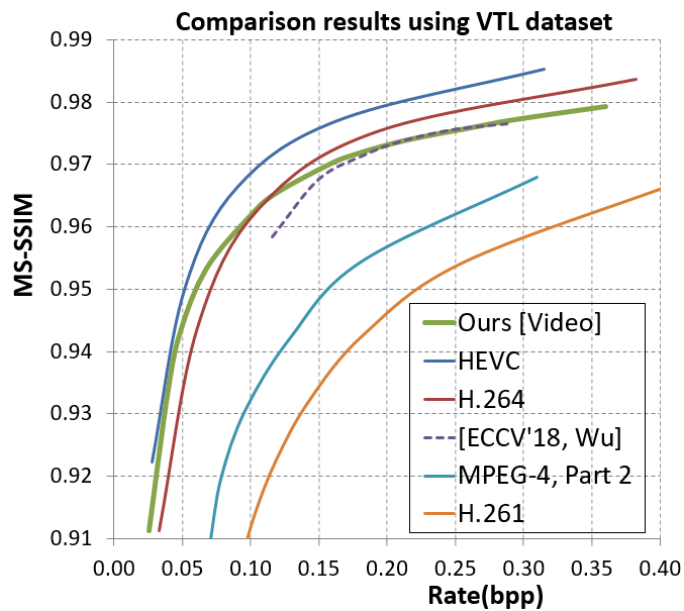


Fig. 4.6 Comparison results using different datasets.

The performance using VTL is shown in Fig. 4.6. For HEVC/H.265 and H.264, we use the official software HM 16.0 [103] and JM 19.0 [104] with random access configuration. The GOP is set as 8 and intra period is also 8 to make the comparison fair. For MPEG-4 Part2 and H.261, we use FFMPEG software. The data point of [99] are from their original paper. It is observed that our

method outperforms MPEG-4 and H.261 significantly and is competitive with H.264. Moreover, we offer a wide range of bit rates and achieve better performance even at low bit rate than the work [99], which benefits from our proposed interpolation loop and temporal energy compaction.

To cover a variety of video contents, we also test our codec using common test sequences, following the work [100], whose results are a little worse than H.264. The RD curves of eight sequences are shown in Fig. 4.7 and Fig. 4.8 and Fig. 4.9. It can be observed that our method outperforms H.264 for most sequences, and even outperforms HEVC/H.265 for sequences BasketballPass and BQSquare.

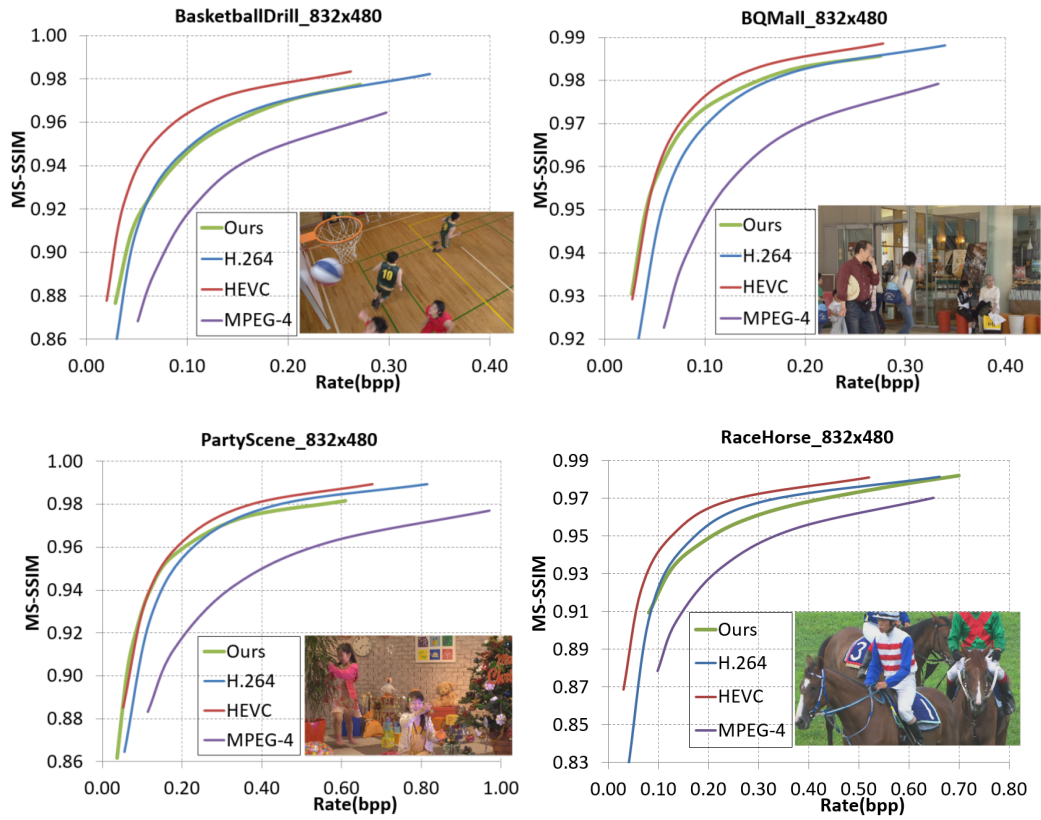


Fig. 4.7 Comparison results for each video sequence with the resolution of 832×480 .

4.5.3 Visualization

Some reconstructed frames from VTL dataset are shown in Fig. 4.10 and Fig. 4.11. Using the interpolation loop, the rate can be greatly saved. Clear block artifacts are observed for MPEG-4 compressed frames. Many details and shapes, such as woman's eyes in Fig. 4.11, are destroyed in H.264 compressed frames. Unlike them, our approach do not have any block artifacts to produce visually pleasant results.

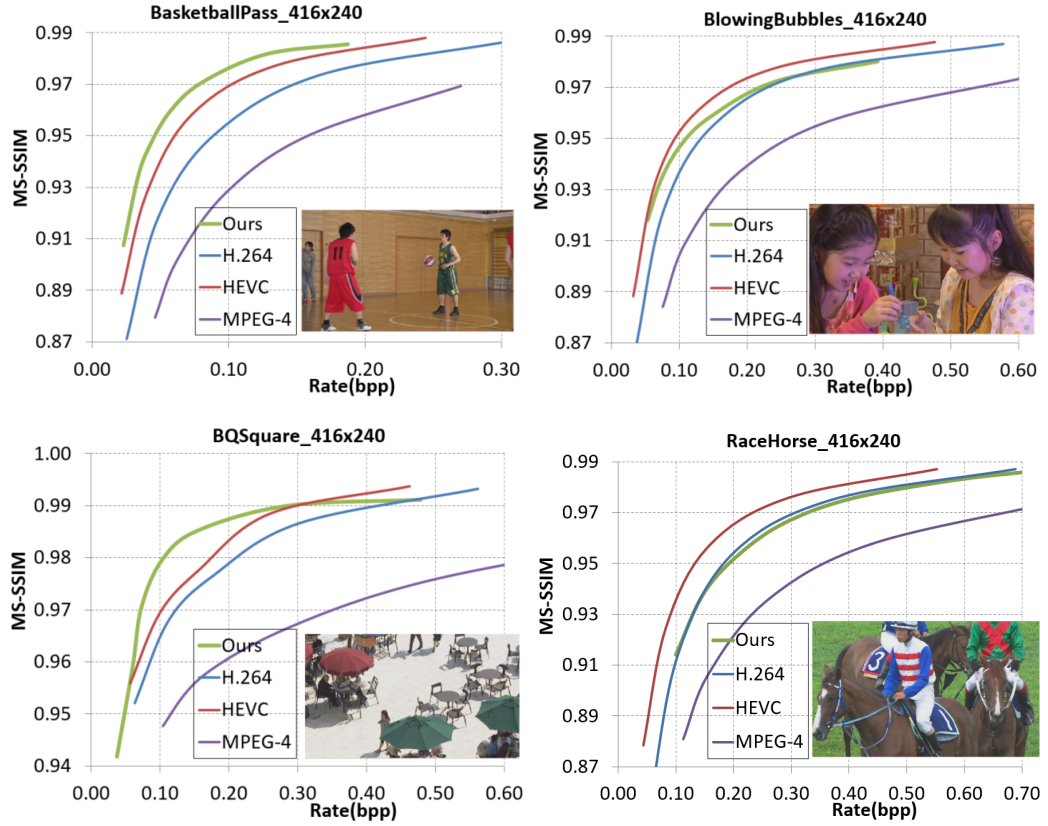


Fig. 4.8 Comparison results for each video sequence with the resolution of 416×240 .

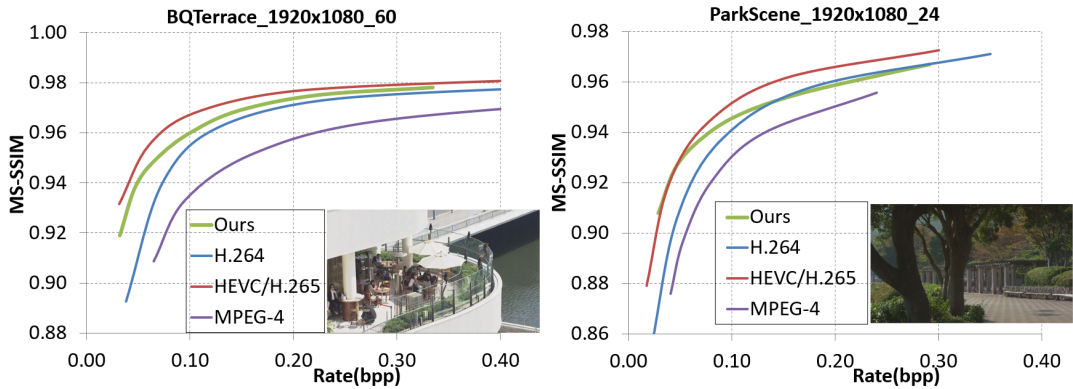


Fig. 4.9 Comparison results for each video sequence with the resolution of 1920×1080 .

Two more reconstructed frames with the resolution of 352×288 from VTL dataset are shown in Fig. 4.12 and Fig. 4.13. Fig. 4.12 visualizes one frame in sequence *silent_cif*, where we can observe that the details in the sweater are well kept in our method, while are largely destroyed in other cases. Fig. 4.13 shows one frame in sequence *container_cif*, where the water waves look more natural and no block boundaries appear in our method. All the used dataset is publicly available.



Fig. 4.10 Example of one reconstruction frame in Video akiyo_cif from VTL dataset.



Fig. 4.11 Example of one reconstruction frame in Video paris_cif from VTL dataset.

4.6 Chapter Summary

In this chapter, we have proposed a learning based video compression through spatial-temporal energy compaction. This proposal is an extension of deep learning based image compression with spatial energy compaction in Chapter 3, by extending image compression to video compression. I generalize the learning-based video compression, by considering an interpolation loop and adaptive interpolation period selection based on entropy of temporal information. The residual information after interpolation is also transmitted to decoder side. This mechanism is similar to random assess

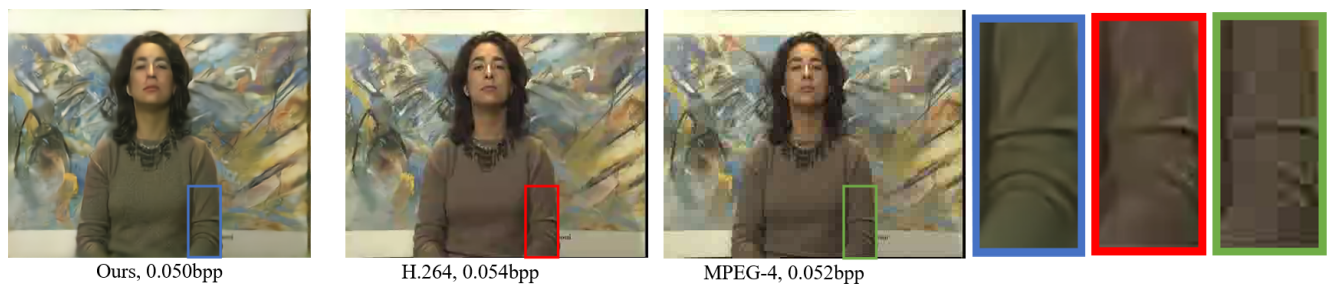


Fig. 4.12 Example of one reconstruction frame in the video sequence (silent_cif, 352×288) from VTL dataset.



Fig. 4.13 Example of one reconstruction frame in the video sequence (container_cif, 352×288) from VTL dataset.

configuration with B-frames. The group of picture size can be adaptively selected according to the motion characteristics of input sequences. Experimental results demonstrate that the proposed video compression approach outperform MPEG-4, and is competitive with widely used H.264. The proposed method can achieve visually better results than traditional standards.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In order to achieve high-accuracy quality assessment and high efficiency image/video compression based on deep learning, we propose several algorithms in this thesis.

In Chapter 2, a fully-blind and fast image quality predictor using convolutional neural networks is proposed, which mainly consists of two strategies. One is called distortion clustering strategy. This strategy is proposed based on the distribution function of the intermediate layer of the CNNs to make IQA achieve better performance for different kinds of distortion artifacts. Second is pre-saliency map. After analyzing the relation between saliency map and CNN prediction errors, we accelerate the IQA to adaptively apply CNN computation and assign weights for non-skipped patches. Experimental results demonstrate our proposed method achieve high correlation coefficient with subjective quality scores, and outperform previous IQA methods. Besides, by taking advantage of pre-saliency map, our method can achieve low time complexity.

In Chapter 3, a lossy image compression based on convolutional autoencoder is proposed. First of all, I compare the performance of different architectures. Then I have presented the CAE architecture and discussed the impact of different network structure on the coding performance. A mathematical analysis with regard to the energy compaction property based on CAE is provided to define a normalized coding gain, which is a measure of compression capability. Based on the above analysis, a regularizer is given to be incorporated into a loss function to train the CAE models to achieve a higher coding gain. The experimental results revealed that our method outperforms the image compression standard HEVC-intra in terms of MS-SSIM quality metric. A thorough perceptual quality study is conducted based on this algorithm.

In Chapter 4, a learning based video compression through spatial-temporal energy compaction is proposed. The spatial energy compaction is based on Chapter 3. Then I generalize the learning-based image compression to video compression, by considering an interpolation loop and adaptive interpolation period selection based on entropy of temporal information. Experimental results demonstrate that my proposed video compression approach outperform MPEG-4, and is competitive with commonly used H.264. My method can achieve visually better results than H.264/AVC.

In short, this thesis has validated the effectiveness of applying deep learning to image compression and quality assessment. Superior performance has been demonstrated compared to existing works.

5.2 Future Work

In the future, there are several feasible research directions that require further discussions and study, as shown in Fig. 5.1. In total there are four possible directions, that is deep learning based lossy image compression, deep learning based lossless image compression, deep learning based lossy video compression and deep learning based lossless video compression.

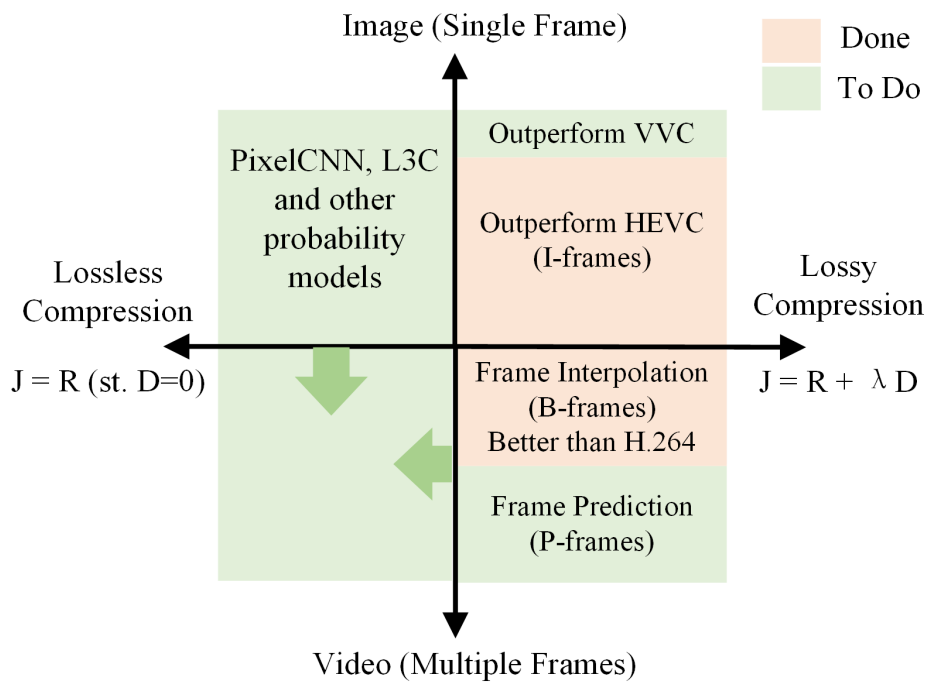


Fig. 5.1 Future directions

Lossy compression usually consider the rate-distortion tradeoff, that is, to consider to minimize the distortion with the constraint of encoding bits, while lossless compression mainly consider how to estimate an accurate entropy model to match the marginal distribution from the viewpoint of information theory, and cross-entropy is usually used to represent the required encoding bits to transmit the original images. Thus, the loss functions are obviously different. The loss function for lossy compression is $J = R + \lambda \times D$, but for lossless compression, loss function is only $J = R$. On the one hand, lossless compression can be regarded as an learning-based entropy coders and can be incorporated into lossy compression to further improve the coding performance. On the other hand, lossless compression can predict the probability model for one variable, and lossy compression only samples one value with the largest probability. From this aspect, many architectures of lossy compression can be extended to use in lossless compression. For example, ETH group developed the learned lossless Compression (L3C) in CVPR 2019 for images with hierarchical autoencoder structures, which looks similar to the hyper autoencoder in lossy compression. Therefore, there is a strong relationship between lossy and lossless compression.

Images and videos also have strong relationship between them. Images can be regarded as single still frame, while videos are extension of images along the time dimension, by consisting of multiple frames with temporal redundancy. That is the most obvious difference between them. For video compression, not only the texture characteristic of single still frame, but also the delay, temporally motion characteristic between adjacent frames, bits allocation and many other factors are also important for the video perceptual quality, which is different from images. From this aspect, I can also consider a good differentiable and perceptual video quality metric when optimizing the neural networks. For example, VMAF (Video Multimethod Assessment Fusion), proposed by Netflix in 2016, is a popular quality measurement recently. Some better distortion measurements can also be considered in learning-based compression architecture, instead of PSNR or MS-SSIM only.

In terms of the targets of each direction, next step for lossy image compression is to outperform versatile video coding (VVC) with intra profile, by further exploring more efficient entropy estimation and network structure. Besides, PSNR is difficult to reach traditional codecs compared to MS-SSIM and the inner reasons can be analyzed. Regarding lossy video compression, next step I need to consider the better way to capture the motion vector, because fast moving sequences are not friendly to current learning based interpolation networks. For instance, deep learning based frame prediction for videos can be one alternative, targeting at compression of video P-frames. Regarding lossless image compression, some recent deep learning methods, such as PixelCNN, PixelRNN, learned lossless compression(L3C) and bit back coding, have

investigated several solutions. In the future I can develop more accurate probability models to achieve promising results. Regarding deep learning based lossless video compression, I have not witnessed some related studies, but the progress on learned lossless image compression and lossy video compression can provide some hints for lossless video compression.

Bibliography

- [1] Z. Cheng, M. Takeuchi, J. Katto, “A Pre-Saliency Map Based Blind Image Quality Assessment via Convolutional Neural Networks”, IEEE Intl. Symposium on Multimedia, pp. 1-6, Dec. 11-13, 2017.
- [2] Z. Cheng, M. Takeuchi, K. Kanai, J. Katto, “A Fast No-reference Screen Content Image Quality Prediction using Convolutional Neural Networks”, IEEE Intl. Conf. on Multimedia and Expo (ICME) workshop, July 23-27, 2018.
- [3] Z. Cheng, M. Takeuchi, K. Kanai, J. Katto, “A Fully-blind and Fast Image Quality Predictor with Convolutional Neural Networks”, IEICE Trans. on Fundamentals, Vol.E101-A, No.9, pp.1557-1566, Sep. 2018.
- [4] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity”, IEEE Trans. on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004. <http://www.ece.uwaterloo.ca/~z70wang/research/ssim/>
- [5] H. R. Sheikh, A. C. Bovik, G. de Veciana, “An information fidelity criterion for image quality assessment using natural scene statistics”, IEEE Trans. on Image Processing, vol. 14, no. 12, pp. 2117-2128, Dec. 2005.
- [6] L. Zhang, Lei Zhang, X. Mou and D. Zhang, “FSIM: A Feature Similarity Index for Image Quality Assessment”, IEEE Trans. on Image Processing, vol. 20, no. 8, pp. 2378-2386, Aug. 2011.
- [7] L. Zhang, Y. Shen and H. Li, “VSI: A Visual Saliency-Induced Index for Perceptual Image Quality Assessment”, IEEE Trans. on Image Processing, vol. 23, no. 10, pp. 4270-4281, Oct. 2014. <http://sse.tongji.edu.cn/linzhang/IQA/VSI/VSI.htm>
- [8] J. Wu, W. Lin, G. Shi, and A. Liu, “Reduced-reference image quality assessment with visual information fidelity”, IEEE Trans. on Multimedia, vol. 15, no. 7, pp.1700-1705, June 04, 2013.

- [9] S. Bosse, Q. Chen, M. Siekmann, W. Samek, T. Wiegand, "Shearlet-based reduced reference image quality assessment", *IEEE Int. Conf. on Image Processing*, pp. 2052-2056, Sep. 25-28, 2016.
- [10] Y. Zhang, J. Wu, et al., "Reduced-Reference Image Quality Assessment Based on Discrete Cosine Transform Entropy", *IEICE Trans. Fundamentals*, Vol. E98-A, No. 12, pp. 2642-2649, Dec. 2015.
- [11] A. K. Moorthy and A. C. Bovik, "Blind Image Quality Assessment: From Natural Scene Statistics to Perceptual Quality", *IEEE Trans. on Image Processing*, vol. 20, no. 12, pp. 3350-3364, Dec. 2011.
- [12] M. A. Saad, A. C. Bovik and C. Charrier, "Blind Image Quality Assessment: A Natural Scene Statistics Approach in the DCT Domain", *IEEE Trans. on Image Processing*, vol. 21, no. 8, pp. 3339-3352, Aug. 2012.
- [13] A. Mittal, A. K. Moorthy and A. C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain", *IEEE Trans. on Image Processing*, vol. 21, no. 12, pp. 4695-4708, Dec. 2012.
- [14] P. Ye, J. Kumar, L. Kang and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment", *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1098-1105, Jun 16-21, 2012.
- [15] X. Min, G. Zhai, K. Gu, et al. Blind quality assessment of compressed images via pseudo structural similarity, *Proceedings of the IEEE International Conference on Multimedia and Expo(ICME)*, 11-15 July 2016.
- [16] L. Kang, P. Ye, Y. Li, D. Doermann, "Convolutional Neural Networks for No-Reference Image Quality Assessment", *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1733-1740, June 23-28, 2014.
- [17] Y. Li, L. M. Po, L. Feng and F. Yuan, "No-reference image quality assessment with deep convolutional neural networks", *IEEE Intl. Conf. on Digital Signal Processing*, pp. 685-689, Oct. 16-18, 2016.
- [18] P. P. Dash, A. Wong, A. Mishra, "VeNICE: A very deep neural network approach to no-reference image assessment", *IEEE Intl. Conf. on Industrial Technology*, pp. 1091-1096, March 22-25, 2017.

- [19] C. Sun, H. Li, W. Li, “No-reference image quality assessment based on global and local content perception”, Visual Communications and Image Processing (VCIP), pp. 1-4, Chengdu, Nov. 27-30, 2016.
- [20] S. Bosse, D. Maniry, T. Wiegand and W. Samek, “A deep neural network for image quality assessment”, IEEE Intl. Conf. on Image Processing (ICIP), pp. 3773-3777, Phoenix, Sep. 25-28, 2016.
- [21] C. Pan, Y. Xu, Y. Yan, K. Gu and X. Yang, “Exploiting neural models for no-reference image quality assessment”, Visual Communications and Image Processing, pp. 1-4, Chengdu, Nov. 27-30, 2016.
- [22] L. Zuo, H. Wang and J. Fu, “Screen content image quality assessment via convolutional neural network”, IEEE Intl. Conf. on Image Processing (ICIP), pp. 2082-2086, Phoenix, Sep. 25-28, 2016.
- [23] J. Kim and S. Lee, “Fully Deep Blind Image Quality Predictor”, IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 1, pp. 206-220, Feb. 2017.
- [24] H. Wang, J. Fu, W. Lin, S. Hu, C. C. Jay Kuo and L. Zuo, “Image Quality Assessment Based on Local Linear Information and Distortion-Specific Compensation”, IEEE Trans. on Image Processing, vol. 26, no. 2, pp. 915-926, Feb. 2017.
- [25] L. Itti, C. Koch and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 11, pp. 1254-1259, Nov. 1998.
- [26] Jonathan Harel, J. Harel, C. Koch, and P. Perona, “Graph-based visual saliency”, Intl. Conf. on Neural Information Processing Systems (NIPS), pp. 545-552, Dec. 4-7, 2006.
- [27] S. Chung, M. G. Chung, “An Image Quality Assessment Using Mean-Centered Weber Ratio and Saliency Map”, IEICE Trans. Inf.& Syst., Vol. E99-D, No. 1, pp. 138-140, Jan. 2016.
- [28] K. Gu, S. Wang, H. Yang, et al. Saliency-Guided Quality Assessment of Screen Content Images, IEEE Trans. on Multimedia, Vol. 18, No. 6, June 2016.
- [29] Z. Ni, L. Ma, H. Zeng, C. Cai, K. Ma, Screen content image quality assessment using edge model, IEEE Intl. Conf. on Image Processing (ICIP), 25-28 Sep. 2016.

- [30] Y. Fang, L. Yan, L. Li, and L. Wu, No reference quality assessment for screen content images, Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 10-14 July 2017.
- [31] W. Zhou, L. Yu, Y. Zhou, W. Qiu, M. Wu and T. Luo, Local and Global Feature Learning for Blind Quality Evaluation of Screen Content and Natural Scene Images, IEEE Trans. on Image Processing, vol. 27, no. 5, pp. 2086-2095, May 2018.
- [32] The implementation of different saliency maps in MATLAB:
<http://www.vision.caltech.edu/harel/share/gbvs.php>
- [33] K. Sharifi and A. Leon-Garcia, "Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video", IEEE Trans. Circuits Syst. Video Technol., vol. 5, no. 1, pp. 52-56, Feb. 1995.
- [34] N. Ponomarenko, et al., "Image database TID2013: Peculiarities, results and perspectives", Signal Processing: Image Communication, vol. 30, pp. 57-77, Jan. 2015.
- [35] H. R. Sheikh, M. F. Sabir and A. C. Bovik, "A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms", IEEE Trans. on Image Processing, vol. 15, no. 11, pp. 3440-3451, Nov. 2006.
- [36] N. Ponomarenko, et al, "TID2008 - A Database for Evaluation of Full-Reference Visual Quality Assessment Metrics", Advances of Modern Radioelectronics, Vol. 10, pp. 30-45, Jan. 2009.
- [37] H. Yang, Y. Fang, and W. Lin, "Perceptual quality assessment of screen content images", IEEE Trans. on Image Processing, vol. 24, no. 11, pp. 4408-4421, Nov. 2015.
- [38] Jia, Yangqing et al., "Caffe: Convolutional Architecture for Fast Feature Embedding", arXiv:1408.5093, 2014.
- [39] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, "Deep Convolutional AutoEncoder-based Lossy Image Compression", Picture Coding Symposium, pp. 1-5, June 24-27, 2018.
- [40] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, "Energy Compaction-Based Image Compression Using Convolutional AutoEncoder", IEEE Transactions on Multimedia, Aug. 2019.

- [41] Z. Cheng, P. Akyazi, H. Sun, J. Katto, T. Ehrahimi, "Perceptual Quality Study on Deep Learning based Image Compression", Intl Conf. on Image Processing (ICIP), Taipei, Taiwan, Sep. 22-25, 2019.
- [42] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, , "Performance Comparison of Convolutional AutoEncoders, Generative Adversarial Networks and Super-Resolution for Image Compression", CVPR Workshop CLIC, Salt Lake City, UT, USA, June 18-22, 2018.
- [43] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, "Deep Residual Learning for Image Compression", CVPR Workshop CLIC, Long Beach, California, USA, June 16-20, 2019.
- [44] G. K Wallace, "The JPEG still picture compression standard", IEEE Trans. on Consumer Electronics, vol. 38, no. 1, pp. 43-59, Feb. 1991. Libjpeg, <https://jpeg.org/jpeg/software.html>
- [45] Majid Rabbani, Rajan Joshi, "An overview of the JPEG2000 still image compression standard", ELSEVIER Signal Processing: Image Communication, vol. 17, no. 1, pp. 3-48, Jan. 2002. OpenJPEG, <https://jpeg.org/jpeg2000/software.html>
- [46] F. Bellard, <https://bellard.org/bpg/>
- [47] G. J. Sullivan, J. Ohm, W. 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.
- [48] S. Li et al., "Closed-Form Optimization on Saliency-Guided Image Compression for HEVC-MSP", IEEE Transactions of Multimedia, Vol. 20, No. 1, Jan. 2018.
- [49] S. Zhu, M. Li, C. Chen, S. Liu, and B. Zeng, "Cross-Space Distortion Directed Color Image Compression,", IEEE Transactions of Multimedia, Vol. 20, No. 3, March 2018.
- [50] Z. Chen, Y. Li, F. Liu, Z. Liu, X. Pan, W. Sun, Y. Wang, Y. Zhou, H. Zhu, S. Liu, "CNN-Optimized Image Compression with Uncertainty based Resource Allocation", CVPR Workshop and Challenge on Learned Image Compression (CLIC), pp. 1-4, June 17-22, 2018.
- [51] W. Tao et al., "An End-to-End Compression Framework Based on Convolutional Neural Networks," 2017 Data Compression Conference (DCC), pp. 463-463, Snowbird, UT, 2017.
- [52] P. Vincent, H. Larochelle, Y. Bengio and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders", Intl. conf. on Machine Learning (ICML), pp. 1096-1103, July 5-9. 2008.

- [53] Lucas Theis, Wenzhe Shi, Andrew Cunningham and Ferenc Huszar, “Lossy Image Compression with Compressive Autoencoders”, Intl. Conf. on Learning Representations (ICLR), pp. 1-19, April 24-26, 2017.
- [54] Johannes Balle, Valero Laparra, Eero P. Simoncelli, “End-to-End Optimized Image Compression”, Intl. Conf. on Learning Representations (ICLR), pp. 1-27, April 24-26, 2017.
- [55] Johannes Balle, D. Minnen, S. Singh, S. J. Hwang, N. Johnston, “Variational Image Compression with a Hyperprior”, Intl. Conf. on Learning Representations (ICLR), pp. 1-23, 2018.
- [56] J. Balle, “Efficient Nonlinear Transforms for Lossy Image Compression”, Picture Coding Symposium, 2018.
- [57] D. Minnen, J. Balle, G. Toderici, “Joint Autoregressive and Hierarchical Priors for Learned Image Compression”, arXiv.1809.02736.
- [58] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, L. V. Gool, “Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations”, Neural Information Processing Systems (NIPS) 2017, arXiv:1704.00648v2.
- [59] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, L. V. Gool, “Conditional Probability Models for Deep Image Compression”, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 17-22, 2018.
- [60] M. Li, W. Zuo, S. Gu, D. Zhao, D. Zhang, “Learning Convolutional Networks for Content-weighted Image Compression”, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), June 17-22, 2018.
- [61] G. Valenzise, A. Purica, V. Hulusic, M. Cagnazzo, Quality Assessment of Deep-Learning-Based Image Compression, IEEE 20th International Workshop on Multimedia Signal Processing, Vancouver, Canada, Sep. 2018.
- [62] G. Toderici, S. M.O’Malley, S. J. Hwang, et al., “Variable rate image compression with recurrent neural networks”, arXiv: 1511.06085, 2015.
- [63] G. Toderici, D. Vincent, N. Johnson, et al., “Full Resolution Image Compression with Recurrent Neural Networks”, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, July 21-26, 2017.

- [64] Nick Johnson, Damien Vincent, David Minnen, George Toderici, et al., “Improved Lossy Image Compression with Priming and Spatially Adaptive Bit Rates for Recurrent Networks”, arXiv:1703.10114.
- [65] S. Santurkar, D. Budden, N. Shavit, “Generative Compression”, Picture Coding Symposium, pp. 1-5, June 24-27, 2018.
- [66] O. Rippel, L. Bourdev, “Real-time Adaptive Image Compression”, arXiv: 1705.05823.
- [67] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, “Generative Adversarial Networks for Extreme Learned Image Compression”, arXiv:1804.02958.
- [68] K. He, X. Zhang, S. Ren and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”, IEEE Intl. Conf. on Computer Vision (ICCV), pp. 1026-1034, Santiago, 2015.
- [69] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition”, 2016 IEEE Conference on Computer Vision and Pattern Recognition, arXiv.1512.03385, 2015.
- [70] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, arXiv:1412.6980, pp.1-15, Dec. 2014.
- [71] C. Dong, C. C. Loy, K. He, X. Tang, “Image Super-resolution using Deep Convolutional Networks”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, No. 2, Feb 1, 2016. arXiv.1501.00092.
- [72] C. Dong, C. C. Loy, X. Tang, “Accelerating the Super-Resolution Convolutional Neural Network”. arXiv.1608.00367.
- [73] W. Shi, J. Caballero, F. Huszar, et al. “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network”, Intl. IEEE Conf. on Computer Vision and Pattern Recognition, June 26-July 1, 2016.
- [74] N.S. Jayant and P. Noll, “Digital coding of waveforms”, Englewood Cliffs NJ, Prentice-Hall, 1984.
- [75] J.Katto and Y.Yasuda: “Performance Evaluation of Subband Coding and Optimization of Its Filter Coefficients”, SPIE Visual Communication and Image Processing, Nov.1991.

- [76] J. Deng, W. Dong, R. Socher, L. Li, K. Li and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database”, IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1-8, June 20-25, 2009.
- [77] Kodak Lossless True Color Image Suite, Download from <http://r0k.us/graphics/kodak/>
- [78] Z. Wang, E. P. Simoncelli and A. C. Bovik, “Multiscale structural similarity for image quality assessment”, The 36-th Asilomar Conference on Signals, Systems and Computers, Vol.2, pp. 1398-1402, Nov. 2013.
- [79] G. Bjontegaard, “Calculation of Average PSNR Differences between RDcurves”, ITU-T VCEG, Document VCEG-M33, Apr. 2001.
- [80] JPEG XL: Additional Information to the Final Call for Proposals, ISO/IEC JTC 1/SC 29/WG1 N80024, 80th meeting Berlin, Germany, 7-13 July 2018.
- [81] Software: HEVC Test Model (HM 16.18+SCM-8.7), Available at https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.18+SCM-8.7/
- [82] JPEG software libjpeg, <https://jpeg.org/jpeg/software.html>
- [83] JPEG2000 official software OpenJPEG,
- [84] A.Radford, L. Metz, S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, arXiv: 1511.06434.
- [85] Workshop and Challenge on Learned Image Compression, CVPR, <http://www.compression.cc/challenge/>
- [86] C. Dong, C. C. Loy, K. He, X. Tang. “Learning a Deep Convolutional Network for Image Super-Resolution”, European Conference on Computer Vision (ECCV), 2014.
- [87] <https://github.com/lucastheis/rangecoder>
- [88] JPEG XT reference software, v1.53, Available at <http://jpeg.org/jpegxt/software.html>.
- [89] Software: Kakadu, v7.10.2, Available at <http://www.kakadusoftware.com>.
- [90] Subjective video quality assessment methods for multimedia applications, ITU-T P.910, April 2008.

- [91] Methodology for the Subjective Assessment of the Quality of Television Pictures, ITU-R BT.500-13, Jan. 2012.
- [92] General Viewing Conditions for Subjective Assessment of Quality of SDTV and HDTV Television Pictures on Flat Panel Displays, ITU-R BT.2022, Aug. 2012.
- [93] C. E. Shannon, "A Mathematical Theory of Communication", The Bell System Technical Journal, Vol. 27, pp. 379-423, July, 1948.
- [94] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, "Learning Image and Video Compression through Spatial-Temporal Energy Compaction", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, USA, June 16-20, 2019.
- [95] White paper: Cisco vni forecast and methodology, 2016-2021. 2016.
- [96] "Overview of the MPEG-4 Standard", Coding of Moving Picture and Audio, ISO/IEC JTC1/SC29/WG11 N3156, Dec. 1999.
- [97] T. Wiegand, G. J. Sullivan, G. Bjontegaard, 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, July. 2003.
- [98] G. J. Sullivan and J. R. Ohm, "Versatile video coding Towards the next generation of video compression", Picture Coding Symposium, Jun. 2018.
- [99] C-Y Wu, N. Singhal, P. Krahenbuhl, "Video Compression through Image Interpolation", 15th European Conference on Computer Vision, September 2018.
- [100] T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma. "Deepcoder: A deep neural network based video compression". 2017 IEEE Visual Communications and Image Processing (VCIP), pp. 11C4, Dec 2017.
- [101] S. Niklaus, L. Mai and F. Liu, "Video Frame Interpolation via Adaptive Separable Convolution", IEEE International Conference on Computer Vision (ICCV) 2017.
- [102] Video Trace Library. <http://trace.eas.asu.edu/index.html>
- [103] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, G. J. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Encoder Description", Document JCTVC-R1002, Sapporo, Jul. 2014. https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

-
- [104] A. M.Tourapis, K. Suhring, G. Sullivan, “H.264/14496-10 AVC Reference Software Manual”, Document JVT-AE010, London, UK, 28 June- 3 July 2009. <http://iphome.hhi.de/suhring/tml/download/>

List of Figures

1.1	The merit and demerit of adopting deep learning to compression and quality assessment	3
1.2	The overall diagram and relation of contents for different chapters	6
2.1	Examples of some NSIs in LIVE database [35]	9
2.2	Examples of some SCIs in SIQAD database [37]	10
2.3	The Proposed IQA framework for natural scene images using pre-saliency map. . .	11
2.4	Examples of images and corresponding saliency map: (a)(e)Distorted image "parrots", "monarch", (b)(f)Saliency map using Itti [25], (c)(g)Saliency map using Harel [26], (d)(h)Fast Saliency Map [32].	14
2.5	The average prediction error in salient/homogenous regions.	15
2.6	The effect of threshold on IQA accuracy.	16
2.7	The architecture of CNN for Distortion Recognition.	17
2.8	The histogram of FC1 results for a reference image and various distorted versions. .	18
2.9	The clustering results of distortion types with K-means.	19
2.10	Overview of the proposed fully-blind IQA framework for natural scene images. . .	20
2.11	The architecture of IQA-oriented CNN.	21
2.12	Examples of images, corresponding saliency maps and prediction errors: (a)(g) Distorted images, (b)(h) saliency map using Itti [25], (c)(i) saliency map using Harel [26], (d)(j) fast saliency map [32], (e)(k) S value for each patch, (f)(l) the absolute value of prediction errors produced by CNNs for each patch.	22
2.13	Accuracy and complexity with different ranges.	24
2.14	The convergence curves of the training loss and test loss.	27
2.15	Scatter plots of actual DMOS with (a) PSNR, (b) SSIM [4], (c) VSI [7], (d) BRISQUE [13], (e) CORNIA [14], (f) FFIQP on LIVE database.	29
2.16	Scatter plots of actual DMOS with (a) BLIINDS-II [12], (b) BRISQUE [13], (c) CORNIA [14], (d) FFIQA for cross-database validation.	32

2.17	Scatter plots of actual DMOS with (a) PSNR, (b) SSIM [4], (c) FSIM [6] (d) VSI [7] and (e) proposed method on the SIQAD database.	36
3.1	Block diagram of CAE based image compression.	42
3.2	The CAE network structure.	42
3.3	The GAN structure.	43
3.4	Block diagram of super-resolution based compression.	45
3.5	RD curves of three methods.	47
3.6	The network structure of anchors we used.	48
3.7	Network structure of proposed deep residual learning, where the solid and dotted lines denote the shortcut connection without and with size change, respectively. . .	51
3.8	The network structure of one residual unit.	52
3.9	Block diagram of the proposed CAE based image compression. (The detailed block for downsampling/upsampling is shown in Fig. 3.10)	56
3.10	Downsampling/Upsampling Units with two (De)Convolution Filters.	57
3.11	The effect of activation function in CAE.	58
3.12	Examples of three images and their corresponding feature maps arranged in raster-scan order ($N_6 = 32$): (a)(d)(g) Raw images, (b)(e)(h) Generated 32 feature maps for Y-component by CAE, and the size of each feature map is $\frac{H}{8} \times \frac{W}{8}$, (c)(f)(i) Rotated Y feature maps by PCA, arranged in vertical scan order.	59
3.13	Overview of our proposed CAE image compression architecture. Conv and Deconv use 3×3 the kernel size with a stride of 1. The block for the downsampling/upsampling unit is shown in Fig. 3.14, where Q represents quantization, and AE and AD represent the arithmetic encoder and arithmetic decoder. Here, the factorized entropy model produces a context model that generates an estimated entropy and serves for AE and AD. During the test, I use the JPEG2000 entropy coder. Let us suppose that the number of down(up)sampling units is n and that the number of channels before Q is K , and the latent representation y has the dimension of $\frac{H}{2^n} \times \frac{W}{2^n} \times K$. Additionally, N denotes the number of filters in the downsampling unit. In our experiments, $n = 3$, $K = 48$, $N = 128$, and $C = 3$ for the RGB image.	61
3.14	Two Types of Downsampling/Upsampling Units.	62
3.15	Diagram of a single neuron.	65
3.16	Construction of fake codes to calculate B_k	67
3.17	Statistical observations on energy property.	67
3.18	RD curves of color images for the proposed CAE, JPEG, and JPEG2000	72

3.19	Examples of raw image (a) and reconstructed images (300×300) cropped from Kodak images using (b)JPEG, (c)JPEG2000 and (d)CAE.	73
3.20	BD-rate of the proposed CAE with JPEG2000 as the benchmark.	74
3.21	RD curves of gray images for our proposed CAE and Balle’s work.	74
3.22	Loss curves of different variants.	76
3.23	Visualization of different quantization methods.	78
3.24	Visualization Examples for Baseline.	78
3.25	Visualization Examples for PCA based bit allocation.	79
3.26	Visualization Examples for MAP based bit allocation.	79
3.27	Visualization Examples for Our proposed energy compaction based bit allocation. .	80
3.28	Comparison between different bit allocation methods.	81
3.29	Performance comparison with compression standards in terms of PSNR.	82
3.30	Comparison with related studies using Kodak dataset.	84
3.31	Example of one reconstruction image (kodim01) with approximately 100:1 compression ratio from Kodak dataset.	84
3.32	Example of one reconstruction image (kodim21) with an approximate compression ratio of 200:1 from Kodak dataset.	85
3.33	Visualization of ringing artifacts in compressed images with comparable bitrate about 0.6bpp.	85
3.34	Proposed learned image compression approach.	87
3.35	Performance of recent works on Kodak dataset.	88
3.36	Test images in this study.	89
3.37	Results of MOS vs. bitrate with corresponding confidence interval.	90
3.38	Pairwise Comparison for each bitrate, where for “n/m” in each cell, m denotes how many contents are comparable for each pairwise comparison, and n denotes how many times the codec on y-axis outperforms the codec on x-axis.	91
3.39	Performance with respect to MS-SSIM and PSNR.	92
4.1	Overview of our proposed learning video compression with spatial-temporal energy compaction.	99
4.2	Examples of Temporal Energy Histogram for R_T	100
4.3	Ablation Study.	102
4.4	The histogram of H_T in VTL dataset.	103
4.5	Ablation Study.	104
4.6	Comparison results using different datasets.	104
4.7	Comparison results for each video sequence with the resolution of 832×480	105
4.8	Comparison results for each video sequence with the resolution of 416×240	106

4.9	Comparison results for each video sequence with the resolution of 1920×1080 . . .	106
4.10	Example of one reconstruction frame in Video akiyo_cif from VTL dataset.	107
4.11	Example of one reconstruction frame in Video paris_cif from VTL dataset.	107
4.12	Example of one reconstruction frame in the video sequence (silent_cif, 352×288) from VTL dataset.	107
4.13	Example of one reconstruction frame in the video sequence (container_cif, $352 \times$ 288) from VTL dataset.	108
5.1	Future directions	110

List of Tables

2.1	Time Comparison between different saliency map methods with LIVE images. . .	13
2.2	The mapping table of distortion types.	19
2.3	Time comparison between different saliency map methods with SIQAD images. . .	22
2.4	The effect of the kernel size in the convolution layer.	26
2.5	The effect of the number of features in the convolution layer.	26
2.6	Comparison results of PCC on the TID2008 database.	27
2.7	Comparison results of SROCC on the TID2008 database.	28
2.8	Accuracy performance on the LIVE database.	29
2.9	Comparison results on the LIVE database.	30
2.10	Complexity reduction with the pre-saliency map.	30
2.11	Complexity comparison results (in ascending order).	31
2.12	Comparison results trained on LIVE and tested on TID2008.	32
2.13	Comparison results on the SIQAD database.	33
2.14	Comparison results for each distortion type on the SIQAD database.	34
2.15	Complexity comparison results.	35
3.1	The effect of different input sizes.	44
3.2	The effect of different code sizes and interpolation sizes.	44
3.3	The effect of different quantization bits.	45
3.4	The effect of adaptive strategy for super-resolution.	46
3.5	Performance comparison with 0.15bpp constraint.	47
3.6	The effect of kernel size on Baseline on Kodak, optimized by MSE with $\lambda = 0.015$	49
3.7	The effect of kernel size on HyperPrior on Kodak, optimized by MSE with $\lambda = 0.015$	49
3.8	The effect of kernel size in the auxiliary autoencoder on Kodak, optimized by MS-SSIM with $\lambda = 5$	50
3.9	Comparison of residual networks and upsampling operations on Kodak, optimized by MS-SSIM with $\lambda = 5$	50
3.10	The effect of wide bottleneck on Kodak dataset.	52

3.11	Rate control on CLIC validation dataset [85].	52
3.12	Results on CLIC validation dataset [85].	53
3.13	The model size analysis of HyperPrior-9.	54
3.14	The model complexity of different architectures.	55
3.15	Notations.	64
3.16	Average running time comparison.	75
3.17	Influence of different types of downsampling units	75
3.18	Influence of different number of downsampling units	76
3.19	Influence of different number of channels	77
3.20	Coding gain averaged on Kodak.	81
3.21	Codecs considered in this paper.	88

Publication Lists

JOURNAL PAPERS

- [1] **Zhengxue Cheng**, Heming Sun, Masaru Takeuchi, Jiro Katto, “Energy Compaction-Based Image Compression Using Convolutional AutoEncoder”, IEEE Trans. on Multimedia, Aug. 2019.
- [2] **Zhengxue Cheng**, Masaru Takeuchi, Kenji Kanai, Jiro Katto, “A Fully-blind and Fast Image Quality Predictor with Convolutional Neural Networks”, IEICE Trans. on Fundamentals, Vol.E101-A, No.9, pp.1557-1566, Sep. 2018.
- [3] **Zhengxue Cheng**, Heming Sun, Dajiang Zhou, Shinji Kimura, “Accelerating HEVC Inter Prediction with Improved Merge Mode Handling”, IEICE Trans. on Fundamentals, Vol.E100-A, No.02, pp.546-554, Feb. 2017.
- [4] Liang Qian, **Zhengxue Cheng**, Zheng Fang, Lianghai Ding, Feng Yang, Wei Huang, “A QoE-Driven Encoder Adaptation Scheme for Multi-User Video Streaming in Wireless Networks”, IEEE Trans. on Broadcasting, Vol. 63, No. 1, pp. 20-31, March 2017.
- [5] Heming Sun, **Zhengxue Cheng**, Amir Masoud Gharehbaghi, Shinji Kimura, Masahiro Fujita, “Approximate DCT Design for Video Encoding based on Novel Truncation Scheme”, IEEE Trans. on Circuits and Systems, pp. 1-14, Dec. 2018.
- [6] Kenji Kanai, Bo Wei, **Zhengxue Cheng**, Masaru Takeuchi and Jiro Katto, “Methods for Adaptive Video Streaming and Picture Quality Assessment to Improve QoS/QoE Performances,” IEICE Trans. on Comm., July 2019.

INTERNATIONAL CONFERENCE PAPERS (WITH PEER REVIEW)

- [7] **Zhengxue Cheng**, Pinar Akyazi, Heming Sun, Jiro Katto, Touradj Ebrahimi, “Perceptual Quality Study on Deep Learning based Image Compression”, Intl Conf. on Image Processing (ICIP), Taipei, Taiwan, Sep. 22-25, 2019.
- [8] **Zhengxue Cheng**, Heming Sun, Masaru Takeuchi, Jiro Katto, “Learning Image and Video Compression through Spatial-Temporal Energy Compaction”, Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, USA, June 16-20, 2019.

- [9] **Zhengxue Cheng**, Heming Sun, Masaru Takeuchi, Jiro Katto, “Deep Residual Learning for Image Compression”, CVPR Workshop, Long Beach, California, USA, June 16-20, 2019.
- [10] **Zhengxue Cheng**, Masaru Takeuchi, Kenji Kanai, Jiro Katto, “A Fast No-reference Screen Content Image Quality Prediction using Convolutional Neural Networks”, Intl. Conf. on Multimedia and Expo (ICME) workshop, July 23-27, 2018.
- [11] **Zhengxue Cheng**, Heming Sun, Masaru Takeuchi, Jiro Katto, “Deep Convolutional AutoEncoder-based Lossy Image Compression”, Picture Coding Symposium (PCS), pp. 1-6, San Francisco, CA, USA, June 24-27, 2018.
- [12] **Zhengxue Cheng**, Heming Sun, Masaru Takeuchi, Jiro Katto, “Performance Comparison of Convolutional AutoEncoders, Generative Adversarial Networks and Super-Resolution for Image Compression”, CVPR Workshop CLIC, Salt Lake City, UT, USA, June 18-22, 2018.
- [13] **Zhengxue Cheng**, Masaru Takeuchi, Jiro Katto, “A Pre-saliency Map Based Blind Image Quality Assessment via Convolutional Neural Networks”, IEEE International Symposium on Multimedia (ISM), Taichung, Taiwan, Dec.11-13, 2017.
- [14] **Zhengxue Cheng**, Lianghai Ding, Wei Huang, Feng Yang, Liang Qian, “A unified QoE Prediction Framework for HEVC Encoded Video Streaming over Wireless Networks”, IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp.1-6, Cagliari, Italy, June 7-9, 2017.
- [15] **Zhengxue Cheng**, Lianghai DING, Wei HUANG, Feng YANG, Liang Qian., “Subjective QoE based HEVC Encoder Adaptation Scheme for Multi-User Video Streaming”, IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp.1-6, Nara, Japan, June 1-3, 2016.
- [16] **Zhengxue Cheng**, Heming Sun, Dajiang Zhou, Shinji Kimura, “Merge Mode based Fast Inter Prediction for HEVC”, IEEE Visual Communications and Image Processing (VCIP), pp. 1-4, Singapore, Dec. 13-16, 2015.
- [17] **Zhengxue Cheng**, Heming Sun, Landan Hu, Shinji Kimura, “A Fast Level Filtering Algorithm for Inter Prediction in HEVC Encoder”, Intl. Technical Conference on Circuits /Systems, Computers and Communications (ITC-CSCC), pp.404-407, Seoul, Korea, June 2015.
- [18] Masaki Yasumaru, **Zhengxue Cheng**, Ryota Yokoyama, Kenji Kanai and Jiro Katto, “Accuracy Evaluations of Contact-Free Heart Rate Measurement

- Methods Using 4K Facial Images”, IEEE International Conference on Consumer Electronics (ICCE), Jan. 2019.
- [19] Katsuhiko Hirao, **Zhengxue Cheng**, Masaru Takeuchi and Jiro Katto, “Convolutional Neural Network Based Inverse Tone Mapping for High Dynamic Range Display Using LUCORE,” IEEE International Conference on Consumer Electronics (ICCE), Jan. 2019.
- [20] Chao Liu, Heming Sun, Junan Chen, **Zhengxue Cheng**, Masaru Takeuchi, Jiro Katto, Xiaoyang Zeng and Yibo Fan, “Dual Learning-based Video Coding with Inception Dense Blocks,” Picture Coding Symposium (PCS) 2019, Nov. 2019.
- [21] Yusuke Sakamoto, Shintaro Saika, Masaru Takeuchi, Tatsuya Nagashima, **Zhengxue Cheng**, Kenji Kanai, Jiro Katto, Kaijin Wei, Ju Zengwei and Xu Wei, “Acceleration of Perceptual Quality Driven Adaptive Video Coding for Live Streaming,” IEEE International Symposium on Multimedia (ISM) 2018, Dec. 2018.
- [22] Katsuhiko Hirao, **Zhengxue Cheng**, Masaru Takeuchi and Jiro Katto, “Deep Inverse Tone Mapping Optimized for High Dynamic Range Display”, International Conference on ICT Convergence (ICTC) 2018, Oct. 2018.
- [23] Masaru Takeuchi, Shintaro Saika, Yusuke Sakamoto, Tatsuya Nagashima, **Zhengxue Cheng**, Kenji Kanai, Jiro Katto, Kaijin Wei, Ju Zengwei and Xu Wei, “Perceptual Quality Driven Adaptive Video Coding Using JND Estimation”, Picture Coding Symposium (PCS), June. 2018.

DOMESTIC CONFERENCE PAPERS

- [24] **Zhengxue Cheng**, Heming Sun, Masaru Takeuchi, Jiro Katto, “Lossy Image Compression using Deep Convolutional AutoEncoder”, 情報処理 AVM 学会研究会, June 2018.
- [25] **Zhengxue Cheng**, Masaru Takeuchi, Jiro Katto, “A Blind Image Quality Assessment with Convolutional Neural Networks”, PCSJ/IMPS 2017, Nov. 2017.
- [26] **Zhengxue Cheng**, Heming Sun, Jiro Katto, “A Water Wave Removal Approach for Efficient Video Coding,” PCSJ/IMPS 2018, Nov. 2018.
- [27] Rige Su, **Zhengxue Cheng**, Jiro Katto, “Optimized Image Compression Based on Recurrent Neural Network”, PCSJ/IMPS 2019, Nov. 2019.

- [28] Rige Su, **Zhengxue Cheng**, Jiro Katto, “Recurrent Neural Network based Image Compression”, 信学会 IE 研究会, March 2019.
- [29] 安丸昌輝, 横山怜汰, **程 正雪**, 金井謙治, 甲藤二郎, “4K 顔画像を活用した JadeICA 心拍計測法の精度評価”, 信学会 IE 研究会, March 2019.
- [30] 安丸 昌輝, 横山 怜汰, **程 正雪**, 金井 謙治, 甲藤 二郎, “UHD 顔画像を用いた非接触心拍計測法の精度評価”, PCSJ/IMPS 2018, Nov. 2018.
- [31] 坂本 悠輔, 雑賀 新太郎, 竹内 健, 長島 達也, **程 正雪**, 金井 謙治, 甲藤 二郎, Kaijin Wei, Ju Zengwei, Xu Wei, “主観画質を考慮したライブストリーミング配信のための動画像圧縮符号化”, PCSJ/IMPS 2018, Nov.2018.
- [32] 竹内 健, 坂本 悠輔, 雑賀 新太郎, 長島 達也, **程 正雪**, 金井 謙治, 甲藤 二郎, Kaijin Wei, Ju Zengwei, Xu Wei, “JND 推定を用いた知覚品質主導の適応的動画像圧縮符号化”, PCSJ/IMPS 2018, Nov. 2018.
- [33] 坂本悠輔, 竹内健, 雑賀新太郎, 長島達也, **程正雪**, 金井謙治, 甲藤二郎, Kaijin Wei, Ju Zengwei, Xu Wei, “主観画質を考慮したインターネット動画配信のための動画像圧縮符号化”, 信学会 IE 研究会, Oct. 2018.

My View of Embodiment Informatics

At first, I would like to express my great thanks to Graduate Program for Embodiment Informatics, which gives me many valuable opportunities during my doctoral course. These valuable experiences include four-month overseas internship at EPFL in Lausanne, Switzerland, the Summer School with master and doctor students from Tsukuba University and Tsinghua University, one-month English training program at UC Davis, leading forums, colloquiums with very interesting topics and so many other workshops.

Three years ago, I became one part of this program. My first impression on Embodiment Informatics is the combination of mechanism and information, to make invisible and untouchable things to be visible and touchable in the daily life. However, through the study during my doctoral course, I have realized that it not only represents this definition as its name, but also this program encourages researchers to identify possible areas for innovation and try them, foster researcher to integrate advanced technology from different fields. For example, based on this I can have the courage to try deep learning-based compression, which was very rarely investigated at first. Embodiment informatics includes the fast development of robots currently, but its meaning is not limited to this. Embodiments informatics can have much wider and deeper meaning, but the core concept of this program and its core abilities, such as foresight, imagination and leadership, will be the valuable resources to stay with us for a lifetime.

In my understanding, this program can make researchers to grow up no matter what happens, to adapt to new environments quickly, to have open-minds to new techniques and emerging topics, to possibly lead some novel fields in the future. Through the Graduate Program for Embodiment Informatics, I become more self-confident, because interesting events and activities make me to have wide knowledge from different research fields, good communication skills and improve three key abilities foresight, imagination and leadership. Through the activities and classes provided by Graduate Program of Embodiment informatics, I get to have a deep understanding what is Embodiment informatics and what kind of researchers we could become in the future.

Finally, I also would like to thank all the professors, students and staff involved in leading program. Without them, I could not enjoy my doctoral program and learn many kinds of knowledge. Kobo is a great place to exchange ideas and discuss their own researches. At Kobo, I took part in a joint research on gait phase detection to submit the ROBIO conference. I joined the special seminar to build a self-made companion robot. I hope junior students can enjoy the life here, and open minds to communicate with surrounding people. All these experiences will be a wonderful reward for our life journey. I believe that leading program will foster leading researchers both in both academic and industry.