SCREEN CONTENT VIDEO QUALITY ENHANCEMENT (SCVQE) BASED ON MACHINE LEARNING

ZIYIN HUANG

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University

Department of Electrical and Electronic Engineering

Screen Content Video Quality Enhancement (SCVQE) Based on Machine Learning

Ziyin Huang

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

Doctor of Philosophy

April 2025

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

Ziyin Huang (Name of student)

Abstract

The increasing popularity of intelligent terminals has led to a higher demand for screen content videos. Applications such as the cloud gaming, video conference, online education, etc., rely heavily on Screen Content Coding (SCC). The impact of the COVID-19 pandemic in 2020 further accelerated the necessity of online education and virtual conferences, making SCC indispensable for effective screen sharing. This paradigm shift has elevated SCV from a niche to mainstream media. Consequently, enhancing the quality of screen content videos has become a critical challenge. In this thesis, we conduct an in-depth study on deep-learning-based VQE of SCC and propose effective learning frameworks based on the characteristics of screen content videos (SCVs).

Firstly, we study the dedicated tools - Intra Block Copy (IBC) and palette (PLT) modes in the SCC standard, which induces the corresponding compression loss of the decoded video. Therefore, we propose a novel post-processing network for enhancing decoded screen content videos based on the coding mode information embedded in the coded bitstream. By fusing three binary mode masks derived from dedicated coding tools with the corresponding decoded frame, we aim to elevate the quality of SCVs.

Secondly, different from natural videos, screen content videos often feature abrupt scene switches and frame freezing instances, leading to visible distortions in compressed videos. Existing alignment-based models struggle to effectively enhance scene switch frames and lack efficiency when dealing with frame freezing situations. Therefore, we propose a novel alignment-free method that effectively handles both scene switches and frame freezing. In our approach, we develop a spatial and temporal feature extraction module to compress and extract spatio-temporal information from three groups of frame inputs. This enables efficient handling of scene switches. In addition, an edge aware block

is proposed to extract edge information, which guides the model to focus on restoring the high-frequency components in frame freezing situations. The fusion module is then designed to adaptively fuse the features from three groups, considering different positions of video frames, to enhance frames during scene switch and frame freezing scenarios.

Thirdly, existing multiple-frame models using a fixed range of neighbor frames face challenges in effectively enhancing frames during scene switches and lack efficiency in reconstructing high-frequency details. To address these limitations, we present a novel method proficient in managing scene switches and reconstructing high-frequency information. In the feature extraction part, we develop long-term and short-term feature extraction streams, in which the long-term feature extraction stream learns the contextual information, and the short-term feature extraction stream extracts more related information from shorter input to assist the long-term stream to handle fast motion and scene switches. To further enhance the frame quality during scene switches, we incorporate a similarity-based neighbor frame selector before feeding frames into the short-term stream. This selector identifies relevant neighbor frames, aiding in the efficient handling of scene switches. To dynamically fuse the short-term feature and long-term features, the muti-scale feature distillation focuses on adaptively recalibrating channel-wise feature responses to achieve effective feature distillation. In the reconstruction part, a high-frequency reconstruction block is proposed for guiding the model to restore the high-frequency components.

The frameworks proposed in this thesis are evaluated through comparisons with other state-of-the-art methods, including the posed databases and the in-the-wild databases. Ablation studies and robustness tests confirm the promising performance of our frameworks, highlighting the efficacy of the novel designs in enhancing screen content quality.

List of Publications

Journal Papers

[1] **Ziyin Huang**, Yui-Lam Chan, Ngai-Wing Kwong, Sik-Ho Tsang, Kin-Man Lam, and Wing-Kuen Ling, "Long Short-term Fusion by Multi-scale Distillation for Screen Content Video Quality Enhancement," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 35, pp. 7762-7777, 2025.

[2] **Ziyin Huang**, Yui-Lam Chan, Sik-Ho Tsang, Ngai-Wing Kwong, Kin-Man Lam, and Wing-Kuen Ling, "Spatio-temporal feature learning for enhancing video quality based on screen content characteristics," *Journal of Visual Communication and Image Representation*, vol. 104, pp. 104270, 2024.

[3] **Ziyin Huang**, Yui-Lam Chan, Sik-Ho Tsang, and Kin-Man Lam, "Mode Information Guided CNN for Quality Enhancement of Screen Content Coding," *IEEE Access*, vol. 11, pp. 24149-24161, 2023.

[4] Ngai-Wing Kwong, Yui-Lam Chan, Sik-Ho Tsang, **Ziyin Huang**, and Kin-Man Lam, "Multi-frame spatiotemporal feature and hierarchical learning approach for no-reference screen content video quality assessment," IEEE Transactions on Multimedia, early accepted.

Conference Papers

[1] **Ziyin Huang**, Yue Cao, Sik-Ho Tsang, Yui-Lam Chan, and Kin-Man Lam, "Quality enhancement of screen content video using dual-input CNN," *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp. 797-803.

[2] **Ziyin Huang**, Yui-Lam Chan, Ngai-Wing Kwong, Sik-Ho Tsang, Kin-Man Lam, and Wing-Kuen Ling, "Frame Similarity-Based Screen Content Video Quality Enhancement via Adaptive Long Short-Term Fusion," *2024 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2024, pp. 1-5.

Acknowledgement

I would like to express my deepest gratitude to my supervisor, Dr. Yui-Lam Chan, for his unwavering support, astute guidance, and continuous encouragement throughout my research. His valuable insights and scientific rigor not only shaped my views on research, but will also influence my future career.

I am also profoundly thankful to Dr. Sik-Ho Tsang, Dr. Ngai-Wing Kwong, Prof. Kin-Man Lam, Prof. Wing-Kuen Ling, and the rest of the team at the Artificial Intelligence and Digital Signal Processing Laboratory at The Hong Kong Polytechnic University. Collaborating and exchanging ideas with them has significantly enriched my research experience and contributed to the success of my work.

Besides, I am grateful to the Department of Electrical and Electronic Engineering and The Hong Kong Polytechnic University for providing a conducive work environment and financial support for my research.

Lastly, my heartfelt appreciation goes out to my parents, family, and friends for their constant companionship, love, and understanding.

Contents

List of Figures

List of Tables

Abbreviations

Adam: Adaptive Moment Estimation

AI: All-Intra

CSAB: Channel and Spatial Attention Block

CAMs: Class Activation Maps

CF-STIF: Coarse-to-Fine Spatio-Temporal Information Fusion

CPGA: Coding Priors-Guided Aggregation Network

CTU: Coding Tree Unit

CU: Coding Unit

CTC: Common Test Condition

CAT: Content Adaptive Network based on Two Branches

CBAM: Convolutional Block Attention Module

DF: Deblocking Filter

DS-CNN: Decoder-side Scalable CNN

DCAD: Deep CNN based Auto Decoder

DCNs: Deformable Convolutional Networks

EFC: Early Fusion by Concatenation

EAB: Edge Aware Block

EAST: Edge Aware with Spatio-Temporal Information Fusion Network

DEGREE: Edge Guided Recurrent Residual

EleAttG: Element-wise Attention Gate

FLOPs: Floating Point Operations

GFM: Gabor Feature-based Model

GAN: Generative Adversarial Network

HEVC: High Efficiency Video Coding

HFRB: High-Frequency Reconstruction Block

IFCNN: In-loop Filter using the Convolutional Neural Networks

IBC: Intra Block Copy

LFC: Late Fusion Concatenation

LSTA: Long Short-Term Attention

LSFMD: Long Short-term Fusion by Multi-scale Distillation

LUT: Lookup Table

LDMS: Low Delay Main SCC

LR: Low-Resolution

MOS: Mean Opinion Score

MSE: Mean Squared Error

MemNet: Memory Network

MICNN: Mode Information Guided CNN

MGANet: Multi-frame Guided Attention Network

MFQE 1.0: Multi-Frame Quality Enhancement

MDCNN: Multi-layered Deep CNN

MHFD: Multi-scale Hierarchical Feature Distillation

MSRB: Multi-Scale Residual Block

NLRN: Non-local Recurrent Network

PLT: Palette

PSNR: Peak Signal-to-Noise Ratio

QE-CNN: Quality Enhancement CNN

QECF: Quality Enhancement Network using Cross-Frame Information

QoE: Quality of Experience

QPs: Quantization Parameters

RAM: Recurrent Attention Model

RNN: Recurrent Neural Network

RFDA: Recursive Fusion and Deformable Spatiotemporal Attention

RDBs: Residual Dense Blocks

SAO: Sample Adaptive Offset

SCC: Screen Content Coding

SCVQE: Screen Content Video Quality Enhancement

SCVs: Screen Content Videos

SNFS: Similarity-based Neighbor Frame Selector

STN: Spatial Transformer Network

STA: Spatial-Temporal Adaptive

STAM: Spatio-Temporal Attention Module

STDF: Spatio-Temporal Deformable Fusion

STDR: Spatio-Temporal Detail Information Retrieval

STFE: Spatio-Temporal Feature Extraction

STFF: Spatio-Temporal Feature Fusion

STIB: Spatio-Temporal Information Balance

SR: Super-Resolution

TGAF: Temporal Group Alignment and Fusion Network

TSN: Temporal Segment Network

TDD: Trajectory-pooled Deep-convolutional Descriptor

VRCNN: Variable-filter-size Residue-learning CNN

VQA: Video Quality Assessment

WSSS: Weakly Supervised Semantic Segmentation

Chapter 1

Introduction

1.1 Background

With the rapid development of intelligent terminal technology, mobile devices such as smartphones and tablets have made Screen Content (SC) video more and more widespread. Desktop collaboration, screen sharing, cloud gaming, and more have expanded the scope of video applications. Especially with the recent spread of COVID-19, the demand for online education and virtual conferences is rapidly increasing, with Screen Content Coding (SCC) [1] [2] playing a critical role. Unlike the natural video sequence, as shown in the example of Fig. 1.1(a), captured by a camera, the screen content sequence as in Fig. 1.1(b) can be generated from different mobile terminals directly. It is composed of many static or moving computer-generated images and texts. Additionally, screen content frequently includes repeated patterns, such as those found in graphical user interfaces, spreadsheets, or web pages. Moreover, the color palette in screen content videos tends to be more limited compared to that of natural videos. This is because screen content is often sourced from digital sources that use a specific set of colors for icons, text, and simple graphics. By making use of these screen content characteristics, SCC [1] [2] is proposed as an extension of High Efficiency Video Coding (HEVC) [3] to increase the coding efficiency. In addition to the conventional HEVC intra (INTRA) mode [4], the SCC standard adopts two dedicated coding modes, Intra Block Copy (IBC) and palette (PLT) [5]. IBC [6] uses the reconstructed block of the current frame as the prediction block, improveing compression efficiency by over 30% for screen content videos [6]. PLT

enumerates the color value for each coding block to generate a color table and passes an index for each sample to indicate which color in the color table it belongs to. With PLT, compression efficiency is further improved by 15% over the original code with IBC mode [6]. Despite the coding tools introduced to enhance efficiency, compression artifacts persist in screen content videos due to the specialized tools in the SCC standard. To address this, an HEVC codec utilizes a deblocking filter (DF) and a sample adaptive offset (SAO) to eliminate blocking and ringing artifacts, thereby enhancing the quality of the reconstructed frames. In recent years, deep learning has made significant strides in video enhancement, with various neural network architectures proposed to remove the artifacts from reconstructed videos. Examples include In-loop Filter using the Convolutional Neural Networks (IFCNN) [7], Variable filter-size Residue-learning CNN (VRCNN) [8], DeepCNN based Auto Decoder (DCAD) [9], Multi-layered Deep CNN (MDCNN) [10], and Decoder-side Scalable CNN (DS-CNN) [11]. DCAD and DS-CNN, unlike other architectures that replace the in-loop filter, were designed to improve the video quality at the decoder side. The advantage of these post-processing methods is that they do not require modification to the HEVC codec itself. Hence, the structure proposed in this thesis focuses on video post-processing at the decoder side.

1.2 Challenges and Research Problems

Different from natural videos, screen content videos exhibit distinct characteristics in both spatial and temporal domains, posting specific challenges for quality enhancement. Therefore, this thesis will mainly investigate challenging tasks in spatial and temporal domains in the field of screen content video quality enhancement, and provide solutions to effectively address these issues.

1.2.1 The challenge of screen content video quality enhancement in spatial domain

As mentioned before, Fig. 1.1(a) and Fig. 1.1(b) show two typical frames, one from a natural video, and the other from a screen content video. Screen Content Videos (SCVs) differ significantly from traditional video types, primarily featuring two types of content:

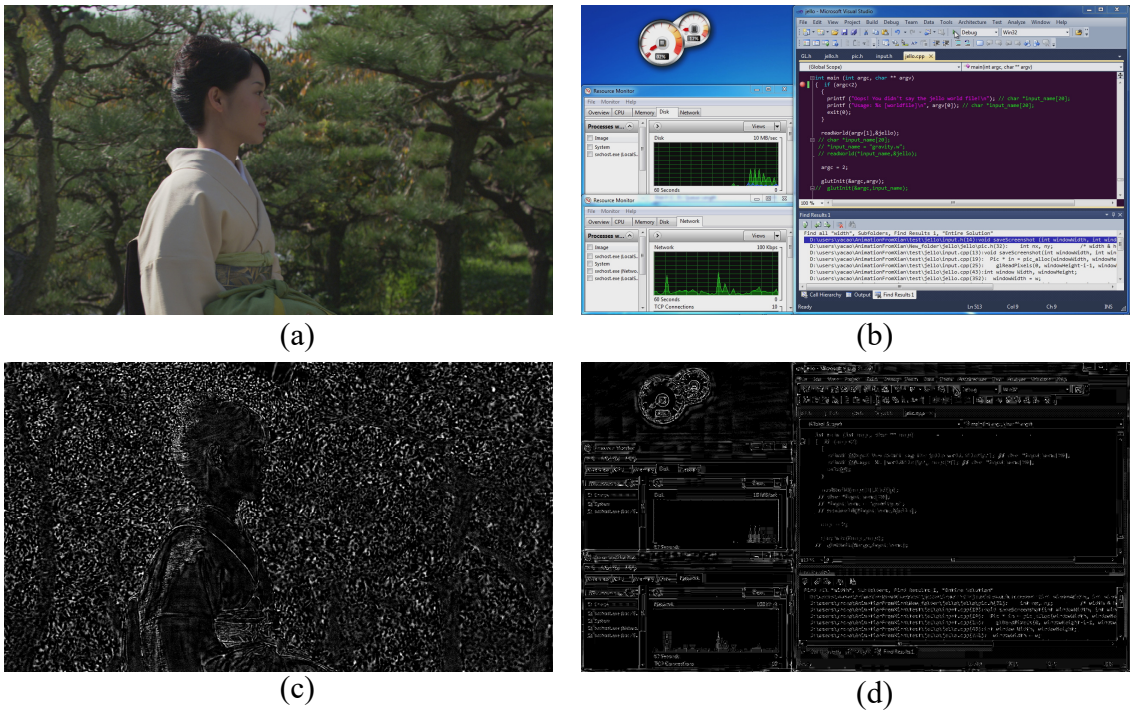


Figure 1.1: (a) Original natural frame, (b) original screen content frame, (c) artifact of natural frame, and (d) artifact of screen content frame.

text and graphics. Graphics in SCVs are characterized by smooth edges and complex textures, whereas text displays sharp edges and simpler textures, as shown in Fig. 1.1(b). Unlike natural videos, which typically consist of smooth backgrounds and rich colors, SCVs contain prominent text and graphic elements with limited pixel color variation. Traditional video enhancement methods fall short when applied to SCVs, as they do not address the high-frequency details such as text and tables effectively. These methods also fail to accommodate the distinct enhancement needs of graphical and textual content; for example, motion compensation techniques in video coding can create artifacts in graphics, while tools like IBC may distort the sharp edges in text. To illustrate this difference, Fig. 1.1(c) and Fig. 1.1(d) depict the artifact distributions in natural and screen content videos, respectively. The artifact distribution is obtained by calculating the difference between the reconstructed frame and the original frame. We can observe in Fig. 1.1(c) that the artifact appears throughout the entire area of the natural content due to its diverse range of colors and camera noise. In contrast, we can see in Fig. 1.1(d) that the artifact mainly occurs in the edge regions of screen content. This significant difference in

the distribution of residual image pixels between natural and screen content highlights the crucial role of edge information in screen content. This thesis will explore the development of a robust edge attention mechanism for target frames. This approach aims to guide the model in adaptively restoring edge information in target frames, thereby enhancing the overall performance of video quality enhancement.

1.2.2 The challenge of screen content video quality enhancement in temporal domain

In the temporal domain, screen content video often consists of static or moving texts and charts. We encode both natural sequence and screen content sequence, and then calculate the PSNR between the reconstructed frames and the original frames as shown in Fig. 1.2 and Fig. 1.3. Notably, the peak signal-to-noise ratio (PSNR) changes in natural video (Fig. 1.2) tend to exhibit relative stability across different frames, whereas the PSNR fluctuations in screen content videos exhibit significant variations. This discrepancy arises from the predominant use of intra prediction and skip mode in SCC to encode uniform, flat background regions with minimal distortion. When the content remains static across multiple frames, the PSNR of a compressed frame remains constant, as shown in shadow region of Fig. 1.3. We refer to this situation as "frame freezing", which is uncommon in natural video due to camera noise. Moreover, during activities like web browsing, abrupt content transitions in the next frame trigger "scene switch", a common occurrence in screen content videos. To identify scene switch frames, we have marked them with dashed lines in Fig. 1.3, the compressed video exhibits significant PSNR drops during scene switches leading to noticeable quality degradations that severely impact the Quality of Experience (QoE) [12]. Existing methods such as flow-based alignment [13, 14] and deformable-based alignment [15, 16] are inadequate for compensating the positions from the neighbor frames when substantial content variations between frames [17].The imprecision of the prediction network can diminish the performance of the quality enhancement network. In addition, the alignment-free method using a multi-frame approach described in [17] extracts the high-quality region from neighbor frames to enhance the target frame. However, during scene switches, the neighbor frames might provide irrelevant information to the quality

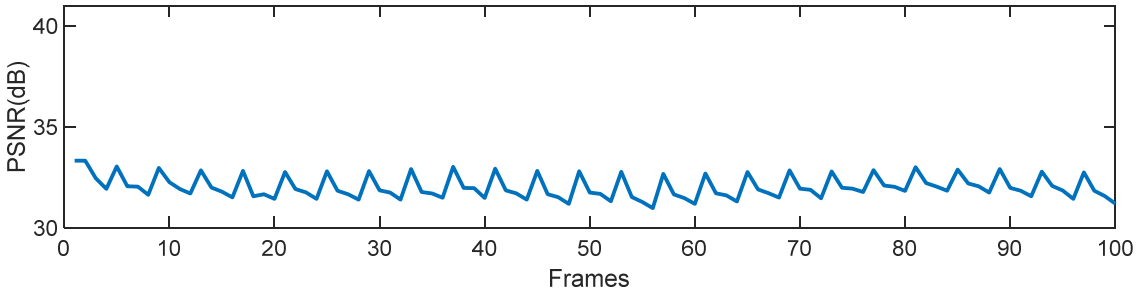


Figure 1.2: PSNR statistics for natural video *basketball*.

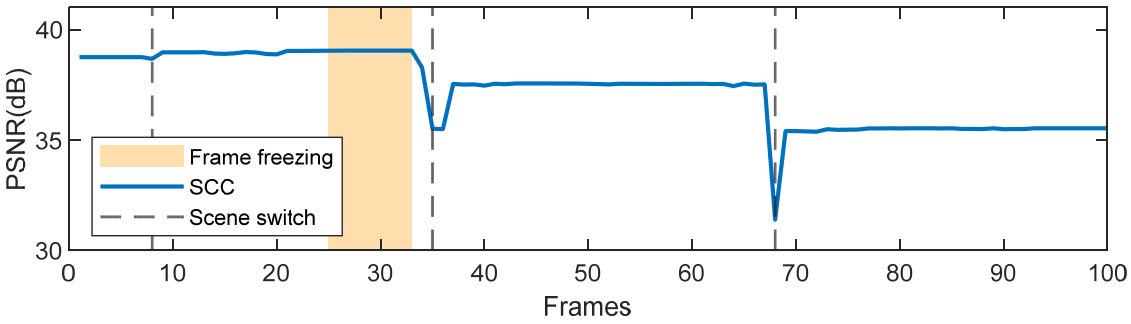


Figure 1.3: PSNR statistics for screen content video *scwebbrowsing*.

enhancement model. Moreover, in the case of frame freezing, the conventional multi-frame structure fails to effectively utilize neighbor frames for extracting valuable information to enhance the target frame. Consequently, this thesis aims to develop novel video quality enhancement methods to address the challenges posed by frame freezing and scene switches in screen content videos.

1.3 Contributions of This Thesis

This thesis introduces three models tailored for screen content video quality enhancement based on the unique characteristics of screen content video.

1.3.1 Mode Information Guided CNN for Quality Enhancement of Screen Content Coding

To distinguish natural content from screen content and enhance the quality of different content types effectively, our Mode Information Guided CNN (MICNN) uses coding modes as guidance. This work explores the relationship between content type and coding mode.

By integrating our proposed binary mode masks into a mode information guided deep network model, extracted SCC modes from the bitstream can be utilized to enhance the quality of SCVs. Specifically, the new branch uses the binary mode masks, which are based on the coding modes of SCC, to exploit the characteristics of SCC, and then guide the neural network for quality enhancement on screen content videos. This work pioneers the incorporation of SCC mode information into the branches to boost SCV quality. Experimental results show that our proposed MICNN is more effective than other networks. We believe that our mask branches can be easily adapted to different single-input models for further quality enhancement of SCC.

1.3.2 Spatio-temporal Feature Learning for Enhancing Video Quality Based on Screen Content Characteristics

To address the challenges of frame freezing and scene switches in screen content videos, this thesis proposes a new approach called the edge aware with spatio-temporal information fusion network (EAST). In our EAST approach, we design a spatio-temporal feature extraction module that can identify features associated with the target frame by extracting information from different groups of input frames. This design effectively addresses the challenges posed by scene switches in screen content videos. Based on the observations in Fig. 1.1, we also develop a novel edge aware block that focuses on extracting high-frequency information from the target frame. This block guides the model in restoring the high-frequency details of the enhanced frame in the spatial domain. To cater to frame freezing scenarios, the inclusion of edge information compensates for the lack of information from neighbor frames. To adaptively enhance target frames in scenarios involving scene switches and frame freezing, we introduce a novel channel and spatial attention block (CSAB), which consists of a channel attention module and a spatial attention module, in the spatio-temporal feature fusion. With the help of spatio-temporal information, this CSAB dynamically allocates attention to different scenarios, ensuring effective enhancement of the overall quality in screen content videos with scene switches and frame freezing. Experimental results demonstrate the significant advancements achieved by the proposed EAST in enhancing the quality of compressed videos, surpassing the current

state-of-the-art methods.

1.3.3 Long Short-term Fusion by Multi-scale Distillation for Screen Content Video Quality Enhancement

To develop a novel video quality enhancement method to address the challenges of scene switches and dramatic motions in screen content videos, this thesis also proposes a Long Short-term Fusion by Multi-scale Distillation (LSFMD) method to restore high-frequency details in compressed screen content videos, particularly improving quality during scene switches. The long short-term feature extraction is designed to capture the relevant features from neighbor frames to assist the model in handling quality enhancement during the scene switches and fast motion. In the short-term feature extraction stream, we introduce a Similarity-based Neighbor Frame Selector (SNFS) that identifies and selects relevant frames among neighbor frames, minimizing disturbances from unrelated frames. This selector ensures that short-term information is extracted from frames with similar content, enhancing the accuracy of the reconstruction. To further improve the quality of the target frame and effectively fuse short-term and long-term information, we design a Multi-scale Hierarchical Feature Distillation (MHFD) mechanism. This mechanism transforms features from different scales and uses local-global attention to distill significant features pertinent to the target frame. In the reconstruction phase, unlike conventional methods by incorporating a High-Frequency Reconstruction Block (HFRB), our proposed LSFMD uses high-frequency information to guide the model in restoring fine details of the target frame. This approach ensures the preservation and enhancement of critical high-frequency details, resulting in better video quality. The experimental findings showcase the substantial progress made by our proposed LSFMD technique in elevating the quality of compressed screen content videos, outperforming existing state-of-the-art methods.

1.4 Organization of This Thesis

The rest of this thesis is organized as follows:

Chapter 2: This chapter briefly reviews related work in the literature from four aspects: single-frame quality enhancement, multi-frame quality enhancement, additional information guided CNN, and attention mechanism, which are relevant to the research

work introduced in this thesis.

Chapter 3: This chapter presents a MICNN to further improve the quality of screen content sequences at the decoder side, along with the proposal of a new dataset.

Chapter 4: This chapter proposes a new approach called the EAST to address the aforementioned challenges of frame freezing and scene switches in screen content videos.

Chapter 5: This chapter proposes an LSFMD method to effectively restore high-frequency details and improve quality during scene switches in compressed screen content videos.

Chapter 2

Literature Review

2.1 Single-frame Quality Enhancement

Single-frame video enhancement task is equivalent to image enhancement. Earlier works mainly focus on the quality enhancement of JPEG compressed images. For example, the AR-CNN [18] was one of the pioneering models to employ a convolutional neural network for image enhancement, effectively learning the nonlinear mapping between original and compressed images using a structure composed of four convolutional layers. Following this, [19] introduced a deeper CNN architecture for image deblocking, achieving substantial improvements over shallower CNN models in terms of both visual and objective quality metrics. The introduction of batch normalization and residual learning led to the development of DnCNN [20], which effectively addressed gradient vanishing issues in deep image enhancement networks. Additionally, a novel dual pixel-wavelet domain deep CNN-based soft decoding network for JPEG compressed images, known as DPW-SDNet [21], was developed. This network utilizes four downsampled versions of a compressed image to create a 4-channel input, thereby enhancing the output with more accurate pixel domain predictions. Galteri et al. [22] introduced a conditional Generative Adversarial Network (GAN) framework, where they innovatively train the model to alternate between generating full-size patches and performing sub-patch discrimination. Similarly, Guo et al. [23] developed a one-to-many network that assesses output quality through a combination of perceptual loss, naturalness loss, and JPEG loss, aiming to optimize various aspects of image fidelity. Furthermore, Liu et al. [24] pioneered the integration of non-local operations

into a recurrent neural network (RNN) through their non-local recurrent network (NLRN). Zhang et al. [25] introduced a residual non-local attention mechanism designed to capture long-range dependencies between pixels, enhancing the contextual understanding of the network. Additionally, the Memory Network (MemNet) [26] was developed for tasks in image restoration, including quality enhancement. This network features a unique memory block that creates long-term memory across CNN layers, effectively restoring middle- and high-frequency signals that are often distorted during the compression process.

There are also some other works proposed for the quality enhancement of compressed video using spatial information from a single frame. For instance, the IFCNN [7] replaced the conventional SAO filter with a three-layer CNN module to improve video quality within the codec. Similarly, the Variable-filter-size Residue-learning CNN (VRCNN) [8] aimed to reduce distortion in videos by modifying internal codec modules. Other approaches focus on post-processing techniques to enhance video quality after decoding. For instance, the DCAD [9] employs ten convolutional layers to utilize spatial information and improve videos at the decoder side. The Quality Enhancement CNN (QE-CNN) [27] was designed to enhance both I and P/B frames, effectively addressing intra- and inter-coding quantization distortion. Additionally, the work in [28] proposed using partition information to boost video quality. Moreover, Yang et al. [11] proposed the DS-CNN approach for video quality enhancement. In [11], DS-CNN-I and DSCNN-B, are used to reduce the artifacts of intra- and inter-coding, as two subnetworks of DS-CNN, respectively. However, these approaches primarily consider spatial information, overlooking the crucial role of temporal information in video quality enhancement.

2.2 Multi-frame Quality Enhancement

In contrast, the Multi-Frame Quality Enhancement (MFQE 1.0) approach [13] by Yang *et al.* has evolved to utilize temporal information for enhancing video quality. This method uses high-quality frames from compressed video as reference frames to improve the quality of low-quality target frames through a Multi-frame CNN. Building upon this, an updated version, MFQE 2.0 [14], was developed to improve efficiency and achieve better performance. These methods employ dense optical flow for motion compensation to aggregate

information from both target and reference frames. However, optical flow alignment is unsuitable for screen content video quality enhancement, as scene switches can disrupt the pixel-wise correspondence between frames, leading to inaccurate optical flow estimation. In addition to flow-based methods, deformable convolution-based methods have been proposed to learn offsets from compressed frames, obtaining aligned features for VQE. An alternative work proposed in [15] is the deformable-based alignment Spatio-Temporal Deformable Fusion (STDF) approach, which adaptively compensates for sampling positions of frames, capturing the most relevant context and removing artifacts in the target frame. The Spatio-Temporal Detail Information Retrieval (STDR) network in [29] incorporates a multi-path deformable alignment module to enhance the accuracy of offset generation by integrating alignment features from various receptive fields. In a related development, a new end-to-end network, termed Coarse-to-Fine Spatio-Temporal Information Fusion (CF-STIF) [30], has been proposed for enhancing the quality of compressed videos. This network advances the field by predicting more precise offsets, aided by its capability to utilize a larger receptive field. Besides, in the natural video, the motion vector can be utilized to guide the enhancement process in Coding Priors-Guided Aggregation Network (CPGA) [31]. Based on STDF, Recursive Fusion and Deformable Spatiotemporal Attention (RFDA) method [32] introduces a recursive fusion module that not only utilizes reference frames within the current time window but also capitalizes on the temporal information from previously enhanced video frames. This approach effectively expands the time window implicitly, allowing RFDA to harness a broader range of temporal data for improved spatio-temporal compensation. However, it is important to note that this recursive fusion module significantly increases computational complexity. A plug-and-play module called Spatio-Temporal Information Balance (STIB) [33] is proposed to refine the aligned reference frame by spatial attention mechanism, in order to remove the noise generated by temporal alignment. While flow-based, deformable-based, and motion vector-based alignments have primarily been proposed for natural video quality enhancement, they may not effectively compensate for the position of the target frame in screen content videos. To enhance screen content videos, the Content Adaptive Network based on Two Branches (CAT) [16] was proposed to perform specific enhancements on text and graphics

separately. Another method, Spatial-Temporal Adaptive (STA) [34], introduced a dual-branch structure for parallel single-frame and multi-frame feature extraction to enhance screen content videos. However, these approaches utilizing deformable convolution may potentially reduce the accuracy of compensating the target frame's position, which reduces their efficiency and practicality. The Quality Enhancement Network using Cross-Frame Information (QECF) [17] introduced a cross-fusion block instead of an alignment-based method. However, QECF was specifically developed for gaming videos, which consist of a series of consistent frames. Meng et al. [35] developed the Multi-frame Guided Attention Network (MGANet), which incorporates a bidirectional convolutional LSTM following a flow-guided motion compensation operation to align multiple input frames effectively. Similarly, FastMSDD [36] employs a multi-scale 3D CNN that delves into multiscale similarities among video frames to improve the quality of HEVC compressed video. However, these LSTM-based methods, as noted by [37], overlay temporal information onto spatial information, thereby failing to concurrently leverage both types of data effectively. A temporal group alignment and fusion network (TGAF) [38] was proposed for the quality enhancement of compressed videos by selecting the frames from the video to form a group of pictures according to the temporal distances to the target frame. However, the skipping selection adds irrelated frames when the scene switch occurs. To address the unique characteristics of screen content videos, which often involve dramatic motion and scene switches, this thesis will propose a novel network that overcomes the limitations of existing multi-frame approaches. This new method is designed to handle the specific challenges posed by screen content videos, ensuring more accurate and effective video quality enhancement.

2.3 Additional Information Guided CNN

At present, some researches [39–42] not only focus on main-stream processing but also introduce the multi-branch architecture to improve the performance of the network. Simonyan et al. [43] initially introduced a two-stream CNN architecture, which processes RGB frames and stacked optical-flow images to recognize activities. Feichtenhofer et al. [44, 45] explored various strategies for fusing predictions from appearance and motion

streams to boost recognition capabilities. Further advancements include the Trajectory-pooled Deep-convolutional Descriptor (TDD) [46], which combines the strengths of two-stream ConvNets with handcrafted features. Peng et al. [47] developed a spatial-temporal attention-based two-stream architecture that collaboratively learns the interplay between static and motion information. Wang et al. [48] introduced the Temporal Segment Network (TSN), which samples snippets sparsely across the entire video to capture long-range temporal dependencies through two-stream features. Zhang et al. [49] incorporated video super-resolution (SR) techniques into the two-stream architecture to enhance low-resolution video activity recognition. Recently, novel multi-branch networks have been designed using the two-stream concept, such as ARTNet [50] and STM [51], which incorporate motion-excited modules, and others like the Nonlocal block [52], which models long-term relations contextually, and SlowFastNet [53], which utilizes RGB inputs sampled at varying frame rates to capture diverse representations.

Besides, a partition-aware convolution neural network was proposed in [28], which uses the partition information produced by the encoder to assist post-processing at the decoder side. Inspired by He et al. [54], another dual-input model proposed by Hoang and Zhou [55], a Deep Recursive Residual Network with Block information (B-DRRN), also employs the mean mask as side information. Sun et al. [56] propose a video quality enhancement framework that combines the distribution information of HEVC compression noise. In contrast, this thesis will design a novel multi-input CNN that utilizes decoded frames with the mode information of SCC as the input. The idea is to utilize three binary masks, including the information of IBC mode, PLT mode, and INTRA mode to further enhance the quality of screen content videos.

Edge information is critical in numerous image processing applications, such as image super-resolution (SR). Consequently, several methods have introduced edge information as an auxiliary branch to enhance the performance of the main network. For instance, Fang et al. [39] developed a soft-edge guided CNN for single-image super-resolution, incorporating separate edge and reconstruction branches to facilitate the fusion of spatial details. Additionally, a novel edge-assisted mechanism [57] has been recently proposed for image SR, leveraging edge information in the gradient domain to guide the feature

learning process. The effectiveness and necessity of edge-assisted or edge-guided image processing methods have been demonstrated by various studies [58, 59]. However, the implementation of these methods can be complex and comes with certain limitations. For example, Yang et al. [42] incorporated image edges into a CNN model and introduced the Edge Guided Recurrent Residual (DEGREE) Network. Despite its innovative approach, DEGREE has some drawbacks, such as directly adding the learned image edge features to the low-resolution (LR) image to produce the final SR image, which essentially relies on residual learning. This method does not fully exploit the potential of image edge information. In response to these challenges, this thesis is going to propose two edge-aware modules designed to maximize the utilization of edge information. These modules can guide the CNN in restoring high-frequency information in the target frame.

2.4 Attention Mechanism

Over the past decade, attention mechanisms have become increasingly significant in the field of computer vision, enhancing performance across a wide range of visual tasks. These tasks include image classification [60, 61], object detection [62, 63], semantic segmentation [64, 65], face recognition [66, 67], person re-identification [68, 69], action recognition [52, 70], few-shot learning [71, 72], medical image processing [73, 74], image generation [75, 76], pose estimation [77], and super-resolution [78, 79]. The initial breakthrough in integrating attention with deep neural networks was the Recurrent Attention Model (RAM) [80]. RAM used a policy gradient approach to recurrently predict important regions and update the network in an end-to-end manner. Following this, various subsequent studies [75, 81] employed similar attention-based approaches, often utilizing RNN as essential components to implement the attention mechanism effectively. Jaderberg et al. [82] introduced the Spatial Transformer Network (STN), which features a sub-network designed to predict an affine transformation for selecting important regions within the input image. The subsequent phase is exemplified by Deformable Convolutional Networks (DCNs) [62, 83], which represent a pivotal development in predicting discriminatory features directly through the network architecture. The evolution continued with the introduction of SENet [60], which pioneered a channel-attention mechanism that adaptively

and implicitly identifies key features in the data. This phase also includes further developments such as Convolutional Block Attention Module (CBAM) [61] and ECANet [84], which enhance the model's focus on relevant features through refined attention strategies. The most recent phase in the development of attention mechanisms is characterized by the adoption of self-attention, initially proposed in natural language processing by Vaswani et al. [85]. This approach has been transformative, as evidenced by its applications in models like BERT [86]. In computer vision, Wang et al. [52] were pioneers in adapting self-attention to the field, introducing a non-local network that has achieved significant success in video understanding and object detection, marking the beginning of what might be considered the self-attention era in computer vision.

The integration of differentiable attention mechanisms into deep learning networks is designed to prioritize relevant parts of the input, such as channels [60], regions [61], or frames [87], by assigning them higher weights compared to less relevant elements. Class Activation Maps (CAMs) [88] exemplify a classical top-down attention mechanism extensively utilized in weakly supervised tasks, including weakly supervised semantic segmentation (WSSS). However, CAMs tend to focus on the most discriminative parts of an image and can suppress other regions, resulting in the potential loss of important details within the regions of interest. To address this limitation, Huang et al. [89] developed a location-aware graph that models the interaction relationships between objects in a video. This approach uses an attention mechanism to effectively blend visual and question representations, enhancing the performance of video question answering by ensuring a more comprehensive understanding of both the visual content and the query context. Miyanishi et al. [90] introduced a two-stream compositional attention network designed to simultaneously capture the appearance and motion features of various entities. This network leverages question features as guidance for the attention mechanism, using them to recurrently infer answers from video content. Meanwhile, Ji et al. [91] developed a method to decompose actions into spatio-temporal graphs, explicitly analyzing action-object interactions. These graph features are then integrated with 3D CNN features through a non-local attention mechanism to enhance action recognition capabilities. However, these approaches depend on the availability of object detectors and annotations, or even more detailed rela-

tionship annotations [91], to identify and model the interactions between different entities accurately. In a related development, Lu et al. [92] proposed a spatio-temporal attention module (STAM), which they incorporated into a 3D-CNN-based two-stream architecture tailored for egocentric action recognition. Additionally, Li et al. [93] utilized an attention module guided by human segmentation to better learn action representations of individuals involved in interactions. They further advanced this approach by introducing a relational modeling method specifically aimed at recognizing human-human interactions. Zhang et al. [94] proposed an element-wise-attention-gate (EleAttG) based on RNN cells, to assign different levels of importance to each input frame. EleAttG is based on a bottom-up attention mechanism, so that the weights are fixed after training, thus resulting in generating attention independent of the input in the inference phase. To overcome the limitations of Ego-RNN and EleAttG, long short-term attention (LSTA) [95] was proposed by simultaneously exploiting a top-down scheme and sequential modeling. Wang et al. [96] employed a third-party detector to obtain position-aware object features as privileged information, and aggregated them into an attention module. Sudhakaran et al. [97] proposed the Ego-RNN model, by introducing the classical CAMs into a ConvLSTM network for egocentric activity recognition.

This thesis adopts attention mechanisms to improve the performance of deep learning-based methods on different quality enhancement tasks. In Chapter 4, a novel edge aware block is introduced, specifically designed to extract high-frequency information from the target frame. This block plays a crucial role in guiding the model to restore high-frequency details of the enhanced frame in the spatial domain. Then, the introduction of CSAB aims to dynamically allocate attention across different scenarios, ensuring effective enhancement of the overall quality in screen content videos with scene switches and frame freezing. Moving on to Chapter 5, an MHFD mechanism is proposed. This mechanism uses local-global attention to distill significant features to the target frame, enabling the capture of more information to address uneven noise distribution commonly found in screen content videos.

Chapter 3

Mode Information Guided CNN for Quality Enhancement of Screen Content Coding

3.1 Proposed Mode Information Guided CNN (MICNN)

3.1.1 Motivation

In recent years, deep learning has made new progress in this field and has achieved impressive performance in video enhancement. A series of neural network architectures [7–11] were proposed to remove the artifacts in reconstructed videos. In addition to the development of network structures, the rich side information in the video bitstream can also help to guide the enhancement process of decoded videos. For example, it was found in [54] that the partition tree in the coding process indicates the corresponding compression loss of the decoded video. To utilize the side information in the HEVC codec, the work in [54] subsequently proposed a double-input network by taking the partition mask into account. The mask is generated based on the partition tree of HEVC, as the side information. Owing to the block dividing process and quantization in HEVC, the artifact of decoded video corresponds highly to the CU information. Because of that, there are some important clues contained in CU information that can be used to eliminate the artifact of decoded videos. The works in [54] and [28] have proven that using the mean mask or the boundary mask can achieve better performance in the post-processing method. However, these

models were designed for natural videos. It still ignores the characteristics of the screen content video. In other words, the utilization of side information is not closely related to the screen content characteristics.

Screen content videos have different characteristics to natural videos, they often contain many uniform and flat areas, repeated patterns, and limited pixel colors. CU information cannot represent these characteristics. Therefore, various mechanisms of video quality enhancement are required for these different types of content. To identify natural content and screen content such that our MICNN can effectively enhance the reconstruction quality of different contents, it can be guided by the coding mode. Fig. 3.1 explains the relationship between content type and coding mode. Fig. 3.1(a) shows a frame with mixed content, and Fig. 3.1(b) illustrates the associated coding modes, highlighted by different colors. As shown in Fig. 3.1(b), red, yellow, and blue boxes are used to represent INTRA, PLT, and IBC, respectively. Compared with generic partition-based masks used in prior work (e.g., CU boundaries or CU-mean maps), SCC mode masks encode semantically richer, tool-specific content. Partition data only indicates block sizes and splits, which correlate loosely with compression strength but not with the underlying cause of distortions. In contrast, the IBC, PLT, and INTRA mode labels directly reveal the prediction mechanism applied to each CU. INTRA is known to encode natural content. IBC and PLT are designed for screen content: (1) IBC can find almost exact matching for certain CUs within the same frame due to the massive existence of texts and computer-generated graphics, and (2) PLT can well handle the CUs with only a few distinct colors. Therefore, the coding modes embedded in the coded bitstream are good candidates for identifying CU content types that can be used to guide the video quality enhancement in screen content videos. In the following section, we propose to use three binary mode masks devised by different coding modes, IBC, INTRA, and PLT, in our new MICNN to improve the visual quality of screen content. Through the input of mode information, MICNN can eliminate different artifacts of decoded screen content videos according to the content encoded by different coding modes.

In other words, this chapter proposes a novel post-processing network for enhancing decoded screen content videos based on the coding mode information embedded in the

coded bitstream. Three binary mode masks derived from the dedicated coding tools in SCC are fused with the corresponding decoded frame. Besides, for the dataset limitation, we establish a large-scale dataset containing 9810 frames for screen content videos.

3.1.2 Binary Mode Mask

Based on the above motivation, three binary mode masks, M_{IBC}, M_{PLT}, and M_{INTRA} are defined based on different coding modes—IBC, PLT, and INTRA, respectively. They are used for different types of content, resulting in different artifacts in the decoded SC video.

Suppose $Mode(CU\,(x,y))$ is the coding mode in which the pixel location (x,y) belongs to a particular CU, and $M_{mode}(x,y)$ is the binary value of the element at (x,y), where $mode \in IBC, PLT,\ INTRA$. $M_{mode}(x,y)$ is set to 1 when (x,y) belongs to the CU encoded as $mode \in IBC, PLT,\ INTRA$. Otherwise, $M_{mode}(x,y)$ is filled with the value of 0. Then, the binary values of the elements of M_{IBC}, M_{PLT}, and M_{INTRA} can be generated as follows:

$$M_{IBC}\,(x,y) = \begin{cases} 1, & if\ Mode(CU(x,y))\epsilon IBC \\ 0, & if\ Mode(CU(x,y)) \notin IBC \end{cases} \tag{3.1}$$

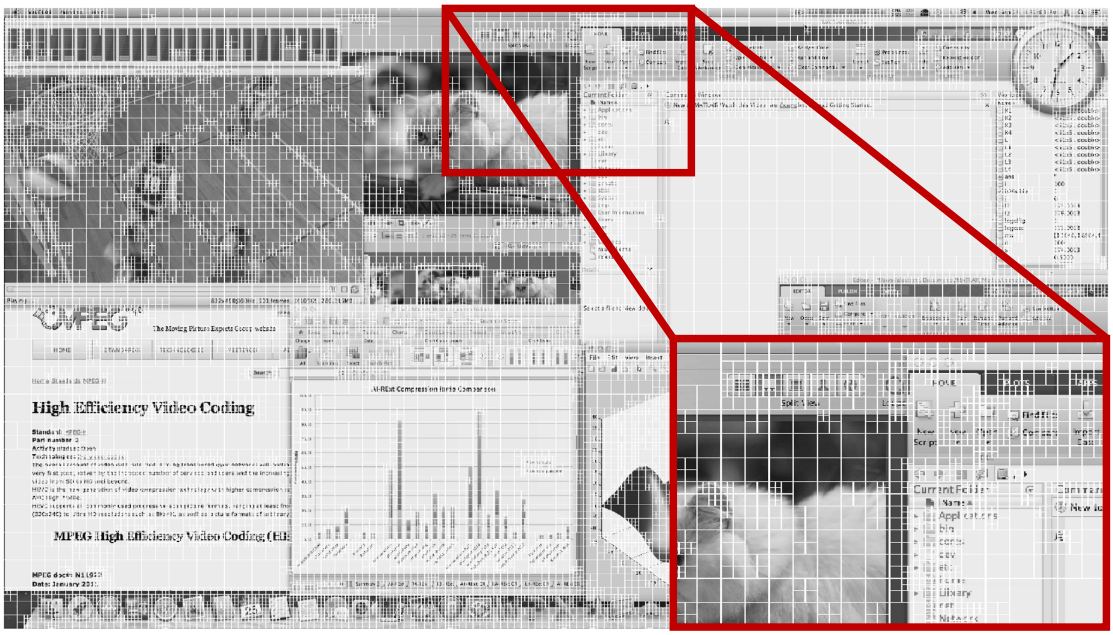
$$M_{PLT}\,(x,y) = \begin{cases} 1, & if\ Mode(CU(x,y))\epsilon PLT \\ 0, & if\ Mode(CU(x,y)) \notin PLT \end{cases} \tag{3.2}$$

$$M_{INTRA}\,(x,y) = \begin{cases} 1, & if\ Mode(CU(x,y))\epsilon INTRA \\ 0, & if\ Mode(CU(x,y)) \notin INTRA \end{cases} \tag{3.3}$$

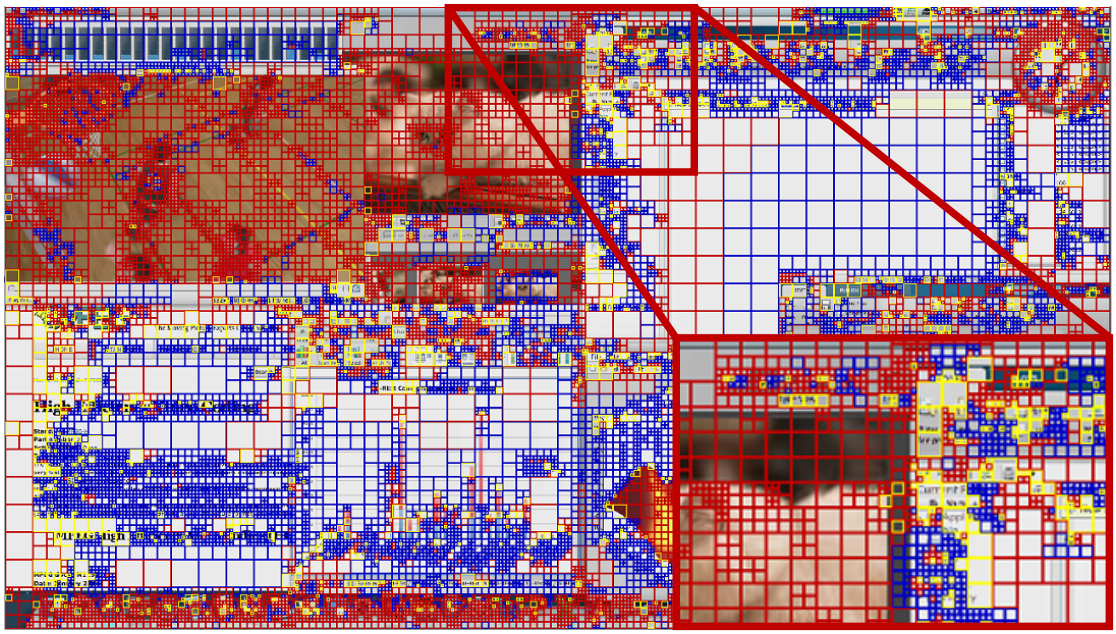
Fig. 3.2 shows the examples of the IBC binary mode mask, PLT binary mode mask, and INTRA binary mode mask based on the assigned values using Eq.(3.1), Eq.(3.2), and Eq.(3.3).

3.1.3 Model Structure

The baseline CNN architecture is shown in Fig. 3.3(a), where our proposed MICNN is adopted. The MICNN architecture consists of three components, i.e., feature extraction, feature fusion, and reconstruction. In the feature extraction stage, one main branch and



(a)



(b)

Figure 3.1: (a) Original frame, and (b) associated coding modes (red: INTRA, yellow: PLT, blue: IBC).

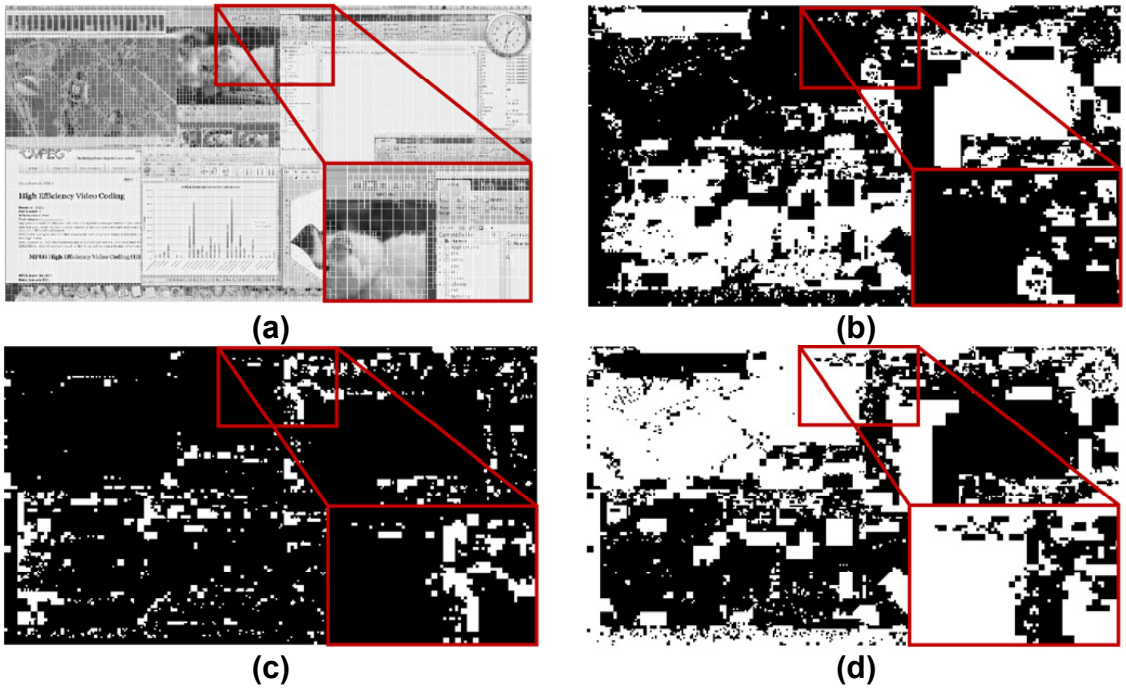


Figure 3.2: Examples of three binary mode masks. (a) Original frame with CU partition, (b) IBC binary mask, (c) PLT binary mask, and (d) INTRA binary mask.

three sub-branches are used to extract features. The decoded frame is fed into CNN through the main branch and the binary mode masks M_{IBC}, M_{PLT}, and M_{INTRA} are the inputs of the three sub-branches.

The binary mode masks are the side information. They are fed into the neural network and combined with the decoded frame. Therefore, the order of the three binary mode masks fused in the neural network are considered, and ablation study related to various orders will be made later. From Fig. 3.3(b), we can see the detail of our proposed fusion method. The features extracted from different binary mode masks will be added to the feature extracted from decoded frame in order.

Moreover, Residual Dense Blocks (RDBs) represented in Fig. 3.3(c) are stacked as the main branch of the proposed MICNN. As shown in Fig. 3.3(c), the RDB contains three groups of convolutional layers that are in dense connection [98]. Each group consists of two convolutional layers with a size of 3×3 and two ReLU activation functions. Meanwhile, the residual connection in each RDB is employed to reduce the gradient vanishing problem and help the backpropagation. Compared with the original residual block as shown in Fig.

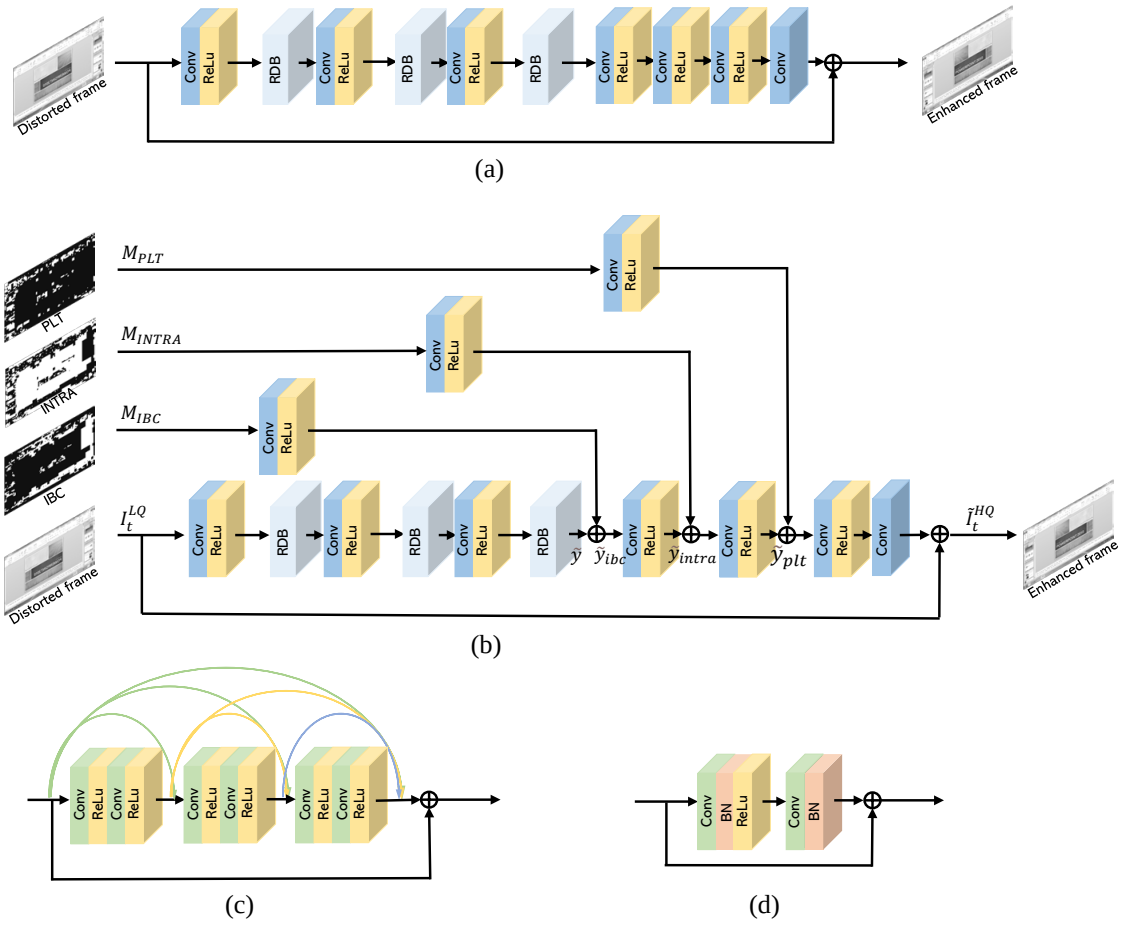


Figure 3.3: (a) The baseline CNN structure without binary mode masks, (b) the proposed MICNN structure, (c) Residual Dense Block (RDB), and (d) Traditional Residual Block.

3.3(d), RDB uses dense connection which can exploit hierarchical features.

To formulate the MICNN model proposed in Fig. 3.3(b), we denote a low-quality frame at time t as $I_t^{LQ} \in \mathbb{R}^{H \times W}$, where H and W indicate the vertical and horizontal resolutions of the frame. The enhanced frame is represented by $\tilde{I}_t^{HQ} \in \mathbb{R}^{H \times W}$.

The composite non-linear mapping including convolutional operation and activation function (ReLU) is denoted as $H_{cr}(\cdot)$. In addition, the RDB is denoted as $H_{RDB}(\cdot)$. The output of the main branch in the feature extraction stage can then be obtained by

$$\widetilde{y} = H_{RDB}(H_{cr}(H_{RDB}(H_{cr}(H_{RDB}(H_{cr}(I_t^{LQ})))))) \tag{3.4}$$

The output of the sub-branches in the feature extraction stage can be formulated as:

$$\widetilde{m}_{ibc} = H_{cr}(M_{IBC}) \tag{3.5}$$

$$\widetilde{m}_{intra} = H_{cr}(M_{INTRA}) \tag{3.6}$$

$$\widetilde{m}_{plt} = H_{cr}(M_{PLT}) \tag{3.7}$$

where $\widetilde{m}_{ibc}$, $\widetilde{m}_{intra}$, and $\widetilde{m}_{plt}$ are defined as the feature maps of the IBC mode mask, INTRA mode mask, and PLT mode mask, respectively. These feature maps are then integrated into the main branch in the feature fusion stage, which can be formulated as:

$$\widetilde{y}_{ibc} = \widetilde{y} + \widetilde{m}_{ibc} \tag{3.8}$$

$$\widetilde{y}_{intra} = H_{cr}(\widetilde{y}_{ibc}) + \widetilde{m}_{intra} \tag{3.9}$$

$$\widetilde{y}_{plt} = H_{cr}(\widetilde{y}_{intra}) + \widetilde{m}_{plt} \tag{3.10}$$

where $\widetilde{y}_{ibc}$, $\widetilde{y}_{intra}$, and $\widetilde{y}_{plt}$ denote the output after adding the IBC mode mask, the INTRA mode mask, and the PLT mode mask in order, respectively. Finally, the reconstructed frame can be generated as:

$$\tilde{I}_t^{HQ} = H_c(H_{cr}(\widetilde{y}_{plt})) + I_t^{LQ} \tag{3.11}$$

where $H_c(\cdot)$ denotes the convolutional operation.

The proposed network is trained in an end-to-end manner. To optimize our model, we apply Mean Squared Error (MSE) as the loss function. The loss function L is represented as:

$$L = \sqrt{\left\| I_t^{HQ} - \tilde{I}_t^{HQ} \right\|^2} \tag{3.12}$$

where I_t^{HQ} is the ground truth frame at time t, $\tilde{I}_t^{HQ}$, represents the enhanced frame generated by our model.

3.2 Proposed PolyUSCC Dataset

The work of this chapter mainly focuses on video quality enhancement of SC sequences. However, the number of SC sequences is limited. To avoid overlapping with the sequences provided in the Common Test Condition (CTC) [99], SC sequences were gathered from other sources [100], [101], or self-capture [102] to form our dataset, "PolyUSCC". Thirty-four HEVC standard video sequences of various resolutions HEVC standard video sequences of various resolutions (1920×1080, 1680×1050, 1280×720) are adopted, as shown in Table 3.1. These sequences can be divided into three types: text and graphics with motion (TGM), animation (A), and mixed (M) content. The mixed content contains natural content and screen content. The text and graphics with motion (TGM) consists of text, graphic and animation. The animation (A) only contains the gaming content. To make the database focusing on the different types of screen content, the number of TGM sequences is twice the amount of the mixed content. The dataset consists of three parts. First, to guarantee data reliability and availability, half of the dataset (15 sequences) are provided from the JCT-VC [103] but not included in CTC. Second, there are 5 SC sequences from Tsang *et al.* [101]. To enrich the text and graphics with motion content and mixed content sequences, we further capture 14 video sequences by ourselves. During the evaluation of the proposed MICNN, 27 sequences are used for training and the remaining 7 sequences are used for validation, as shown in Table 3.1.

Table 3.1: Dataset

Dataset	Ref.	Sequence	Frame	Type	Resolution
Training set	[102]	airplanevideocmd	300	M	1920×1080
		consolecmd	300	TGM	1920×1080
		consoledocument	300	TGM	1920×1080
		consolenew	300	TGM	1920×1080
		docgooglemap	300	TGM	1920×1080
		docvideoplanets	300	M	1920×1080
		googlemap	300	TGM	1920×1080
		purecmd	300	TGM	1920×1080
		sccconsolecmdcpu	300	TGM	1920×1080
		cmd3	300	TGM	1920×1080
		PolyuEIEweb1	100	M	1920×1080
		Polyuwebcmdvideo2	100	M	1920×1080
		Polyuwebvideo1	100	M	1920×1080
	[101]	MsStore	100	M	1680×1050
		NewsBrowse	100	M	1680×1050
		PaperPdf	100	TGM	1680×1050
		VisualStudio	100	M	1680×1050
	[100]	BitstreamAnalyzer	300	TGM	1920×1080
		ChineseDocumentEditing	300	TGM	1920×1080
		CircuitLayoutPresentation	300	TGM	1920×1080
		ClearTypeSpreadsheet	300	TGM	1920×1080
		scWeb	500	TGM	1920×1080
		sccadwaveform	200	TGM	1920×1080
		scdoc	500	TGM	1920×1080
		scpcblayout	200	TGM	1920×1080
		scpptdocxls	200	TGM	1920×1080
		scvideoconferencingdocharing	300	TGM	1920×1080
Validation set	[100]	BigBuck	404	TGM	1920×1080
		EnglishDocumentEditing	300	TGM	1920×1080
		KimonoError1	1006	M	2560×1440
		MissionControlClip1	600	M	2560×1440
		scviking	300	A	1280×720
	[101]	YouTube	100	M	1680×1050
	[102]	consolenew2	300	TGM	1920×1080

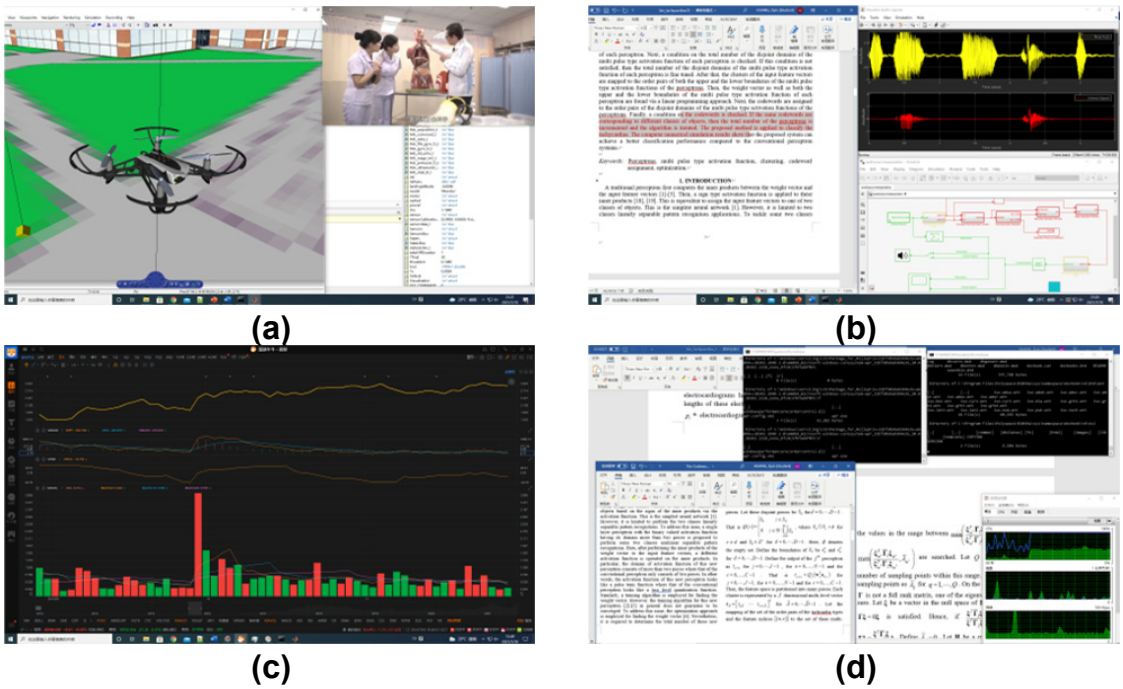


Figure 3.4: Examples of self-captured sequences. (a) airplanevideocmd, (b) consoledocument, (c) consolenew, and (d) cmd3.

3.3 Experiments and Analysis

3.3.1 Implementation Details

Training of MICNN requires a dataset of training examples, which are pairs of inputs and the corresponding outputs. The video sequences in PolyUSCC were encoded by the HEVC reference software HM16.20-SCM8.8 [3] under All-Intra (AI) configuration as the input of networks, while the uncompressed raw video sequences were used as the output of networks. Considering that different Quantization Parameters (QPs) in HEVC have different compression results with varying degrees of artifacts, four different QPs of 22, 27, 32, and 37 were set to ensure that the results of the experiments are more representative. One model was trained for one QP. For each frame, only the luminance channel (Y channel) was considered as input for training. To exploit SCC side information, we generate three per-frame binary mode masks at the decoder by instrumenting HM16.20-SCM8.8. Specifically, after the mode decision for each coding unit (CU), we write its chosen mode into three preallocated mask buffers that share the frame's spatial resolution. For the IBC mask, all pixels within a CU decoded in IBC mode are set to 1 (others 0).

Likewise, for the PLT mask, CUs decoded in PLT mode are marked 1 (others 0). For the INTRA mask, CUs decoded with conventional HEVC intra prediction are marked 1 (others 0). Therefore, for each pixel, we have a one-hot vector representing the coding mode. This produces three aligned binary matrices named MIBC, MPLT, and MINTRA, which are synchronized with the decoded Y frame and emitted alongside it. These masks capture the SCC mode partitioning directly from the bitstream without re-encoding or re-analysis, ensuring accurate guidance for the MICNN during training and inference. Model construction and training were based on PyTorch. The patch size of each input image and its corresponding ground truth were 64×64. We randomly selected one patch from one frame for each iteration. To guarantee the robustness of our dataset, we select all frames in our training process. In our experiments, the learning rate was set to 0.0001 for QP37. We fine-tuned the learning rate as QP decreases. The adaptive moment estimation (Adam) optimization method was used to train the model for 500 epochs. A computer equipped with Windows 10 operating system, Intel i9-10900K CPU, 64 GB RAM, and NVIDIA 3090Ti GPUs was used to perform the model training.

The test set contains 12 video sequences provided in the CTC [99], none of which is the same as the training set and validation set. This is essential to avoid overfitting issue.

3.3.2 Objective Visual Quality Assessment

In this section, we compare QECNN [27], DCAD [9], Partition-aware CNN [28], and QECF [17] with our proposed MICNN. Table 3.2 and Table 3.3 show the average PSNR improvement (ΔPSNR) and the average SSIM improvement (ΔSSIM), respectively, over all frames of each test sequence. In these two tables, the best ΔPSNR/ΔSSIM is highlighted in bold and the underline number is the second-best ΔPSNR/ΔSSIM. We can see that our proposed baseline and MICNN outperform other methods in most cases. Meanwhile the proposed MICNN achieves better performance than the proposed single input model. It demonstrates the benefit of using our proposed SCC mode masks.

When QP is 37, the highest ΔPSNR of our MICNN approach reaches 1.20 dB, i.e., for sequence *scwebbrowsing*. The average PSNR of our MICNN approach is 0.58 dB, which is 0.03dB higher than that of our baseline model (0.55 dB), 0.41dB higher than that of QECF (0.17 dB), 0.18dB higher than that of Partition-aware CNN (0.40 dB), 0.14dB higher than

Table 3.2: Overall ΔPSNR of Different Models at QP=22,27,32,37

QP	Sequence	QECNN	DCAD	Partition-aware CNN	QECF	baseline	MICNN
37	BasketballScreen	0.31	0.40	0.38	0.38	0.45	**0.46**
	ChineseEditing	0.21	0.37	0.37	0.22	**0.48**	0.43
	MissionControlClip2	0.31	0.42	0.44	0.37	0.48	**0.55**
	MissionControlClip3	0.35	0.45	0.48	0.43	0.49	**0.58**
	scconsole	-0.31	-0.10	-0.47	-0.17	0.01	**0.17**
	scdesktop	0.27	0.54	0.39	-0.03	**0.79**	0.76
	scflyingGraphics	0.48	0.66	0.69	-0.43	0.81	**0.88**
	scmap	0.20	0.33	0.37	0.19	0.40	**0.42**
	scprogramming	0.29	0.40	0.39	0.19	0.53	**0.54**
	scrobot	0.06	0.09	0.08	**0.24**	0.12	0.14
	scSlideShow	0.55	0.71	0.64	0.26	0.82	**0.87**
	scwebbrowsing	0.94	1.03	1.03	0.34	1.19	**1.20**
	Average	0.31	0.44	0.40	0.17	0.55	**0.58**
32	Average	0.12	0.33	0.03	0.13	0.37	**0.43**
27	Average	0.10	0.27	-0.05	-0.01	0.28	**0.36**
22	Average	0.03	0.19	-0.47	-0.01	0.22	**0.30**

Table 3.3: Overall ΔSSIM(10^{-3}) of Different Models at QP=22,27,32,37

QP	Sequence	QECNN	DCAD	Partition-aware CNN	QECF	baseline	MICNN
37	BasketballScreen	2.17	2.81	2.50	2.88	3.37	**3.47**
	ChineseEditing	1.93	3.00	3.39	1.80	3.92	**4.11**
	MissionControlClip2	1.59	2.24	2.09	2.06	2.85	**3.16**
	MissionControlClip3	1.75	2.61	2.27	2.51	3.09	**3.56**
	scconsole	-0.09	0.01	-0.27	-0.15	0.19	**0.67**
	scdesktop	0.56	0.82	0.50	-0.11	1.13	**1.32**
	scflyingGraphics	0.69	1.53	1.32	-7.39	2.32	**2.39**
	scmap	0.15	3.11	4.22	0.99	3.93	**4.70**
	scprogramming	1.05	2.12	1.61	1.09	2.81	**2.99**
	scrobot	-0.64	0.01	-0.79	**4.99**	0.93	1.20
	scSlideShow	0.99	1.50	1.18	0.67	1.78	**1.90**
	scwebbrowsing	0.84	1.30	0.58	0.29	1.60	**1.80**
	Average	0.92	1.76	1.55	0.80	2.33	**2.61**
32	Average	0.30	0.78	0.15	0.64	1.01	**1.34**
27	Average	0.09	0.41	0.13	-0.01	0.52	**0.57**
22	Average	-0.01	0.13	-0.07	0.01	0.16	**0.22**

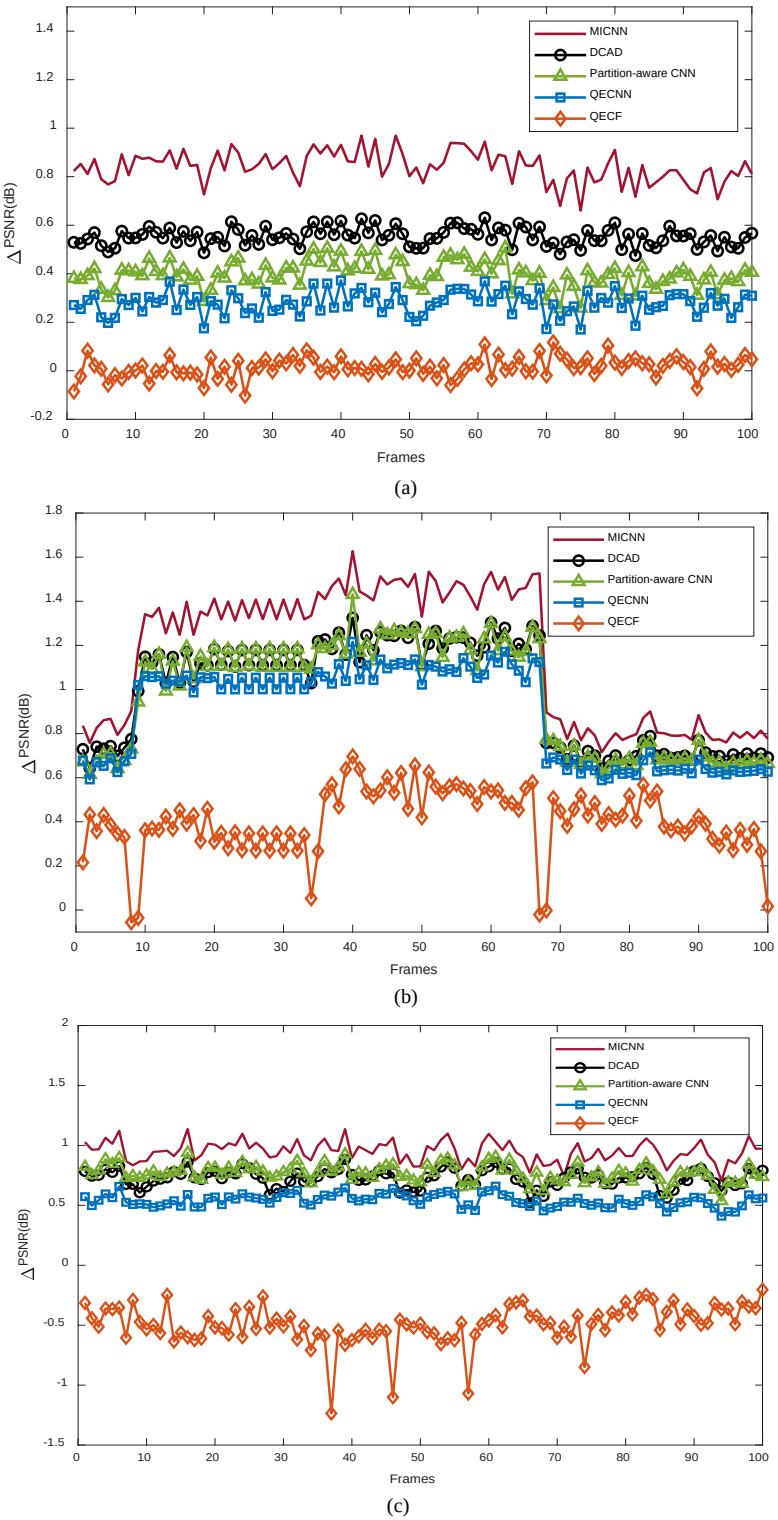


Figure 3.5: Δ PSNR curves of partition-aware CNN, DCAD, QECNN, QECF and our MICNN method for sequences, (a) scdesktop, (b) scwebbrowsing, and (c) scflyingGraphics.

Table 3.4: Overall BD-rate(%) of Different Model at QP=22,27,32,37

Sequences	QECNN	DCAD	Partition-aware CNN	QECF	baseline	MICNN
BasketballScreen	-1.67	-3.40	-1.54	-1.66	-3.80	**-4.22**
ChineseEditing	-0.55	-1.24	-0.09	-0.54	-1.66	**-1.68**
MissionControlClip2	-1.61	-3.38	-1.67	-1.76	-4.06	**-4.43**
MissionControlClip3	-1.78	-3.56	-1.53	-1.75	-4.15	**-4.54**
scconsole	0.73	-0.17	2.96	0.30	-0.11	**-0.59**
scdesktop	-0.15	-1.09	0.96	0.07	-1.38	**-1.79**
scflyingGraphics	-1.30	-2.73	-0.89	1.00	-2.96	**-3.26**
scmap	-2.37	-4.41	-2.54	-1.21	-5.01	**-5.76**
scprogramming	-0.77	-2.23	-0.33	-0.68	-2.65	**-3.34**
scrobot	-0.18	-1.10	0.03	-1.60	-1.38	**-2.07**
scSlideShow	-3.51	-5.83	-2.24	-1.56	-6.57	**-6.76**
scwebbrowsing	-1.42	-2.25	-0.63	-0.51	-1.85	**-2.42**
Average	-1.21	-2.62	-0.63	-0.83	-2.97	**-3.41**

that of DCAD (0.44 dB), and 0.27dB higher than that of QECNN (0.31 dB). It is noted that QECF includes some specific idea to enhance gaming content. However, it is found that our proposed method can also handle gaming content and text content. Compared with the QECF, our MICNN can achieve an acceptable PSNR improvement (0.14dB) and SSIM improvement (0.0012) in gaming content sequence scrobot and outperform other sequences. In addition, ΔPSNR curves of three pure screen content videos for DCAD, QECNN, partition-aware CNN, QECF, and our proposed MICNN are shown in Fig. 3.5. The *scdesktop* is mixcontent. The *scwebbrowsing* and *scflyingGraphics* are pure screen content. By utilizing the proposed binary mode masks, MICNN can achieve highest PSNR in each frame of different content. That means our proposed method is robust.

BD-rate [99] is used to indicate the bitrate savings of these models under the equivalent PSNR. Experimental results are compared and tabulated in Table 3.4. It shows that our proposed MICNN can achieve higher BD-rate savings than its corresponding baseline. Again, this demonstrates the effectiveness of using mode masks. Our MICNN obtains an average BD-rate savings of 3.41%, while the second-best method achieves an average BD-rate savings of only 2.97%. For the test sequence *scSlideShow*, up to 6.76% BD-rate saving is obtained for the Y component under AI configuration. We conjecture that our MICNN well exploits the mode information to further enhance the decoded frame quality

and reduce the BD-rate.

3.3.3 Subjective Visual Quality Comparison

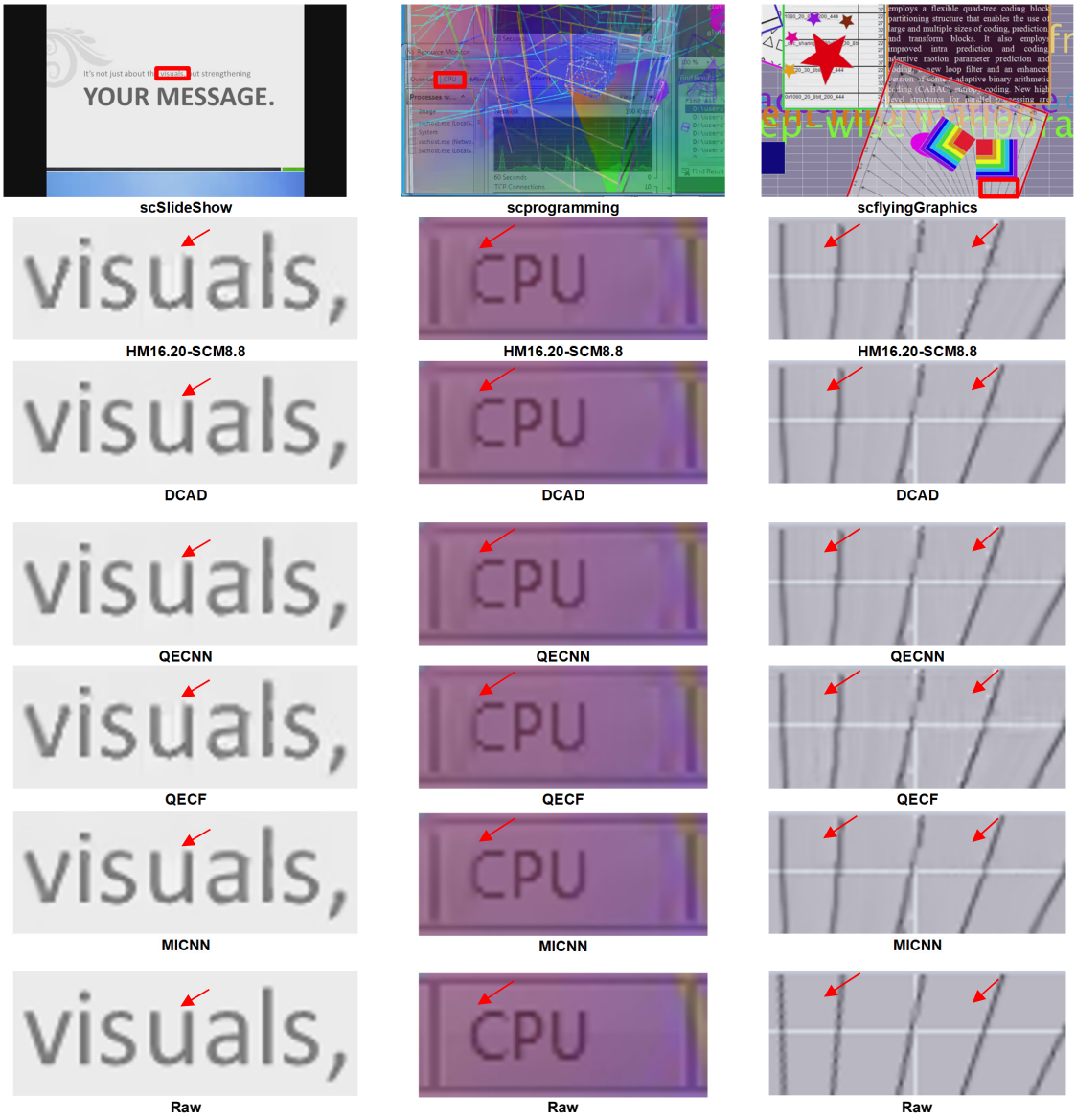


Figure 3.6: Subjective visual quality comparison at QP = 37 on (a) *scSlideShow*, (b) *scprogramming*, and (c) *scflyingGraphics*.

This section compares the subjective quality of different models. Fig. 3.6 shows the subjective visual quality performance of various models on the sequences *scSlideShow*, *scprogramming*, and *scflyingGraphics* with QP = 37. From this figure, we can see that the reconstructed frame of HM16.20-SCM8.8 has obvious compression artifacts, which

cannot be completely removed by DCAD, QECNN, or QECF. As shown in Fig. 3.6, our MICNN eliminates the artifacts more effectively than other models. For *scSlideShow* and *scprogramming*, it can be observed that the character is blurry, and there are some blocking artifacts in the background around the character, but it becomes clearer after being processed by our proposed MICNN. For *scflyingGraphics*, the lines are blurry in the reconstructed frame but become shaper in MICNN. In addition, in the reconstructed frame, the flat areas around the lines contain many artifacts. MICNN can smooth these areas. All these examples in Fig. 3.6 show that MICNN is superior to the other models in terms of subjective visual quality. There are no uneven regions at the CU boundary and no visual blocking effect from the frame processed by MICNN. This again shows that our MICNN can make use of the mode information to enhance the decoded frame quality subjectively.

3.3.4 Quality enhancement at various QPs

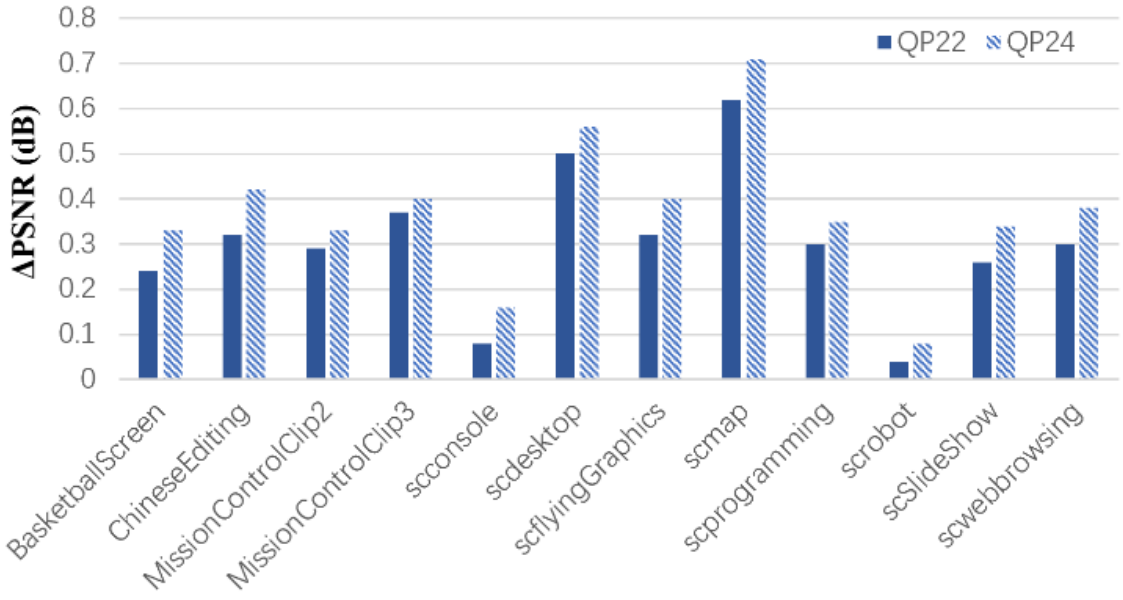
To verify the generalization ability of the MICNN model on various QPs, we additionally encode all test sequences at QP of 24, 29, 34, 39 when the model is trained at different QPs, *i.e.* QP=22, 27, 32, and 37. The performance in terms of ΔPSNR is shown in Fig. 3.7. Fig. 3.7(a) shows the PSNR improvement of the model trained at QP = 22 and tested at QP = 22 and 24. In Fig. 3.7(b) , the model is trained at QP = 27 and tested at QP = 27 and 29. Similarly, Fig. 3.7(c) and Fig. 3.7(d) show ΔPSNR of the model trained at QP =32 and 37 and tested at different QPs = 32 and 34, 37 and 39, respectively. As shown in this figure, each trained model can obtain acceptable quality enhancement on decoded videos at adjacent QPs, which verifies the generalization ability on various QPs.

3.3.5 Model Parameters and Computational Complexity

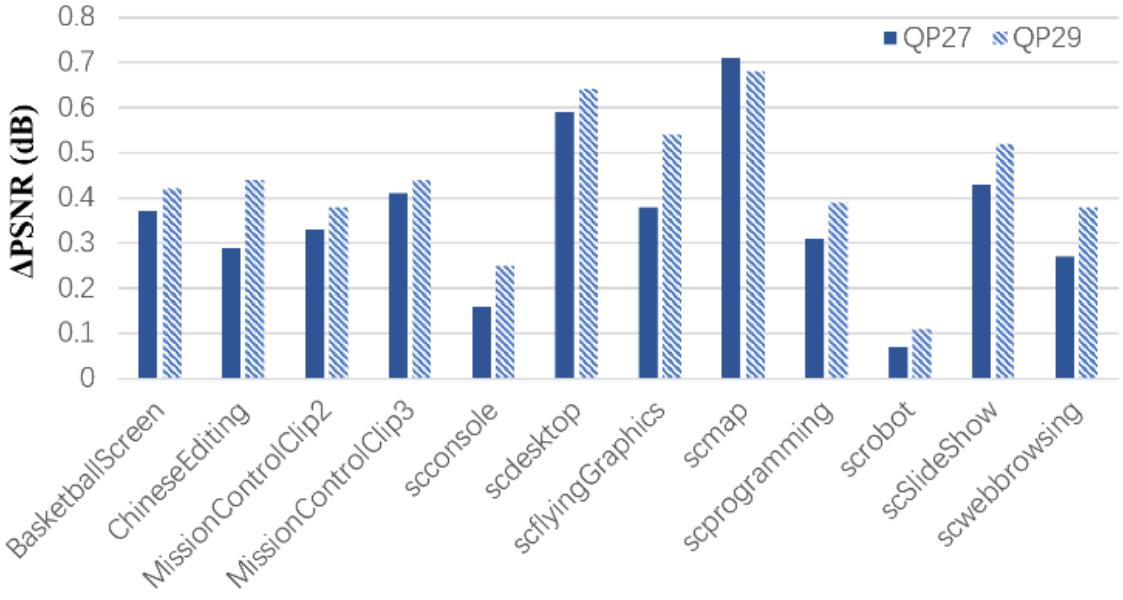
Table 3.5: Comparision of Model Size

Model	QECNN	DCAD	Partition-aware CNN	QECF	baseline	MICNN
Model size (KB)	451.78	296.64	3114.31	764.067	1268.16	1269.89

To evaluate the computational complexity of various models, we follow the measure-

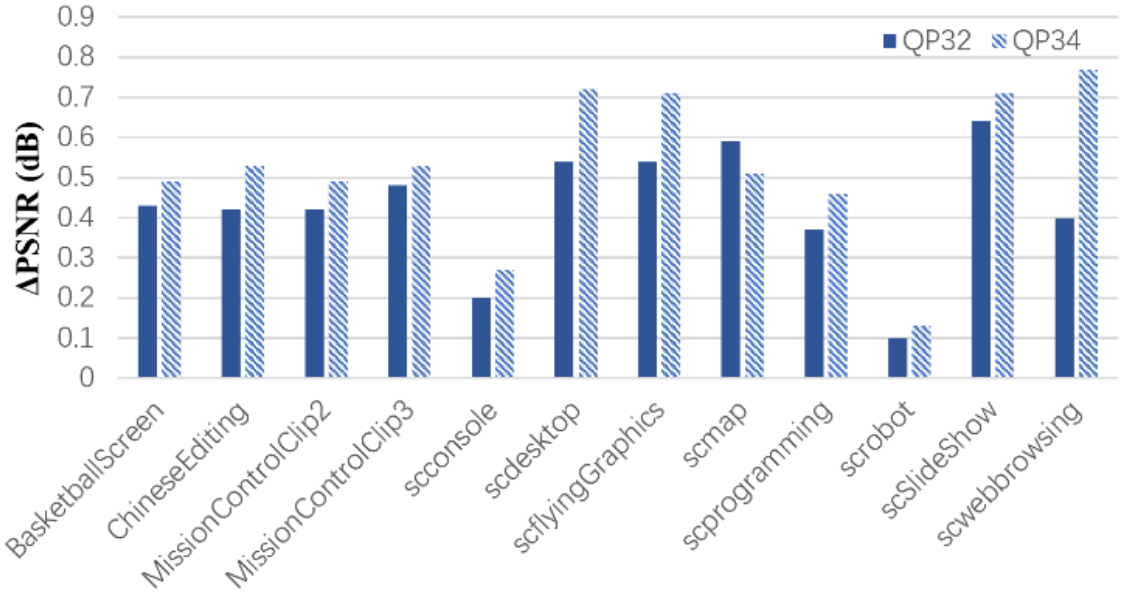


(a)

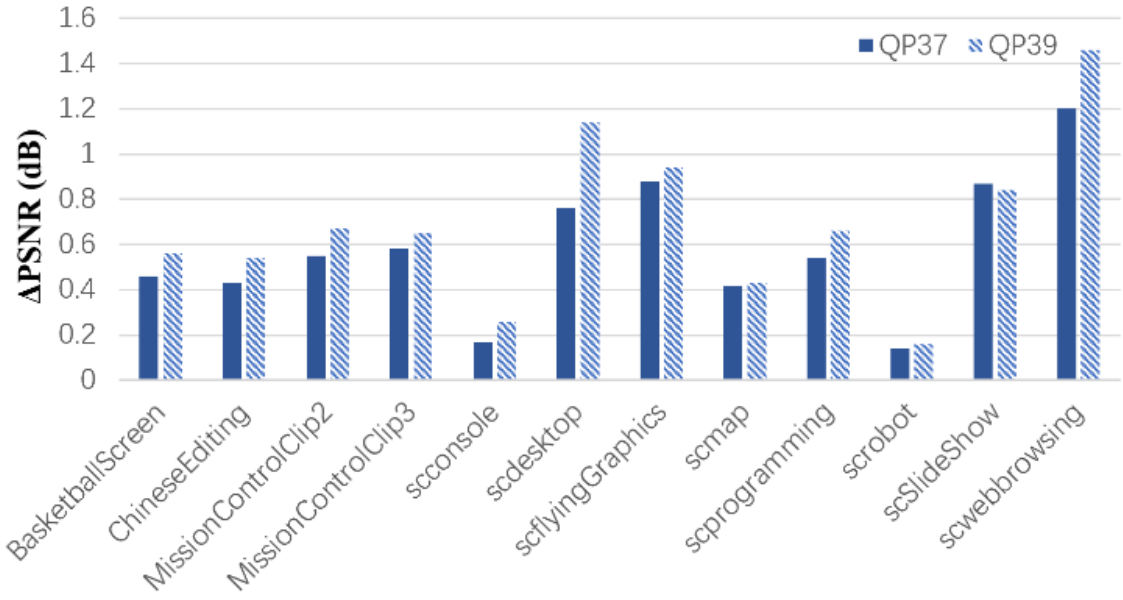


(b)

Figure 3.7: ΔPSNR of the model trained and tested at different QPs under AI configuration. (a) Trained at QP=22 and tested at QPs 22 and 24, (b) trained at QP=27 and tested at QPs 27 and 29.



(c)



(d)

Figure 3.7: ΔPSNR of the model trained and tested at different QPs under AI configuration. (c) trained at QP=32 and tested at QPs 32 and 34, and (d) trained at QP=37 and tested at QPs 37 and 39.

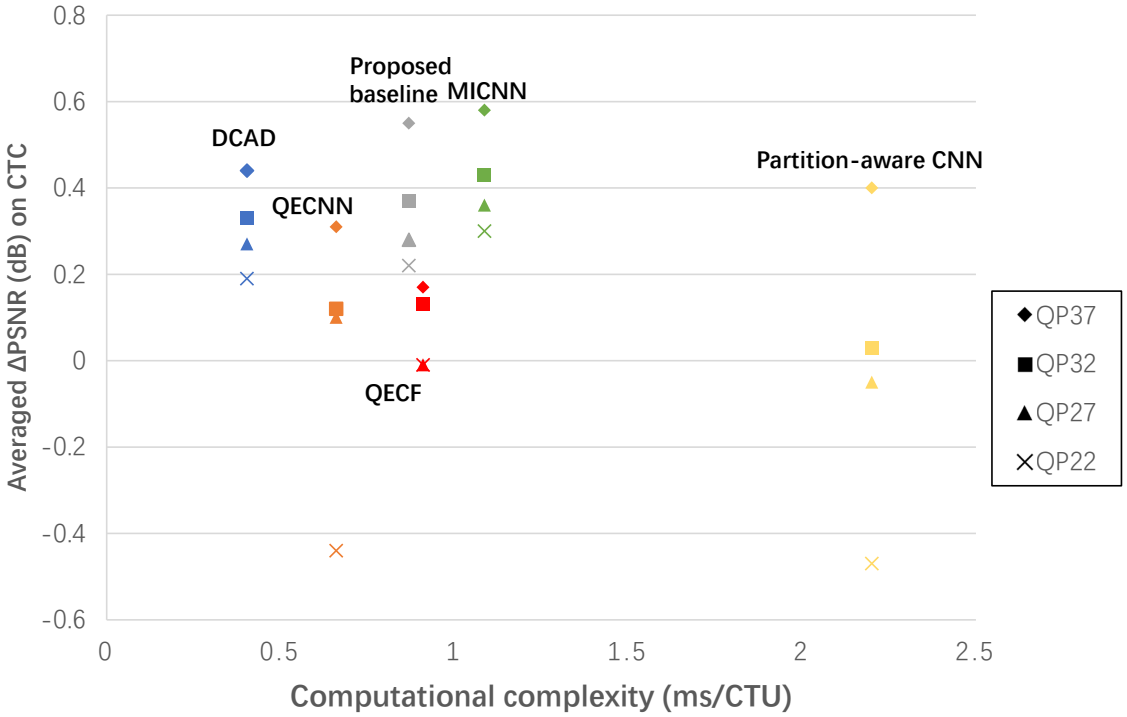


Figure 3.8: Average ΔPSNR against computational complexity of different methods in the decoder side.

ment metric of other post-processing algorithms [11, 28] by computing the running time per Coding Tree Unit (CTU) at the decoder side. Experiments were conducted using Intel i9-10900K CPU, 64 GB RAM, and NVIDIA 3090Ti GPUs. Fig. 3.8 shows the average ΔPSNR against running time per CTU for MICNN, DCAD [9], QECNN [27], QECF [17], and partition-aware CNN [28] methods. The results shown in this figure are calculated over all the test sequences on average. In Fig. 3.8, the running times of DCAD, QECNN, QECF, partition-aware CNN are 0.40 ms per CTU, 0.66 ms per CTU,0.91 ms per CTU, and 2.20 ms per CTU, respectively. On the other hand, our proposed MICNN model consumes approximately 1.08 ms per CTU but achieves the highest ΔPSNR over other models. From Table 3.5, we can observe that the performance improvement of our MICNN consumes a reasonable amount of computational time compared to QECNN and DCAD. Moreover, MICNN outperforms partition-aware CNN in both running time and ΔPSNR.

Model complexity in terms of model size for various models is also evaluated in Table 3.5. Model size reflects the number of network parameters. Compared to our baseline model, MICNN adds sub-branches to improve performance without significantly affect-

ing model size. Besides, the proposed MICNN can achieve higher performance than the partition-aware CNN, but with smaller model size. It can be concluded that our MICNN obtains better tradeoff between coding efficiency and model size. In other words, our MICNN is more model-efficient.

3.3.6 Ablation Study

Table 3.6: Different Orders of the Binary Mode Masks at QP=37

Seq.	1	2	3	4	5	6	7
BigBuck	0.4	0.35	0.4	0.41	0.4	0.38	0.39
consolenew2	1.09	1.05	1.06	1.09	1.08	1.07	1.14
EnglishDocumentEditing	1.03	1.07	1.09	1.1	1.12	1.05	1.03
KimonoError1	0.51	0.47	0.49	0.57	0.53	0.54	0.5
MissionControlClip1	0.51	0.48	0.5	0.53	0.52	0.49	0.49
scviking	0.11	0.11	0.11	0.12	0.11	0.11	0.11
Youtube	0.52	0.52	0.54	0.58	0.55	0.52	0.55
Average	0.596	0.579	0.599	0.629	0.616	0.594	0.601

1: ibc-plt-intra 2: plt-ibc-intra 3: intra-plt-ibc 4: ibc-intra-plt 5: intra-ibc-plt 6: plt-intra-ibc 7: mean mask.

Table 3.7: Different Masks at QP=37

Seq.	EFC	LFC	Proposed
BigBuck	0.28	0.40	0.41
consolenew2	0.48	1.13	1.09
EnglishDocumentEditing	0.81	1.01	1.1
KimonoError1	0.38	0.46	0.57
MissionControlClip1	0.39	0.46	0.53
scviking	0.09	0.13	0.12
Youtube	0.43	0.54	0.58
Average	0.409	0.590	0.629

As mentioned in Section 3.1.3, the order of the three binary mode masks fused in our proposed MICNN will affect performance. An ablation study was conducted to decide the order of the three binary mode masks and verify the necessities and the generalization ability of our proposed masks. Various MICNN architectures were compared to find the optimal order of inputting binary mode masks. It includes all possible combinations as in Table 3.8: ibc-plt-intra, plt-ibc-intra, intra-plt-ibc, ibc-intra-plt, intra-ibc-plt, and plt-intra-ibc. These notations represent different orders of the binary mode masks by name.

Table 3.8: Different Fusion Strategies at QP=37

ibc	intra	plt	ΔPSNR	Parameter (KB)
-	-	-	0.579	1268.16
-	✓	-	0.607	1268.74
✓	-	✓	0.617	1269.31
-	✓	✓	0.610	1269.31
✓	✓	-	0.599	1269.31
✓	✓	✓	0.629	1269.89

Table 3.9: Different Baselines at QP=37

Seq.	Mode+Residual Block	Mode+Dense Block	MICNN
BigBuck	0.32	0.39	0.41
consolenew2	0.98	1.04	1.09
EnglishDocumentEditing	0.96	1.08	1.1
KimonoError1	0.43	0.56	0.57
MissionControlClip1	0.43	0.50	0.53
scviking	0.11	0.12	0.12
Youtube	0.49	0.53	0.58
Average	0.531	0.603	0.629

For example, ibc-plt-intra means first use the IBC mode mask, then add the PLT mode mask, and finally use the INTRA mode mask. Furthermore, to verify the superiority of our mode mask, we input the mean mask proposed in [28] into the baseline model in Fig. 3.9(a) with the same number of layers and the same training process. The PSNR improvement of various combinations on the validation set under AI configuration is shown in Table 3.6. It can be seen that ibc-intra-plt can achieve the highest PSNR improvement (0.629 dB) over the SCC baseline at QP=37. It verifies the efficiency of the order of ibc-intra-plt. The ibc-intra-plt order works best because it matches how errors appear in screen content video. IBC regions (texts and repeated UI parts) often have the strongest seams and copy mismatch, so using the IBC mask first helps the model focus on the largest errors early. INTRA regions mainly show blocking and ringing, common in natural parts of mixed content, and affect the QoE, so handling them at the next order. PLT regions are mostly flat with few colors; placing PLT last lets the model pay attention to restoring these areas without damaging earlier extraction of features. This order reduces interference between stages and gives the highest PSNR. To further verify the efficiency of our proposed fusion

approach as in Fig. 3.9(a), we also evaluated two different fusion strategies - Early Fusion by Concatenation (EFC) and Late Fusion Concatenation (LFC), as shown in Fig. 3.9(b) and Fig. 3.9(c), respectively. In EFC, we concatenate the decoded frame and binary mode masks as the input. The main branch of EFC is the same with our proposed MICNN. On the other hand, the subbranch of the LFC is the same as our proposed MICNN. As compared with MICNN, LFC concatenates all feature maps of decoded frame and binary mode masks before the feature reconstruction stage. The PSNR improvements for various fusion strategies are shown in Table 3.7. It can be seen that our proposed fusion strategy can achieve the highest PSNR improvement and it can make better use of the mode information. In Table 3.8, to further verify the contribution of our proposed mode masks, we remove the intra mode mask, ibc mode mask, and plt mode mask, respectively. The result shows that the best performance can be achieved when the three mode masks are adopted. To verify the power of feature extraction of the RDB, we employed the

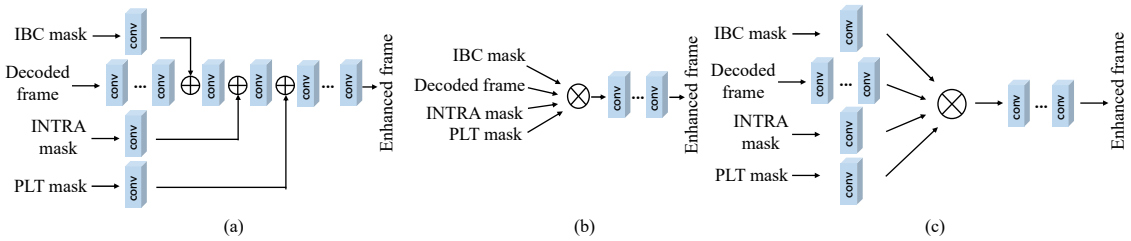


Figure 3.9: Examples of three binary mode masks. (a) Original frame with CU partition, (b) IBC binary mask, (c) PLT binary mask, and (d) INTRA binary mask.

traditional Residual Block as shown in Figure 3.3(d) and the traditional Dense Block [98] instead of the RDB for compression. The results are shown in Table 3.9, the RDB can achieve the highest PSNR performance. Combining residual block and dense connection can help to extract the feature and keep the high frequency details. The reason is that the residual connection can prevent the gradient vanishing and the dense connection can reuse the feature from previous layers.

3.4 Summary

By integrating our proposed binary mode masks into a mode information guided deep network model, SCC modes extracted from the bitstream can be utilized to further improve

SC video quality. Specifically, the new branch uses the binary mode masks, which are based on the coding modes of SCC, to exploit the characteristics of SCC, and then guide the neural network for quality enhancement on screen content videos. This is the first work to incorporate the SCC mode information into the sub-branches for enhancing SC quality. Experimental results show that our proposed MICNN is more effective than other networks. We believe that our mask branches can be easily adopted to different single-input models for further quality enhancement of SCC.

Chapter 4

Spatio-temporal Feature Learning for Enhancing Video Quality Based on Screen Content Characteristics

4.1 Proposed EAST Method

4.1.1 Motivation

As discussed in Section 1.1, artifacts manifest across the entire area of natural content due to its diverse color range and camera noise. In contrast, these artifacts predominantly occur along the edges in screen content. We observe that when the content remains unchanged across multiple frames, the PSNR of a compressed frame remains constant, as illustrated by the shadow region in Fig. 1.3. This phenomenon, known as "frame freezing," is rare in natural videos due to camera noise. Additionally, during activities such as web browsing, the video content may abruptly change in the subsequent frame—a scenario we term a "scene switch." Scene switches occur frequently in screen content videos and are denoted by dashed lines in Fig. 1.3. These switches lead to significant drops in PSNR, resulting in noticeable quality degradation that can greatly affect the Quality of Experience (QoE).

As previously mentioned, existing techniques such as flow-based alignment [13, 14]

and deformable-based alignment [15,16] struggle to accurately adjust for position changes from neighboring frames when substantial content variations occur between frames [17]. The lack of precision in the prediction network can undermine the effectiveness of the quality enhancement network. Alignment-based methods, such as optical flow and deformable convolution, assume that the target frame and its neighbor frames match at the pixel or feature level. In screen content videos, this assumption often fails during sudden scene switches (e.g., a new webpage or a slide change). In these cases, there is no valid correspondence, so the estimated flow or the learned offsets become noisy and wrong. Warping with these wrong fields produces corrupted references, and fusing them will increase artifacts rather than remove them. In frame-freezing segments, where the content barely changes, alignment brings little gain but still adds extra computation and possible errors. Additionally, the alignment-free, multi-frame approach described in [17] extracts high-quality regions from neighboring frames to enhance the target frame. However, during scene switches, these neighboring frames may provide irrelevant information to the quality enhancement model. Furthermore, in instances of frame freezing, the conventional multi-frame structure is ineffective at leveraging neighboring frames to extract useful information for enhancing the target frame. Therefore, there is a critical need to develop a new video quality enhancement method that can effectively address the challenges posed by frame freezing and scene switches in screen content videos.

To address these issues, we introduce a new methodology known as the Edge Aware with Spatio-Temporal Information Fusion Network (EAST). Our EAST approach features a spatio-temporal feature extraction module specifically designed to discern and extract relevant features from various groups of input frames, effectively addressing the complexities introduced by scene switches in screen content videos. Drawing insights from Fig. 1.1, we have also developed an innovative edge aware block that prioritizes the extraction of high-frequency information from the target frame. This block is crucial for restoring the high-frequency details in the spatial domain, enhancing the quality of the frame.

Moreover, in scenarios of frame freezing where the contribution from neighboring frames is minimal, the inclusion of edge information proves essential in compensating for this deficiency. To dynamically enhance target frames under conditions of scene switches

and frame freezing, we introduce the Channel and Spatial Attention Block (CSAB). This block integrates both channel and spatial attention modules within the spatio-temporal feature fusion process. Leveraging spatio-temporal information, the CSAB adeptly allocates attention to varying scenarios, thereby ensuring the effective enhancement of overall video quality in screen content videos affected by scene switches and frame freezing.

4.1.2 Overview of the Framework

Our EAST model, as shown in Fig. 4.1, aims to remove the artifact in screen content video that involves numerous scene switch and frame freezing scenarios. In this context, we denote a low-quality frame at time t as $I_t^{LQ} \in \mathbb{R}^{H \times W}$, where H and W indicate the vertical and horizontal resolutions of the frame. Our model aims to enhance the quality of I_t^{LQ} by considering the preceding and succeeding $R = 3$ frames as references to leverage temporal information. This long input frames can be expressed as $F_L = \left\{ I_{t-R}^{LQ}, \cdots, I_t^{LQ}, \cdots, I_{t+R}^{LQ} \right\}$. To account for scene switches, we divide the input sequence into two parts, ensuring one part belongs to a similar scene as the target frame. This division helps to reduce the influence of unrelated neighbor frames. Consequently, the input frames are split into the previous group of frames $F_{S1} = \left\{ I_{t-R}^{LQ}, \cdots, I_t^{LQ} \right\}$ and the future group of frames $F_{S2} = \left\{ I_t^{LQ}, \cdots, I_{t+R}^{LQ} \right\}$ based on the I_t^{LQ}. F_L, F_{S1}, and F_{S2} are then input into the model through three separate branches as in Fig. 4.1. In this arrangement, the FL branch is dedicated to capturing slow motion or continuous motion, while F_{S1}, and F_{S2} branches are used to handle potential scene switches and dramatic motion. The arrangement ensures efficient extraction of temporal information while minimizing disruptions caused by sudden scene switches. In order to emphasize high-frequency information in the spatial domain, we also extract the edge information f_e of the low-quality frame I_t^{LQ}, as the input of the model. As a result, the enhanced frame $\tilde{I}_t^{HQ} \in \mathbb{R}^{H \times W}$ can be expressed as

$$\tilde{I}_t^{HQ} = H_{EAST}(F_L, F_{S1}, F_{S2}, f_e) \tag{4.1}$$

where $H_{EAST}(\cdot)$ represents the proposed model. Next, we will discuss our proposed framework in Fig. 4.1, which comprises four main parts: the Spatio-Temporal Feature Extraction (STFE), the Edge Aware Block (EAB), the Spatio-Temporal Feature Fusion (STFF),

and three convolutional layers. We will provide a detailed explanation of each component.

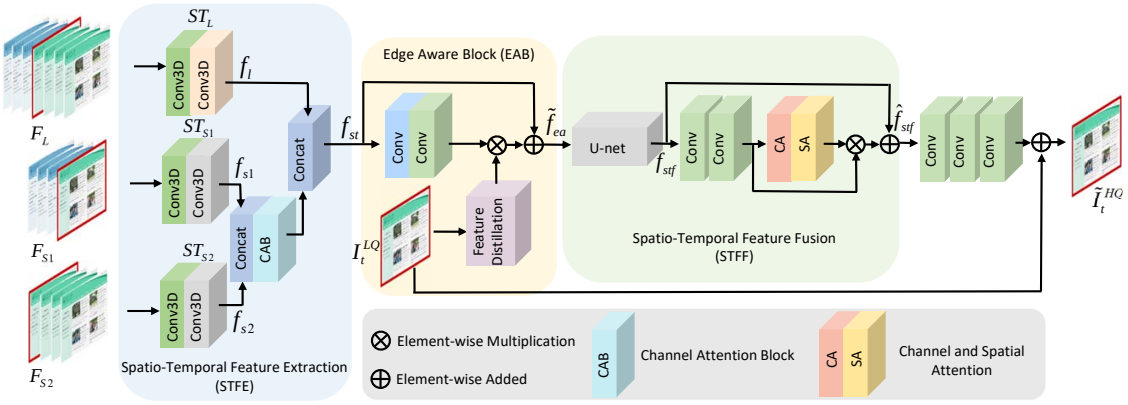


Figure 4.1: Our proposed EAST structure.

4.1.3 Spatio-Temporal Feature Extraction (STFE)

The STFE is designed to capture slow motion or continuous motion, and adapt to potential scene switches and dramatic motion. It extracts features from different groups of input frames, utilizing both spatial and temporal domains. During scene switches, it looks for the temporal features that are most relevant to the target frame. By focusing on these specific temporal features, the quality of the generated frame can be enhanced. By incorporating 3D convolution layers, the STFE can effectively capture and summarize the temporal information. This is particularly advantageous when the content of consecutive frames exhibits similarity. In such cases, incorporating neighbor frames can provide additional information that enhances the quality of the target frame. In the first branch $ST_L(\cdot)$, the $3 \times 3 \times 3$ convolution is operated in the entire input frames, and then $7 \times 3 \times 3$ convolution is utilized to compress the temporal information, which can be formulated as:

$$f_l = ST_L(F_L) \tag{4.2}$$

where f_l is the output feature maps obtained by extracting from the entire input frames, F_L.

Similarly, the second branch $ST_{S1}(\cdot)$ and the third branch $ST_{S2}(\cdot)$ are used to handle potential scene switches and dramatic motion. Through the 3D convolution layers, the

inputs F_{S1} and F_{S2} convolve with $3 \times 3 \times 3$ filters to obtain higher-level feature maps and then through $4 \times 3 \times 3$ convolution layers to obtain feature maps f_{s1} and f_{s2}, respectively. After that, a channel attention block consisting of a channel attention module [60] and a residual block is applied to reinforce the different significance of related features of I_t^{LQ}. The residual block, which is part of the channel attention block, consists of 1×1 and 3×3 convolution layers to limit the model complexity. This design helps to restrict the model complexity while preserving important spatial information within the feature maps. Maintaining spatial information is crucial for later stages when it needs to be combined with edge information. The module can be mathematically formulated as:

$$f_{s1} = ST_{S1}(F_{S1}) \tag{4.3}$$

$$f_{s2} = ST_{S2}(F_{S2}) \tag{4.4}$$

$$f_{st} = [f_l, CAB([f_{s1}, f_{s2}])] \tag{4.5}$$

where $CAB(\cdot)$ represents the channel attention block.

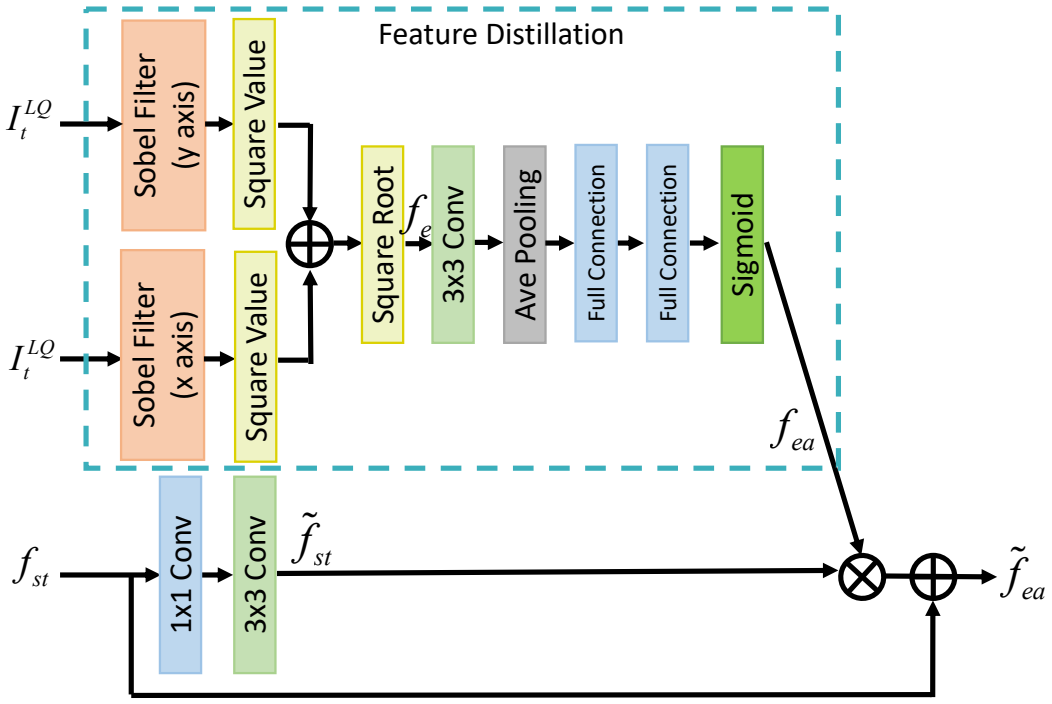


Figure 4.2: Edge aware block (EAB).

4.1.4 Edge Aware Block (EAB)

After extracting the feature associated with the target frame, it is crucial to restore the high-frequency details, which are mainly affected in screen content videos, as shown in Fig 4.2. Besides, in frame freezing, neighbor frames carry little new information because the content is almost unchanged. Traditional multi-frame fusion may bring limited gains. Edge information extracted from the target frame itself provides a strong, spatially precise prior about where high frequencies are. As mentioned in section 1.2.1, one of the challenges of screen content quality enhancement is that the artifact mainly occurs in the edge regions of screen content. The EAB turns the edge map computed from the target frame with a Sobel filter into attention weights that guide the network to focus on these edge regions,), which are the places where screen content artifacts usually appear. In this way, edge information gives a strong, spatially accurate prior that replaces missing temporal cues. Hence, the incorporation of edge information becomes critical as it assists the network in restoring sharp edges of the compressed frame and provides valuable supplementary information during frame freezing. By extracting edge information from the target frame and distilling relevant features as channel weights, the network can effectively utilize the edge information. This process enables the network to emphasize features associated with high-frequency information in low-quality frames. As illustrated in Fig. 4.2, our EAB processes the target frame by applying the Sobel filter [104] along the x-axis and y-axis, respectively. The squared values of the filter outputs are element-wise added and then processed through the square root operation. This process yields the representation of the edge information $f_e \in \mathbb{R}^{1\times H\times W}$ as:

$$G_x = \begin{bmatrix} +2 & 0 & -2 \\ +4 & 0 & -4 \\ +2 & 0 & -2 \end{bmatrix} * I_t^{LQ} \tag{4.6}$$

$$G_y = \begin{bmatrix} +2 & +4 & +2 \\ 0 & 0 & 0 \\ -2 & -4 & -2 \end{bmatrix} * I_t^{LQ} \tag{4.7}$$

$$f_e(i,j) = \sqrt{[G_x(i,j)]^2 + [G_y(i,j)]^2} \tag{4.8}$$

where $i = 1, \cdots$ and $H, j = 1, \cdots, W$. Each point in G_x and G_y contains the horizontal and vertical derivative approximations respectively. Then, f_e is refined through a convolutional layer. To distill the attention weight from the edge information, we employ average pooling $AvgPool(\cdot)$ to generate the attention weight. Finally, the output attention weight $f_{ea} \in \mathbb{R}^{C\times1\times1}$, where C denotes the channel number of feature, generated from two fully-connected layers and the sigmoid function is element-wise multiplied by the input feature $\tilde{f}_{st} \in \mathbb{R}^{C\times H\times W}$ to highlights the features associated with high-frequency information in the low-quality frame I_t^{LQ}, effectively reducing artifacts around sharp edges. The output of the EAB is expressed as:

$$f_{ea} = \sigma(w_{FC}^2(w_{FC}^1 AvgPool(\delta(w_{3\times3}^1 f_e)))) \tag{4.9}$$

$$\tilde{f}_{ea} = f_{ea} \otimes \tilde{f}_{st} + f_{st} \tag{4.10}$$

where $\otimes$ denotes the element-wise multiplication. $\sigma(\cdot)$ and $\delta(\cdot)$ are the sigmoid function and ReLU function, respectively. $w_{3\times3}^1 \in \mathbb{R}^{C\times1\times3\times3}$ is the weight of the first 3×3 convolutional layer. $w_{FC}^1 \in \mathbb{R}^{\frac{C}{r}\times C\times1\times1}$ and $w_{FC}^2 \in \mathbb{R}^{C\times\frac{C}{r}\times1\times1}$ are the weights of the fully-connected layers. Meanwhile, r denotes the scale ratio of channel downsampling, which is introduced in Section 4.2.1.

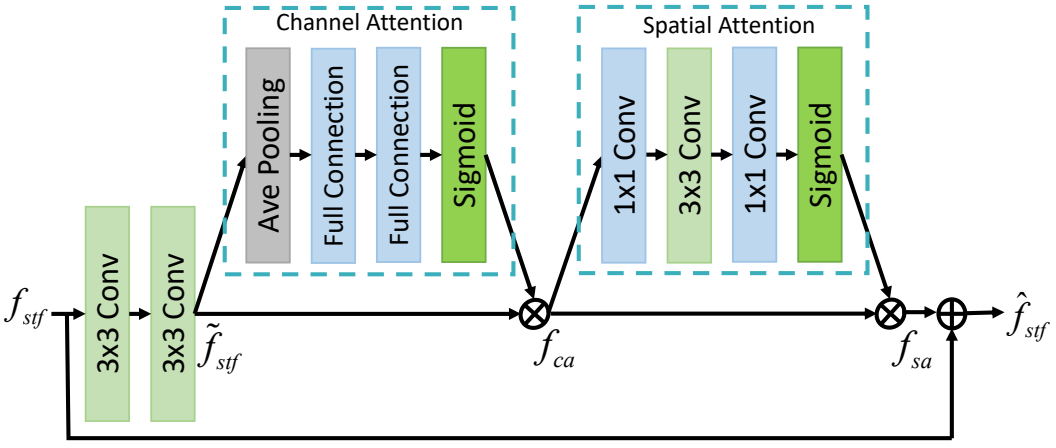


Figure 4.3: Channel and spatial attention block (CSAB).

4.1.5 Spatio-Temporal Feature Fusion (STFF)

The STFF is designed to adaptively enhance frames during scene switches and frame freezing scenarios. It fuses the obtained features in a manner that optimally enhances the target frame, taking into account the specific context and requirements of each scenario. To fuse spatio-temporal information, we employ an autoencoder approach with skip connections, taking into account the structure of U-net [105] and the model's complexity [15]. In order to emphasize the pertinent features of the target frame in both spatial and temporal domains, we introduce the Channel and Spatial Attention Block (CSAB), which consists of the channel attention module $H_{CA}(\cdot)$ and spatial attention module $H_{SA}(\cdot)$ as shown in Fig. 4.3. The channel attention module enables the model to adaptively enhance the target frame, particular under the scene switch and the slow-motion clip. In the scene switch situations, the model places greater emphasis on $ST_{S1}(\cdot)$ and $ST_{S2}(\cdot)$. In contrast, in slow-motion scenarios, the model focuses more on $ST_L(\cdot)$. This attention mechanism allows the model to dynamically adjust its enhancement strategy based on the specific context. To summarize, the overall attention process can be described as follows:

$$f_{ca} = H_{CA}(\tilde{f}_{stf}) \otimes \tilde{f}_{stf} \tag{4.11}$$

$$f_{sa} = H_{SA}(f_{ca}) \otimes f_{ca} \tag{4.12}$$

$$\hat{f}_{stf} = f_{sa} + f_{stf} \tag{4.13}$$

For the channel attention, a 2D average pooling layer along $H \times W$ is adopted to generate attention weights for each channel. Then the combination of two fully-connected layers is utilized to extract the channel interdependencies and increase the sensitivity of network to informative features. Finally, a sigmoid function is applied to normalize the channel attention weights, ensuring they range from 0 to 1. In short, the channel attention is computed as:

$$H_{CA}(\tilde{f}_{stf}) = \sigma(w^4_{FC}\delta(w^3_{FC}AvgPool(\tilde{f}_{stf}))) \tag{4.14}$$

where $w^3_{FC} \in \mathbb{R}^{\frac{C}{r}\times C\times 1\times 1}$ and $w^4_{FC} \in \mathbb{R}^{C\times\frac{C}{r}\times 1\times 1}$ are the weight of the fully-connected layers.

The spatial attention module plays a crucial role in assigning high weights to the spatial features of the target frame. This spatial attention map guides the model to disregard irrelevant features from neighbor frames and distill the useful information, particularly after the combination with edge information. To generate the 2D spatial attention maps, the combination of two 1×1 convolutional layers, two ReLU functions, and a 3×3 convolutional layer is adopted. Following by this, a sigmoid function is utilized to normalize the spatial attention maps. The output of the spatial attention module can be represented as:

$$H_{SA}(f_{ca}) = \sigma(w^2_{1\times1}\delta(w^2_{3\times3}\delta(w^1_{1\times1}f_{ca}))) \tag{4.15}$$

where $w^1_{1\times1} \in \mathbb{R}^{\frac{C}{r}\times C\times1\times1}$ and $w^2_{1\times1} \in \mathbb{R}^{1\times\frac{C}{r}\times1\times1}$ are the weights of the first and second 1×1 convolutional layers. $w^2_{3\times3} \in \mathbb{R}^{\frac{C}{r}\times\frac{C}{r}\times3\times3}$ belongs to the convolutional 3×3 layer.

4.1.6 Training Scheme

To effectively handle the high-frequency information and improve the performance, we adopt the robust Charbonnier loss function in [106] to train our model in an end-to-end manner. The loss function L is represented as:

$$L = \sqrt{\left\| I_t^{HQ} - \tilde{I}_t^{HQ} \right\|^2 + \varepsilon^2} \tag{4.16}$$

where I_t^{HQ} is the ground truth frame at time t, $\tilde{I}_t^{HQ}$, represents the enhanced frame generated by our model, and $\varepsilon = 10^{-3}$ is a constant value used across all experiments.

4.2 Experimental Results

4.2.1 Experimental Setting

Our proposed EAST model mainly focuses on enhancing the video quality of screen content sequences. We set the ratio r as 16 in the attention mechanism. In our EAST framework, each convolutional layer, except for the one in front of the sigmoid function and the final convolutional layer, is followed by a ReLU activation function [107] to introduce non-linearity into the model. To address hardware limitations during training, we also provide a lightweight version EAST-LITE. Compared to the EAST with 48 channel numbers, the

Table 4.1: Overall ΔPSNR Of Different Models at QP=22,27,32,37

QP	Sequence	STDF-R3	QECF	CAT	EAST-LITE	EAST
	BasketballScreen	0.370	0.339	0.304	0.458	**0.494**
	ChineseEditing	0.273	0.244	0.200	0.525	**0.567**
	EnglishDocumentEditing	0.867	0.770	0.951	0.991	**1.007**
	MissionControlClip2	0.545	0.551	0.535	0.612	**0.67**
	MissionControlClip3	0.492	0.503	0.477	0.573	**0.584**
37	Paperpdf	1.281	1.225	1.421	1.500	**1.556**
	Sephora	0.779	0.831	0.864	1.069	**1.162**
	mixframe	0.377	0.418	0.379	0.495	**0.564**
	mixvideo	0.301	0.365	0.329	0.528	**0.637**
	scSlideShow	0.914	0.91	0.878	1.076	**1.096**
	scconsole	-0.135	0.095	0.080	0.134	**0.446**
	scdesktop	0.077	0.179	0.339	0.373	**0.618**
	scmap	0.453	0.373	0.416	0.476	**0.503**
	scprogramming	0.406	0.427	0.403	0.545	**0.589**
	scrobot	0.111	0.107	0.120	0.071	**0.136**
	scwebbrowsing	1.008	0.907	0.969	**1.107**	1.079
	Average	0.507	0.515	0.542	0.658	**0.732**
32	Average	0.488	0.478	0.484	0.619	**0.665**
27	Average	0.437	0.468	0.395	0.502	**0.578**
22	Average	0.373	0.431	0.371	0.432	**0.477**

Table 4.2: Overall ΔSSIM(10^{-3}) Of Different Models at QP=22,27,32,37

QP	Sequence	STDF-R3	QECF	CAT	EAST-LITE	EAST
	BasketballScreen	3.07	2.89	2.99	3.96	**4.23**
	ChineseEditing	2.22	1.63	1.24	4.17	**4.86**
	EnglishDocumentEditing	2.78	2.70	3.27	3.36	**3.58**
	MissionControlClip2	5.06	4.96	4.98	5.55	**5.85**
	MissionControlClip3	4.17	4.12	4.00	4.56	**5.00**
37	Paperpdf	2.87	2.67	3.08	3.07	**3.16**
	Sephora	2.38	2.34	2.79	3.85	**4.18**
	mixframe	4.96	5.39	5.30	5.22	**6.55**
	mixvideo	3.46	3.51	3.05	4.35	**5.03**
	scSlideShow	4.02	3.98	4.21	4.59	**4.65**
	scconsole	1.41	2.08	0.94	1.78	**2.91**
	scdesktop	1.12	1.27	1.48	1.36	**2.00**
	scmap	5.71	3.53	6.26	6.91	**8.01**
	scprogramming	4.90	4.93	4.86	5.82	**6.14**
	scrobot	0.10	-0.55	0.82	0.02	**2.61**
	scwebbrowsing	3.28	3.38	3.56	3.69	**3.81**
	Average	3.22	3.05	3.30	3.89	**4.54**
32	Average	1.76	1.74	1.70	1.94	**2.23**
27	Average	0.73	0.83	0.74	0.85	**0.97**
22	Average	0.37	0.38	0.40	0.41	**0.42**

Table 4.3: Overall ΔGFM(10^{-3}) Of Different Models at QP=22,27,32,37

QP	Sequence	STDF-R3	QECF	CAT	EAST-LITE	EAST
	BasketballScreen	3.99	3.881	3.591	4.361	**4.725**
	ChineseEditing	1.724	1.661	0.946	3.399	**3.519**
	EnglishDocumentEditing	2.931	2.732	3.017	3.25	**3.396**
	MissionControlClip2	5.471	5.274	5.145	5.99	**6.363**
	MissionControlClip3	4.881	4.842	4.481	4.926	**5.259**
37	Paperpdf	3.987	3.906	4.163	4.959	**5.154**
	Sephora	4.248	4.481	4.633	6.865	**7.45**
	mixframe	5.993	5.956	5.527	7.507	**7.657**
	mixvideo	4.244	4.226	3.834	5.919	**6.05**
	scSlideShow	8.927	8.665	9.167	10.286	**10.596**
	scconsole	0.47	0.928	0.6	1.165	**1.762**
	scdesktop	0.768	0.839	0.981	1.114	**1.341**
	scmap	8.993	6.331	9.183	9.213	**10.316**
	scprogramming	3.279	3.388	3.082	4.015	**4.317**
	scrobot	2.924	2.437	3.94	-0.1	**4.223**
	scwebbrowsing	5.807	5.936	6.206	6.441	**6.489**
	Average	4.290	4.093	4.281	4.957	**5.539**
32	Average	1.926	1.830	1.713	2.254	**2.525**
27	Average	0.652	0.712	0.632	0.817	**0.932**
22	Average	0.186	0.190	0.181	0.209	**0.215**

Table 4.4: Overall BD-rate(%) Of Different Models at QP=22,27,32,37

Sequences	STDF [15]	QECF [17]	CAT [16]	EAST-LITE	EAST
BasketballScreen	-5.97	-6.27	-5.58	-7.24	**-7.78**
ChineseEditing	-1.39	-1.44	-1.25	-2.45	**-2.86**
EnglishDocumentEditing	-2.67	-2.59	-2.68	-3.12	**-3.33**
MissionControlClip2	-7.25	-7.54	-7.19	-8.37	**-8.84**
MissionControlClip3	-6.02	-6.24	-5.91	-6.78	**-7.19**
Paperpdf	-4.17	-4.53	-4.19	-5.55	**-5.89**
Sephora	-5.81	-6.07	-5.78	-8.16	**-8.66**
mixframe	-2.33	-2.62	-2.30	-3.05	**-3.44**
mixvideo	-2.05	-2.25	-2.00	-3.04	**-3.46**
scSlideShow	-9.94	-10.05	-9.85	-11.48	**-12.09**
scconsole	-1.55	-1.87	-1.46	-1.91	**-2.54**
scdesktop	-0.57	-0.58	-0.64	-0.69	**-0.86**
scmap	-7.22	-6.42	-6.00	-7.33	**-7.98**
scprogramming	-5.81	-6.39	-6.33	-8.24	**-8.89**
scrobot	-2.15	-2.00	**-2.51**	-1.76	-2.21
scwebbrowsing	-3.14	-2.91	-2.90	-2.97	**-3.33**
Average	-4.25	-4.36	-4.16	-5.13	**-5.58**

channel number of convolution in EAST-LITE is set as 32.

Due to the limited number of available screen content sequences within the CTC [99], we gathered additional screen content sequences from other sources [101] and [100]. In addition, we also captured our own screen content sequences, which collectively form our dataset named "PolyUSCCv2" [102]. Our dataset consists of 50 video sequences of various resolutions including 1920×1080, 1680×1050, 1280×720 are adopted. Among these sequences, 28 videos were adopted for training, 6 videos for validation, and the remaining 16 videos for model testing. The test set contains 12 video sequences provided in the CTC [99] and 4 video sequences are our self-capture sequences, none of which is the same as the training set and validation set. The video sequences were encoded using the HEVC reference software HM16.20-SCM8.8 under Low Delay Main SCC (LDMS) configuration as the input for the networks, while the uncompressed raw video sequences were used as the ground-truths of networks. We utilized four Quantization Parameters (QPs) of 22, 27, 32, and 37 for encoding the sequences, and training a separate model for each QP. During training, only the luminance channel (Y channel) of each frame was considered as input. Model construction and training were implemented based on PyTorch. The patch size of each input image and its corresponding ground truth were 128 × 128. To augment our dataset, we randomly selected 300 patches from one frame for each iteration. In our experiments, the learning rate was set to 0.0001 for all QPs. The Adam optimization method [108] was used to train the model for 500 epochs. A computer equipped with Ubuntu 20.04 operating system, an Intel i9-10900K CPU, 64 GB RAM, and NVIDIA 3090Ti GPUs, was used to perform the model training.

4.2.2 Overall Performance

Objective Visual Quality Assessment: In this section, we compare the proposed EAST method and EAST-LITE method with the state-of-the-art video quality enhancement methods, STDF [15], QECF [17], and CAT [16]. We employ PSNR, a widely recognized objective metric, to assess visual quality at the pixel level across all test sequences. Recognizing that the human eye is the ultimate evaluator of visual quality, we also incorporate the SSIM and the Gabor feature-based model (GFM) [109] for the assessment.

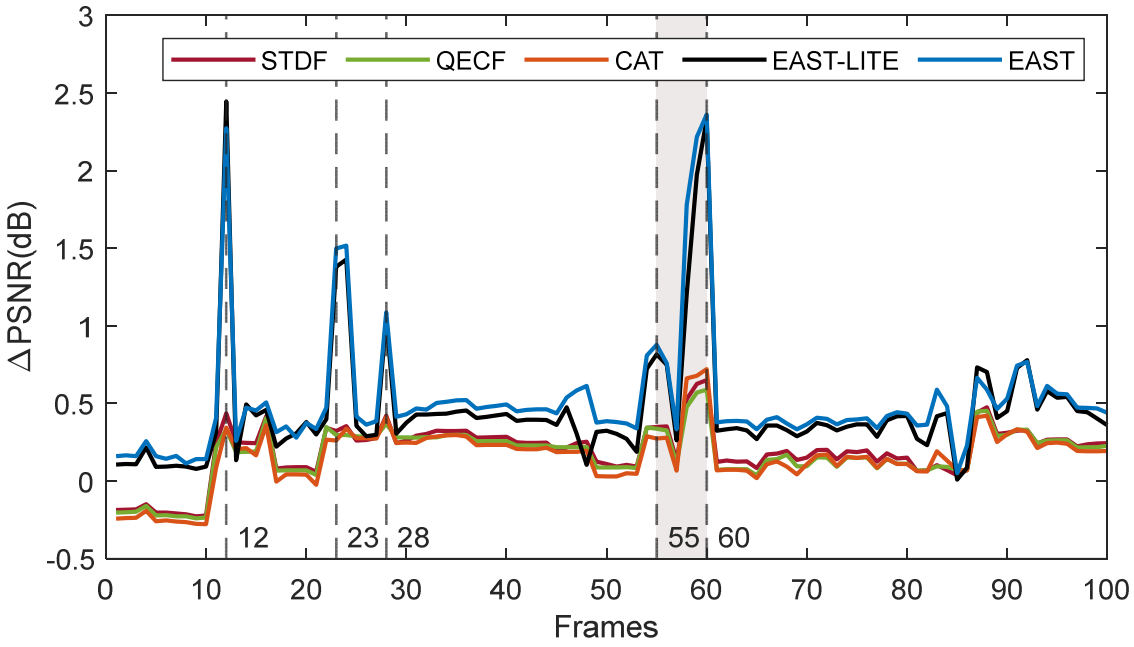
Figure 4.4: Subjective visual quality comparison at QP = 37 on *ChineseEditing*, *mixvideo*, *scwebbrowsing*, and *scmap*.

The GFM is specifically designed to objectively assess screen content in a way that mimics the human visual system's perception. Table 4.1, Table 4.2, and Table 4.3 show the average PSNR improvement (ΔPSNR), the average SSIM improvement (ΔSSIM), and the average GFM improvement (ΔGFM), respectively, over all frames of each test sequence. In both tables, the best PSNR/SSIM/GFM improvement is highlighted in bold, while the underline number represents the second-best PSNR/SSIM/GFM improvement. We can see that our proposed EAST and EAST-LITE outperform other methods in most cases, highlighting the effectiveness of our approach. For instance, when using a QP of 37, our EAST achieves the highest PSNR improvement of 1.556 dB for the *paperpdf* sequence comprising text and graphics. The average PSNR improvement of our EAST is 0.732 dB, which is 11.25% higher than that of EAST-LITE (0.658 dB), 35.06% higher than that of CAT (0.542 dB), 42.14% higher than that of QECF (0.515 dB), and 44.38% higher than that of STDF (0.507 dB). For other QPs (22, 27, and 32), our EAST approach consistently outperforms other state-of-the-art video quality enhancement approaches. Similar trend can be found for ΔSSIM in Table 4.2 and ΔGFM in Table 4.3. It is noted that CAT [16] and STDF [15] utilize deformable convolution to enhance gaming content which is consistent content. However, it is found that our proposed method can also handle gaming content and text content. Compared with the CAT [16] and STDF [15], our EAST can achieve an acceptable ΔPSNR, ΔSSIM, and ΔGFM in gaming content sequence *scrobot*. This demonstrates that our EAST approach not only performs well in reducing the differences incurred at the pixel level but also enhances the quality of frames in the human visual system. To further evaluate the performance, BD-rate [99] is used to indicate the bitrate savings achieved by these models under the equivalent PSNR. The experimental results are compared and tabulated in Table 4.4. Our EAST obtains an average BD-rate savings of 5.58%, while the second-best method EAST-LITE achieves an average BD-rate savings of 5.13%. For the test sequence *scSlideShow* with dramatic motion, up to 12.09% BD-rate saving is obtained for the Y component under LD configuration. We conjecture that our EAST and EAST-LITE effectively restore the high-frequency information, thereby enhancing the quality of decoded frames and reducing the BD-rate.

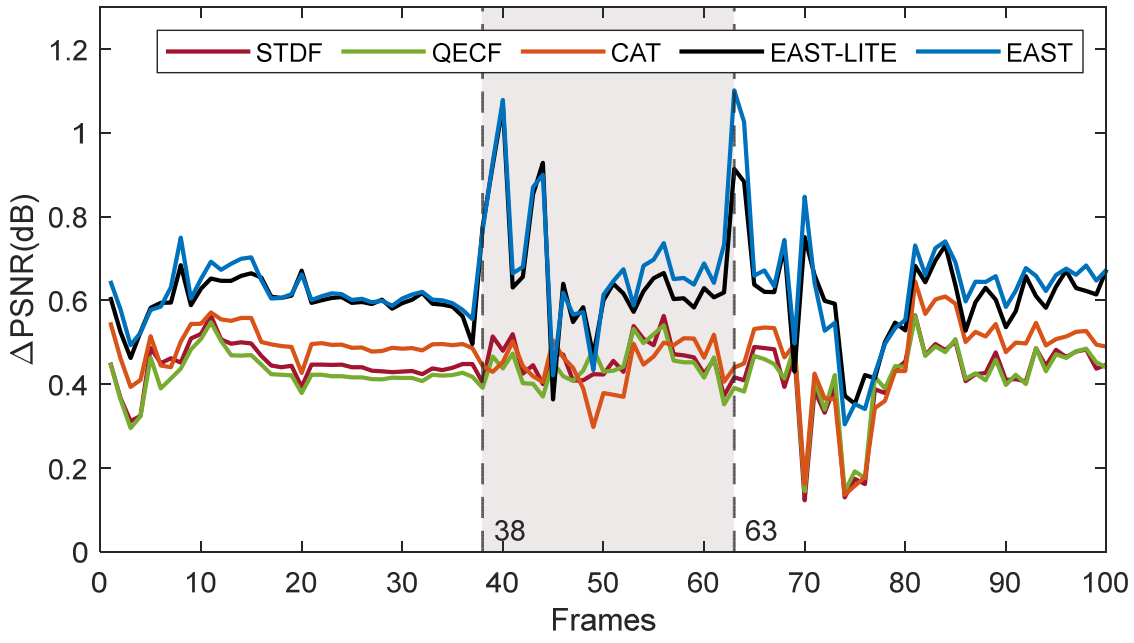
Subjective Visual Quality Comparison: This section compares the subjective

quality of different models. Fig. 4.4 shows the subjective visual quality performance of various models on the sequences *ChineseEditing*, *mixvideo*, *scwebbrowsing*, and *scmap*, all encoded with QP = 37. From this figure, we can clearly see that the reconstructed frames of HM16.20-SCM8.8 exhibit noticeable compression artifacts and suffer from significant loss of high-frequency information details. These artifacts and details cannot be effectively restored by STDF [15], QECF [17], or CAT [16]. As depicted in Fig. 4.4, our proposed EAST and EAST-LITE remove the artifacts and restore the content more effectively than the other models. Taking the *ChineseEditing* sequence as an example, it can be observed that the edges of the background still disappear in other methods, but they are successfully restored by our EAST and EAST-LITE. For *mixvideo* and *scwebbrowsing*, we visualize that the frame during the scene switch situation. There is a loss of high-frequency information, resulting in blurry text and icons appear blurry. However, when applying our proposed approach, these elements become clearer. For *scmap*, the characters are originally blurry and blocking artifacts are presented in the background. After being processed by our EAST and EAST-LITE, the text and the icon are restored, and the artifacts in the background are eliminated. The examples presented in Fig. 4.4 collectively demonstrate the superiority of EAST over other models in terms of subjective visual quality. Once again, this showcases the ability of our EAST model to effectively restore high-frequency information and handle scenarios involving scene switches.

Quality Evaluation on Frame Freezing, Dramatic Motion, and Scene Switches: To evaluate the capability of our proposed EAST in handling frame freezing, dramatic motion, and scene switches, four different types of screen content videos were selected to compute the ΔPSNR curves for STDF, QECF, CAT, and our proposed methods. The *ChineseEditing* sequence contains a lot of font deformation and text editing, while the *scprogramming* sequence involves pop-up windows and window switching. These dynamic motions are commonly seen in our daily life and can pose difficulties for video quality enhancement algorithms. Additionally, the *mixvideo* sequence is composed of spliced videos from CTC [99], allowing us to evaluate the performance of our methods in scenarios involving frame freezing and abrupt scene transitions. On the other hand, the *BasketballScreen* sequence represents a scenario with slow motion. The results are

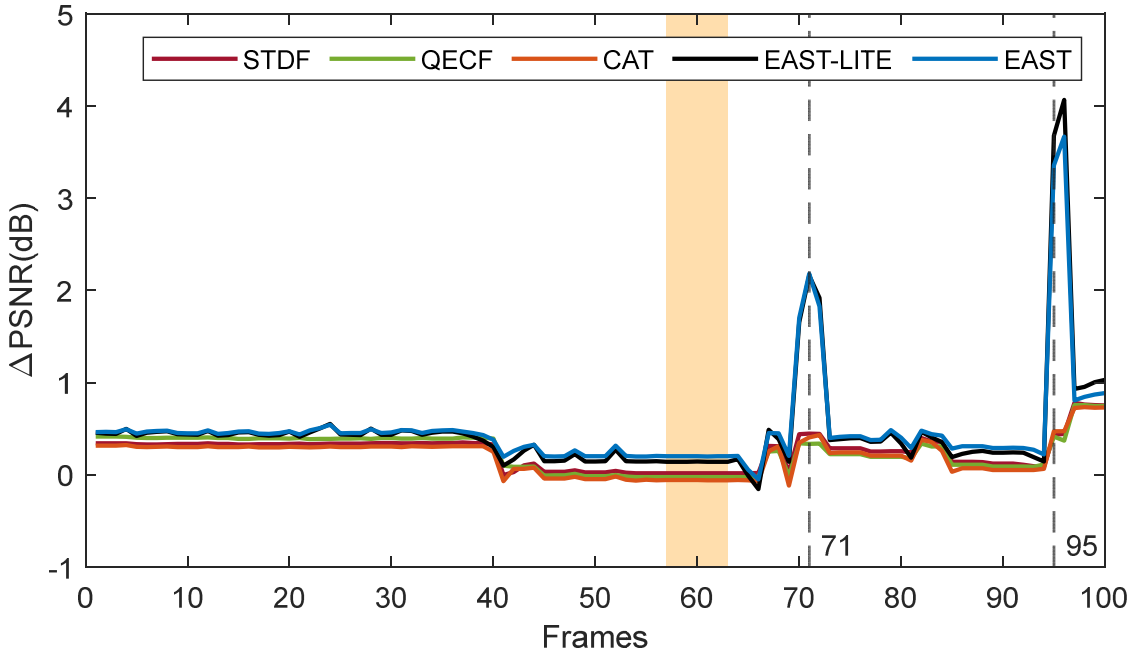


(a)

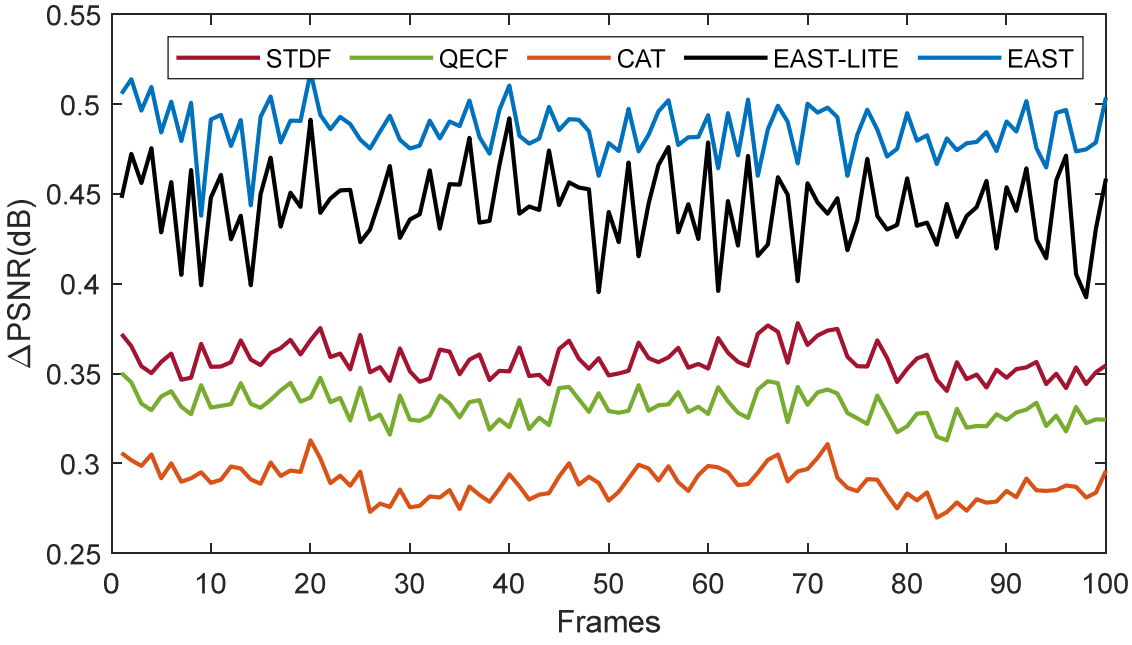


(b)

Figure 4.5: ΔPSNR curves of STDF, QECF, CAT, our EAST-LITE, and our EAST method for sequences, (a) *ChineseEditing*, (b) *scprogramming*.



(c)



(d)

Figure 4.5: ΔPSNR curves of STDF, QECF, CAT, our EAST-LITE, and our EAST method for sequences, (c) *mixvideo*, and (d) *BasketballScreen*.

shown in Fig. 4.5, where employs dashed lines to indicate scene switch frames. Moreover, frames exhibiting dynamic motion and frame freezing are distinguished by gray and yellow shadow regions, respectively. In Fig. 4.5(a), in the *ChineseEditing* sequence, our proposed method exhibits superior performance, particularly during the period of dynamic text deformation from frame 55 to frame 60. The result in Fig. 4.5(b) demonstrates that our proposed EAST consistently outperforms the others from frame 38 to frame 63 in the *scprogramming* sequence. This shadow region encompasses window switches and a pop-up window. It can demonstrate that our proposed method can achieve significant PSNR improvement during periods of dramatic motion. In Fig. 4.5(c), it is evident that frame 71 and frame 95 represent as the switch points between two videos in the *mixvideo* sequence, with frame freezing occurring around frame 60. Notably, our proposed method demonstrates a substantial improvement during these transition points, highlighting its effectiveness in handling abrupt scene transitions. Moreover, our proposed method also achieves PSNR improvement in the case of frame freezing. Furthermore, Fig. 4.5(d) illustrates the performance of EAST on the *BasketballScreen* sequence, characterized by slight motion. We can observe that our method consistently achieves stable quality improvement throughout the entire video sequence. In summary, our approach not only outperforms the other methods during critical periods of dynamic content and scene transitions, but also proves effective in enhancing the quality of videos with slight motion. This robustness to screen content videos highlights the versatility and reliability of our method.

Model parameters and computational complexity: Table 4.5 shows the average ΔPSNR against model parameters for EAST, EAST-LITE, STDF, QECF, and CAT methods. The results are averaged over all the test sequences. We can see that the performance of EAST-LITE outperforms the state-of the-art methods in terms of performance, while maintaining a lower number parameters compared to QECF and CAT. It demonstrates the effectiveness of our proposed methods. In Table 4.6, we also compare the

Table 4.5: Comparision of Model Size

Model	STDF [15]	QECF [17]	CAT [16]	EAST-LITE	EAST
PSNR improvement	0.507	0.515	0.542	0.658	0.732
Parameters (KB)	364.510	773.313	848.546	599.896	921.854

Table 4.6: Comparision of Running Time Per Frame

Frame Size	STDF [15]	QECF [17]	CAT [16]	EAST-LITE	EAST
1280x720	62.392ms	174.132ms	115.740ms	111.967ms	156.197ms
1920x1080	131.676ms	389.500ms	254.043ms	250.465ms	352.132ms
2560x1440	229.589ms	693.045ms	440.913ms	431.750ms	606.510ms

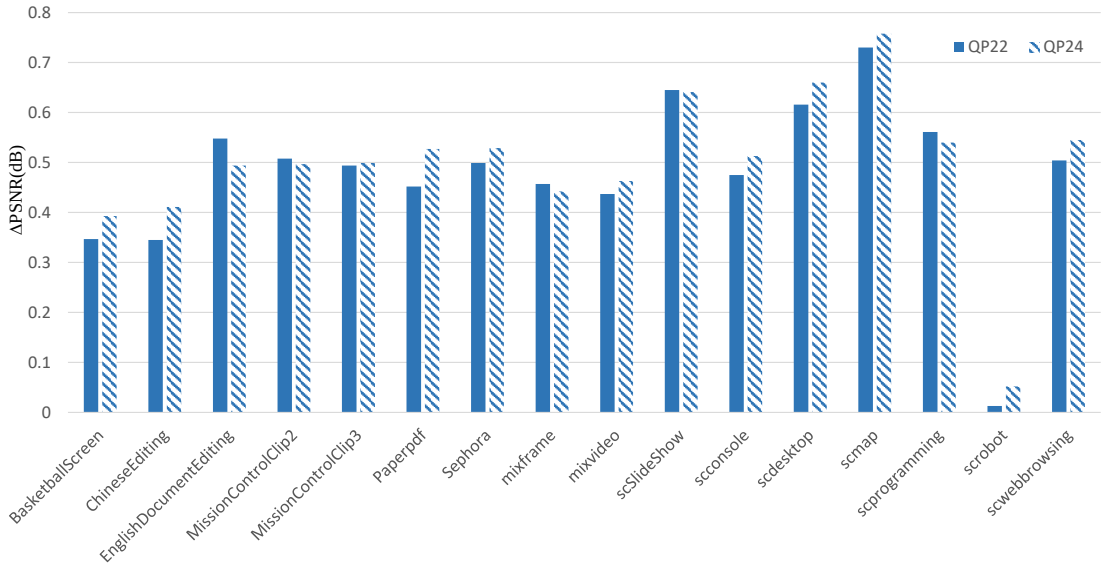
overall time consumption for enhancing one frame at different resolutions using different methods. The results demonstrate that our EAST-LITE outperforms QECF and CAT in both running time and ΔPSNR. It can be concluded that our proposed approach achieves a better trade-off between coding efficiency, model size and computational complexity. In other words, our EAST is more model-efficient.

Quality enhancement at various QPs: To verify the generalization ability of the EAST model across different QPs, we conducted additional encoding of all test sequences at QPs of 24, 29, 34, and 39, while training the model at different QPs: QP = 22, 27, 32, and 37. The performance in terms of ΔPSNR is presented in Fig. 4.6. Fig. 4.6(a) shows the PSNR improvement of the model trained at QP = 22 and tested at QP = 22 and 24. In Fig. 4.6(b), the model is trained at QP = 27 and tested at QP = 27 and 29. Similarly, Fig. 4.6(c) and Fig. 4.6(d) show ΔPSNR of the model trained at QP =32 and 37, respectively, and tested at different QPs = 32 and 34, 37 and 39. As shown in this figure, each trained model can obtain good quality enhancement on decoded videos at adjacent QPs, thereby verifying the model's generalization ability at various QPs.

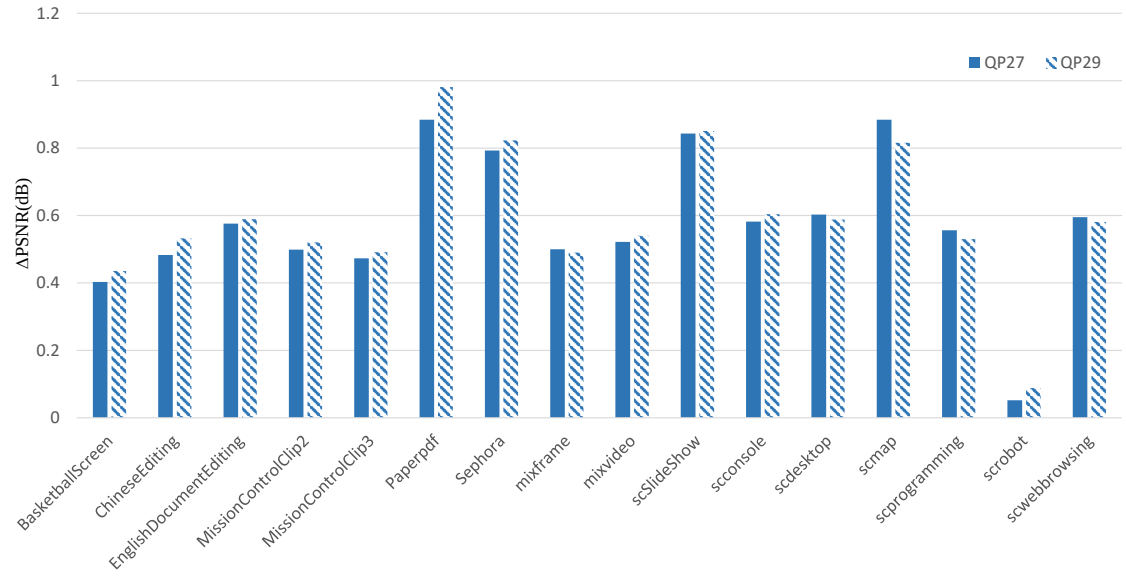
4.2.3 Ablation Study

In this section, we conducted several ablation experiments on the EAST model to analyze its effectiveness in handling frame freezing and scene switches. To evaluate the performance, we present the PSNR curve for frames affected by frame freezing and scene switches. The impact of the attention block is evaluated by calculating the average PSNR improvement across all validation sequences.

Spatio-temporal feature extraction: As discussed in Section 4.1.3, the STFE module can adaptively handle scene switches and slow-motion. To demonstrate its effectiveness, we conducted ablation experiment comparing different combinations of feature layers F_L, F_{S1}, and F_{S2}. Fig. 4.7 illustrates this comparison, where dashed lines indicate

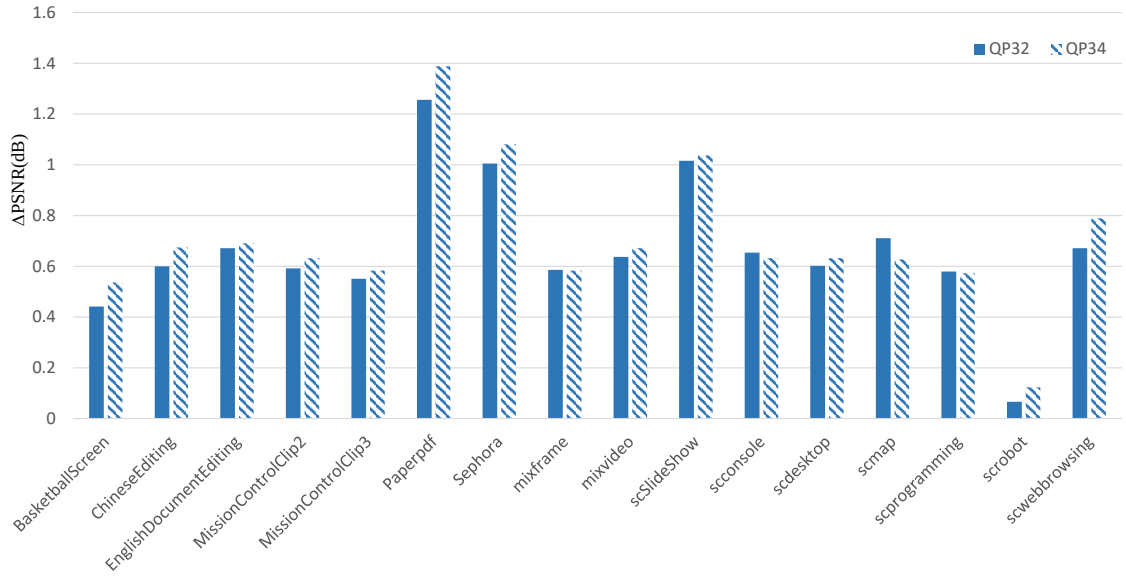


(a)

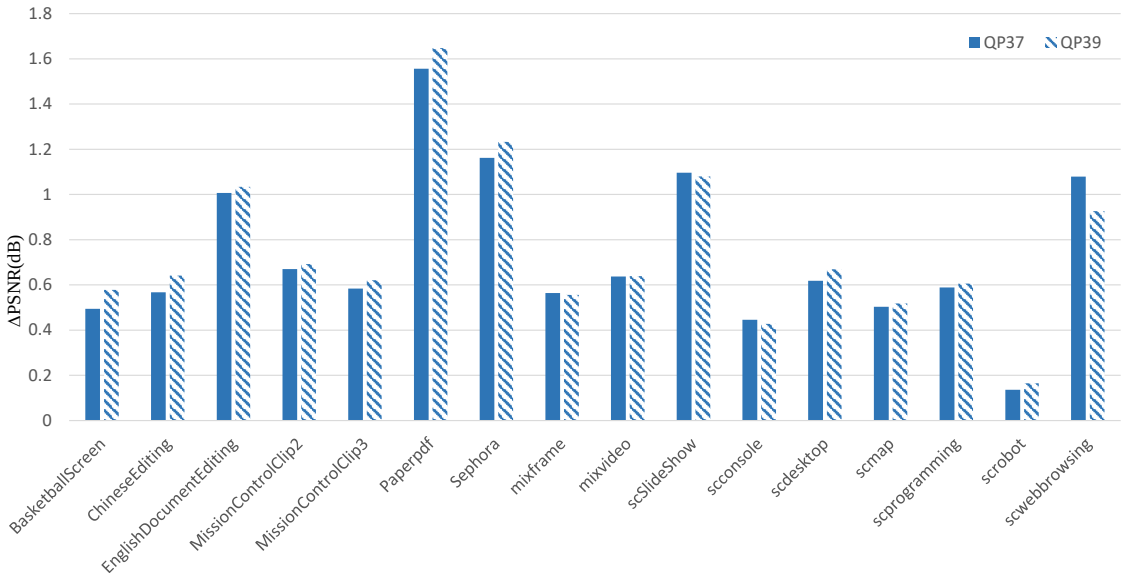


(b)

Figure 4.6: ΔPSNR of the model trained and tested at different QPs. (a) Trained at QP=22, Tested at QP=22 and 24, (b) Trained at QP=27, Tested at QP=27 and 29.



(c)



(d)

Figure 4.6: ΔPSNR of the model trained and tested at different QPs. (c) Trained at QP=32, Tested at QP=32 and 34, and (d) Trained at QP=37, Tested at QP=37 and 39.

scene switch points and the shadow region represents slow motion. It is evident that relying solely on FL input (referred to as "EAST-FL") maintains comparable performance during slow-motion, but fails to address scene switches effectively. Conversely, when using only F_{S1} and F_{S2} inputs (referred to as "EAST-FS"), stable PSNR is achieved during scene switches, but not during slow-motion. These observations confirm the effectiveness of the STFE module in handling both scene switches and slow-motion scenarios.

Table 4.7 presents the results of the ablation experiment for the CAB in the STFE stage. The "No-CAB" column represents the model with the CAB removed, resulting in a PSNR drop of 0.023 dB. This significant decline in performance suggests that without channel attention for neighbor frames, our subsequent modules lack sufficient and effective feature information, leading to an inability to exclude the influence from neighbor frames. To further validate the significance of using channel attention in the STFE stage, we applied the traditional channel and spatial attention module proposed in CBAM [61]. The result is shown in the "CBAM" column. Moreover, our proposed CSAB as in Fig. 4.3 was also applied into the spatio-temporal feature extraction stage as comparison (the "CSAB" column in Table 4.7). The results show that model fails to achieve its optimal performance using CBAM and CSAB, highlighting the importance of incorporating spatial information with edge information at a later stage.

Edge aware block: As discussed in Section 4.1.4, our EAST model, when combined with edge information, can further enhance the quality of compressed frames during frame freezing scenarios. Fig. 4.8 displays the results of the ablation experiment conducted on the EAB for frame freezing. "EAST-NoEdge" represents the EAST model without EAB, resulting in a significant decrease in PSNR during frame freezing compared to other scenarios. This observation highlights the significant contribution of our proposed EAB in enhancing frame quality during frame freezing situations.

Channel and spatial attention block in spatio-temporal feature fusion: Table 4.8 presents the results of the ablation experiment for the CSAB in the STFF stage. When we remove the CSAB, the "No-CSAB" column in Table 4.8 reveals that a ΔPSNR loss of approximately 0.048 dB compared to our method. Besides, we applied the channel attention module of SE-net [60] as the global attention module and presented the result in

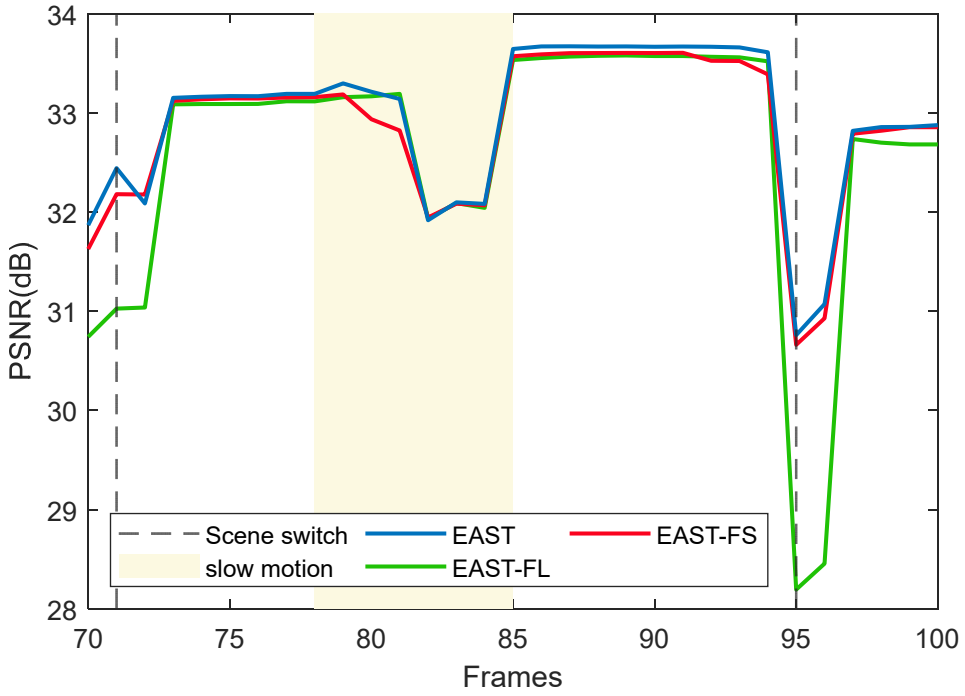


Figure 4.7: PSNR curves of screen content video *mixvideo*.

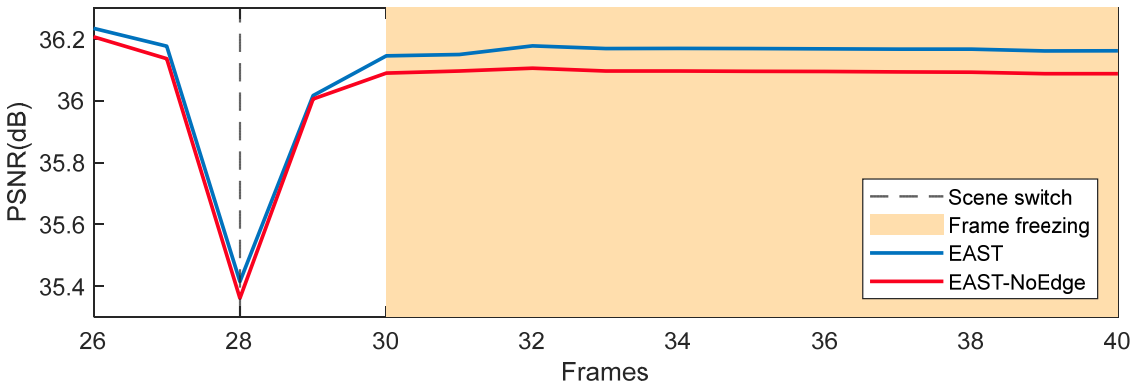


Figure 4.8: PSNR curves of screen content video *Paperpdf*.

the "CA" column, demonstrating a ΔPSNR drop of about 0.024 dB. This indicates that using the combination of channel and spatial attention in the STFF stage makes our model pay more attention to the features of the target frame in the spatio-temporal domain after the combination of edge information. To further validate the effectiveness of our proposed CSAB, we conducted experiments using the channel and spatial attention module of CBAM as an alternative. The results, shown in the "CBAM" column, indicate lower performance compared to our proposed method. This is because CBAM utilizes average pooling and max-pooling to compress the spatial attention maps, resulting in smoothed features and loss of important information in screen content. In contrast, our proposed approach utilizes convolution to generate the spatial attention maps, which greatly keeps the high-frequency details. Therefore, our proposed method is better suited for screen content quality enhancement tasks. Additionally, removing both the CAB and CSAB, as shown in the "No-CAB-CSAB" column, results in a ΔPSNR loss of approximately 0.051dB (the lowest performance in Table 4.8) compared to our method, illustrating the importance of both the CAB and CSAB in our model.

Table 4.7: Different Attention in Spatio-Temporal Feature Extraction at QP=37

Architectures	No-CAB	CSAB	CBAM [61]	Proposed
ΔPSNR	0.489	0.476	0.473	0.512
Parameter (KB)	874.376	922.773	921.954	921.854

Table 4.8: Different Attention in Spatio-Temporal Feature Fusion at QP=37

Architectures	No-CAB-CSAB	No-CSAB	CA [60]	CBAM [61]	Proposed
ΔPSNR	0.461	0.464	0.488	0.461	0.512
Parameter (KB)	832.234	879.712	921.619	921.719	921.854

4.3 Summary

In this chapter, we studied the different characteristics of screen content and natural videos, both in the spatial and temporal domains. Based on our analysis, we proposed an alignment-free approach to enhance the quality of screen content videos. To address the challenges posed by scene switches, we devised the spatial and temporal feature extraction

method that captures the spatial and temporal features from three different groups of input frames, enabling efficient handling of scene switches. To adaptively emphasize features related to the target frame, we incorporated channel attention, allowing our model to focus on the most relevant information. Besides, we added an EAB to guide the model in removing artifacts and preserving the high-frequency details of the target frame. By incorporating the edge information derived from the spatial features of the target frame into our model, we achieved further quality enhancement, particularly in frame freezing scenario. Subsequently, the channel and spatial attention is utilized to distillate the spatial and temporal features that are specifically relevant to the target frame. Through extensive experiments, we demonstrate that our proposed EAST model outperforms state-of-the-art models in terms of quality enhancement for screen content videos.

Chapter 5

Long Short-term Fusion by Multi-scale Distillation for Screen Content Video Quality Enhancement

5.1 Proposed method

5.1.1 Motivation

Existing multi-frame techniques, such as flow-based alignment [13, 14] and deformable-based alignment [15, 16, 34, 38], fall short in addressing substantial content variations between frames [17]. These methods depend on a prediction network to adjust the positions of neighboring frames, but inaccuracies in this network can impair the effectiveness of the quality enhancement network. Moreover, the alignment-free approach detailed in [17], which extracts high-quality regions from neighboring frames to improve the target frame, often fails during scene switches. This is because traditional methods rely on a fixed set of neighboring frames, which may provide irrelevant information under such circumstances, thereby compromising the model's performance. Therefore, there is a critical need for the development of a new multi-frame video quality enhancement method that effectively addresses the unique challenges posed by scene switches in screen content videos.

Based on the unique characteristics of screen content, we propose a Long Short-term

Fusion by Multi-scale Distillation (LSFMD) method to effectively restore high-frequency details and improve quality during scene switches in compressed screen content videos. This method consists of a long short-term feature extraction module and a high-frequency reconstruction module. The long short-term feature extraction module is designed to retain useful information from neighbor frames while minimizing their impact during scene transitions, allowing the model to enhance video quality despite scene switches and rapid motion. The high-frequency reconstruction module focuses on reconstructing the sharp edges, as the artifacts in screen content (SC) videos predominantly occur around these regions. In the short-term feature extraction stream, we introduce a Similarity-based Neighbor Frame Selector (SNFS) that identifies and selects relevant frames among neighbor frames to minimize disturbances from unrelated frames. This selector ensures that short-term information is extracted from frames with similar content, enhancing the accuracy of the reconstruction. The selected frames pass through the Multi-scale Residual Block (MSRB) to capture short-term features for flat areas and text regions using different kernel sizes, while a 3D Residual Block extracts long-term features for contextual information. To effectively fuse short-term and long-term information, we design a Multi-scale Hierarchical Feature Distillation (MHFD) mechanism. This mechanism transforms features from different scales to refine the hierarchical features at various network depths using local-global attention to distill significant features to the target frame which can capture more information for uneven noise distribution in screen content videos. This allows for better handling of scene switches and consistency in scenes. The fused features are then used as input for our proposed High-Frequency Reconstruction Block (HFRB), which utilizes the scale-space theory [110, 111] to factorize the feature map tensors and extract the high-frequency information to guide the model in restoring fine details of the target frame. This approach ensures the preservation and enhancement of critical high-frequency details, resulting in better video quality.

The main contributions of this work are summarized below:

- To the best of our knowledge, our proposed LSFMD is the first approach in screen content video quality enhancement to extract and fuse the long short-term features in the corresponding frames to improve frame quality during scene switches and

restore the high-frequency detail.

- Instead of using a fixed set of neighbor frames to enhance the target frame, an SNFS is proposed to dynamically identify and select the most relevant frames based on content similarity. This adaptive frame selection mechanism minimizes the disturbance from unrelated frames, enhancing the accuracy of the reconstruction.

- To avoid the loss of features with the depth of the network, we propose the MHFD to capture the correlation of hierarchical features between short-term and long-term feature extraction streams to distillate the useful information related to the target frame, making the reconstructed frame more high-quality.

- Different from the conventional reconstruction part using vanilla convolution, the HFRB is proposed to parallelly reuse the high-frequency information of the target frame to adaptively restore the high-frequency details of the reconstructed frame.

5.1.2 Overview of the Framework

Our LSFMD model, as shown in Fig. 5.1, aims to remove artifacts in screen content videos that involve numerous dramatic motion and scene switch scenarios. In this context, we denote a low-quality frame at time t as $I_t^{LQ} \in \mathbb{R}^{H\times W}$, where H and W indicate the vertical and horizontal resolutions of the frame. The main objective of LSFMD is to enhance the quality of I_t^{LQ} by effectively using both short-term and long-term temporal information. To achieve this, our model considers the preceding and succeeding $R = 2$ frames as reference frames to capture the necessary temporal context. The enhanced high-quality frame $\tilde{I}_t^{HQ} \in \mathbb{R}^{H\times W}$ can be expressed as

$$\tilde{I}_t^{HQ} = H_{LSFMD}(\{I_{t-R}^{LQ}, ..., I_t^{LQ}, ..., I_{t+R}^{LQ}\}) \tag{5.1}$$

where $H_{LSFMD}(\cdot)$ represents the proposed LSFMD, $\{I_{t-R}^{LQ}, ..., I_t^{LQ}, ..., I_{t+R}^{LQ}\}$ represents the group of the $2R + 1$ input frames. Next, we will discuss our proposed framework in Fig. 5.1, which comprises two modules: a long short-term feature extraction module and a high-frequency reconstruction module. In the long short-term feature extraction module, we construct two streams: a short-term feature extraction stream and a long-

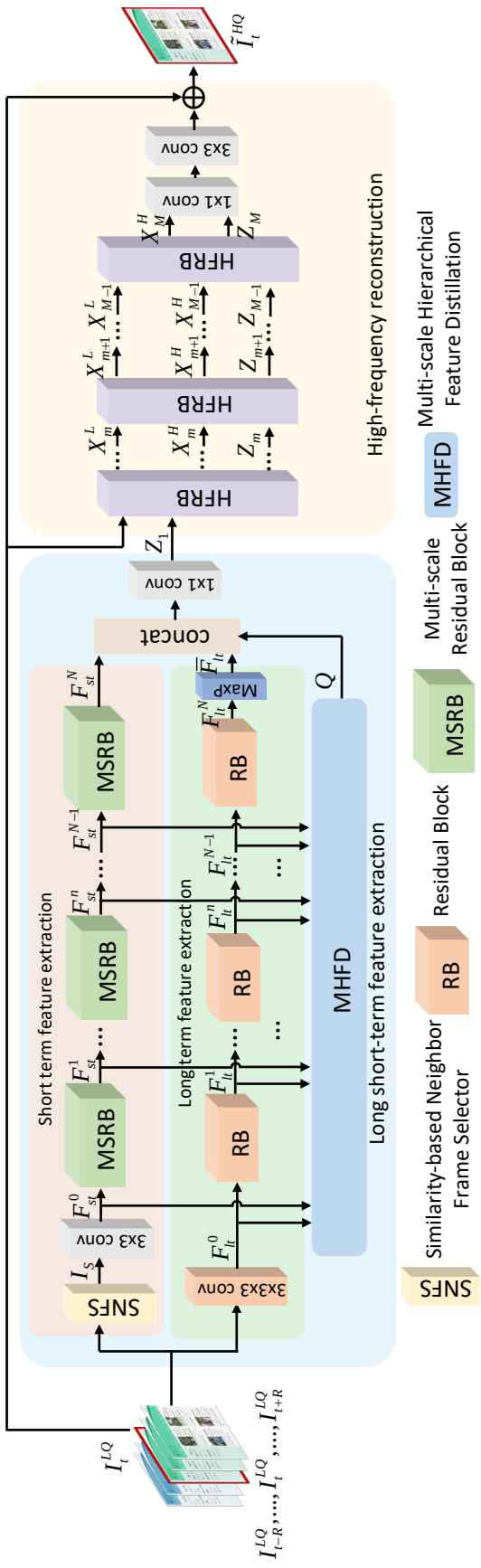


Figure 5.1: Our proposed LSFMD structure, which contains long short-term feature extraction, multi-scale hierarchical feature distillation, and high-frequency reconstruction.

term feature extraction stream. These streams aim to extract short-term and long-term information from input frames of varying lengths. In addition, a multi-scale hierarchical feature distillation (MHFD) is proposed to enhance the reusability and effectiveness of the short-term and long-term features. This approach enables us to handle the scene switch situation adaptively. By assigning weights to the features in an adaptive manner, our network can effectively learn the correlations between short-term and long-term features. In the reconstruction part, we focus on reconstructing the high-frequency information of the target frame. This component plays a crucial role in enhancing the visual quality of the output, particularly in preserving and enhancing the fine details that are often lost during compression. We will provide a detailed explanation of each component within our LSFMD frame in the following subsections.

5.1.3 Long Short-term Feature Extraction

The utilization of single-stream deep neural networks has been widely used for video quality enhancement [15–17]. However, as the depth of the neural network increases, the presence of unrelated features from neighbor frames can hinder the model's ability to effectively learn and extract the relevant information related to the target frame. This issue becomes particularly problematic in scenes with rapid motion and frequent scene switches, as the unrelated features can have a detrimental impact on the quality enhancement of the target frame. To tackle this challenge, it is crucial to focus on the useful and related features of the target frame, especially during situations involving rapid motion and scene switches. Consequently, in contrast to the traditional single-steam method, we propose a long short-term feature extraction stream, where the short-term stream provides the relevant features to assist the long-term stream, enabling a more focused and effective analysis of the target frame.

Within the long-term feature extraction stream, using the 3D Residual Block to extract the long-term feature allows the network to understand the video content in spatial and temporal domain which is crucial for maintaining the integrity of text and graphics across consecutive frames. The structure of the long-term feature extraction stream is shown in Fig. 5.1. We first transform the input sequence to the feature domain by

applying a 3D convolution layer to obtain the initial feature F_{lt}^0 as:

$$F_{lt}^0 = Conv_{3\times3\times3}(\{I_{t-R}^{LQ}, ..., I_t^{LQ}, ..., I_{t+R}^{LQ}\}) \tag{5.2}$$

where $Conv_{3\times3\times3}(\cdot)$ denotes the $3\times3\times3$ convolution layer. Then stacked Residual Blocks compute the features as:

$$F_{lt}^n = H_{RB}^n(F_{lt}^{n-1}), n \in [1, N] \tag{5.3}$$

$$\bar{F}_{lt} = MaxP(F_{lt}^N) \tag{5.4}$$

where N is the total number of residual blocks in the long-term feature extraction, F_{lt}^n represents the extracted features after the n^{th} residual blocks $H_{RB}^n(\cdot)$, and $MaxP(\cdot)$ denotes the maxpooling, which is utilized to transform the feature domain. The output of the long-term feature extraction stream, denoted as $\bar{F}_{lt}$, is obtained by passing the features through the N^{th} residual block $H_{RB}^n(\cdot)$, followed by $MaxP(\cdot)$.

This output, $\bar{F}_{lt}$, encapsulates the contextual information, but special attention must be given to flat areas and repetitive text regions, which are commonly found in screen content videos. These regions require different scale filters to effectively capture more useful information. Inspired by the Multi-Scale Residual Block (MSRB) [112] used in image SR, the short-term information extraction stream utilizes the MSRB to adaptively detect features at different scales. For flat areas, larger filters can be used to capture broader context, which is significant for these regions. In contrast, sharp edges, such as those found in text, are critical features that need to be preserved in screen content videos to maintain readability. To address this, the MSRB employs smaller kernels to capture the high-frequency details associated with text edges, ensuring that the sharpness and clarity of text are retained in the reconstructed frame. The structure of the short-term feature extraction stream is summarized as follows:

$$I_S = H_{SNFS}(\{I_{t-R}^{LQ}, ..., I_t^{LQ}, ..., I_{t+R}^{LQ}\}) \tag{5.5}$$

$$F_{st}^0 = Conv_{3\times3}(I_S) \tag{5.6}$$

$$F_{st}^{n} = H_{MSRB}^{n}(F_{st}^{n-1}), n \in [1, N] \tag{5.7}$$

where $Conv_{3\times3}(\cdot)$ denotes the 3×3 convolution layer. Moreover, H_{SNFS} represents the Similarity-based Neighbor Frame Selector, which identifies the most relevant short-term neighbor frames to the target frame I_S, as discussed in the next subsection, and F_{st}^{n} represents the extracted features after the n^{th} MSRB $H_{MSRB}^{n}(\cdot)$, respectively. We can obtain the output F_{st}^{N} of the short-term feature extraction stream by stacking the MSRB.

Similarity-based Neighbor Frame Selector: Extracting the short-term feature from the shorter input can reduce the disturbance from unrelated neighbor frames. However, during scene transitions, the fixed window for choosing neighbor frames may introduce irrelevant information. To further enhance the frame quality during scene switches, the proposed method incorporates a similarity-based neighbor frame selector (SNFS) in the short-term feature extraction stream. In the SNFS, we employ a sliding window to separate the input frames $\{I_{t-2}^{LQ}, ..., I_{t}^{LQ}, ..., I_{t+2}^{LQ}\}$ to different groups as shown in Fig. 5.2. Group a denotes $\{I_{t-2}^{LQ}, I_{t-1}^{LQ}, I_{t}^{LQ}\}$, group b denotes $\{I_{t-1}^{LQ}, I_{t}^{LQ}, I_{t+1}^{LQ}\}$, and group c denotes $\{I_{t}^{LQ}, I_{t+2}^{LQ}, I_{t+2}^{LQ}\}$, SNFS, then calculates the pearson correlation coefficient [113, 114] between each neighbor frame and target frame in each group. Pearson correlation here is the covariance of the two variables divided by the product of their standard deviations, giving a score in $[-1, 1]$, where higher means more similar content. This calculation allows for the selective identification of frames that are most relevant and pertinent to the target frame. A larger pearson correlation coefficient indicates a greater degree of similarity. In summary, the working process of the proposed SNFS is operated as:

$$\begin{cases} P_a = pearson(I_{t-2}^{LQ}, I_{t}^{LQ}) + pearson(I_{t-1}^{LQ}, I_{t}^{LQ}), \\ P_b = pearson(I_{t-1}^{LQ}, I_{t}^{LQ}) + pearson(I_{t+1}^{LQ}, I_{t}^{LQ}), \\ P_c = pearson(I_{t+1}^{LQ}, I_{t}^{LQ}) + pearson(I_{t+2}^{LQ}, I_{t}^{LQ}), \\ P = \max(P_a, P_b, P_c) \end{cases} \tag{5.8}$$

where $pearson(\cdot)$ denotes the operation to calculate the pearson correlation coefficient between the neighbor frames and the target frame, and P denotes the maximum value among P_a, P_b, and P_c. Once P is determined, the SNFS chooses the group with the larger

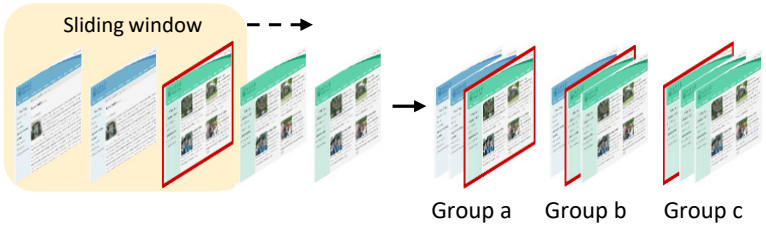


Figure 5.2: SNFS, Group a: $\{I_{t-2}^{LQ}, I_{t-1}^{LQ}, I_{t}^{LQ}\}$, Group b: $\{I_{t-1}^{LQ}, I_{t}^{LQ}, I_{t+1}^{LQ}\}$, Group c: $\{I_{t}^{LQ}, I_{t+2}^{LQ}, I_{t+2}^{LQ}\}$

pearson correlation as the input I_S in Eq. (5.5). This adaptive selection enables quality enhancement, especially in the context of scene switches and fast motion, where the fixed-window approach may introduce irrelevant information. By adopting the pearson correlation-based frame similarity evaluation, the SNFS can effectively identify the most relevant neighbor frames to the target frame, ensuring that the short-term feature extraction stream has access to the most pertinent information for improving the overall video quality.

5.1.4 Multi-scale Hierarchical Feature Distillation

As the depth of the network increases, the extracted short-term and long-term features will gradually disappear during the conduction process. Therefore, taking advantage of hierarchical features becomes crucial for significantly improving model performance. However, many existing models overlook the importance of hierarchical features, as shown in Fig. 5.3 (a), resulting in sub-optimal results. Moreover, simply concatenating all hierarchical features, as in Fig. 5.3 (b), fails to eliminate redundant features, resulting in inefficient video reconstruction. Therefore, an effective method that can exploit hierarchical features and eliminate redundant features is crucial for screen content video quality enhancement. To address this, our proposed MHFD introduces two key components: the Feature Transformation Strategy and the Local-global Channel Attention Mechanism.

Feature Transformation Strategy: The feature transformation component is specially designed to refine the hierarchical features at different network depths through a series of non-linear transformations. This process aims to enhance the representational power of the network. To achieve this, we employ a series of convolutional layers:

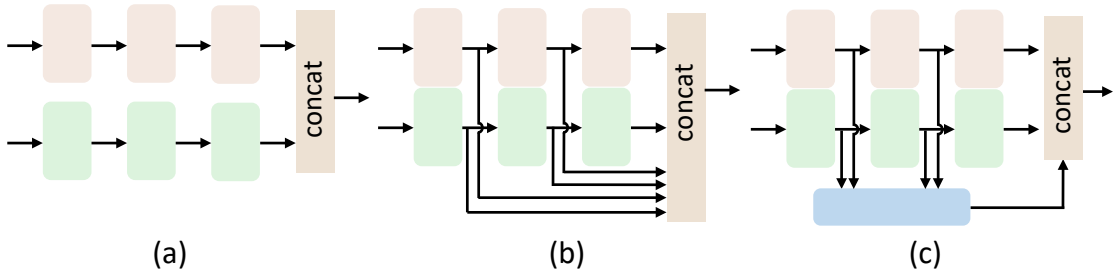


Figure 5.3: Comparisons of different hierarchical feature utilization methods, (a) Structure A, (b) Structure B, and (c) Structure C.

1. Initial feature combination: After obtaining the hierarchical feature from the first convolution layer, we utilize a 1×1 convolution layer to combine the short-term and long-term features.

2. Shallow feature extraction: Subsequently, a 5×5 convolution layer is employed to extract the shallow features. The use of a larger 5×5 receptive field ensures the retention and amplification of salient features.

3. Deeper feature processing: The remaining hierarchical features are processed by a combination of 1×1 and 3×3 convolution layers. This combination allows for capturing finer details by utilizing a smaller receptive field.

The process can be summarized as:

$$\begin{aligned} \tilde{F}^n &= Conv_{k\times k}(Conv_{1\times 1}([MaxP(F_g^n), F_l^n])) \\ k &= \begin{cases} 5, & n = 0 \\ 3, & otherwise \end{cases} \end{aligned} \tag{5.9}$$

where $n = 0, \cdots, N-1$, $\tilde{F}^n$ denotes the feature obtained from the n^{th} feature transformation branch, $Conv_{k\times k}(\cdot)$ presents the $k \times k$ convolution layer, and $[\cdot,\ \cdot]$ denotes the concatenation operation.

Local-global Channel Attention Mechanisms: As in Fig. 5.4, the inputs of MHFD are the hierarchical features obtained through convolution at different scales. However, these features may contain redundancy information. To further distillate the useful information in the target frame, we design a local-global attention mechanism that

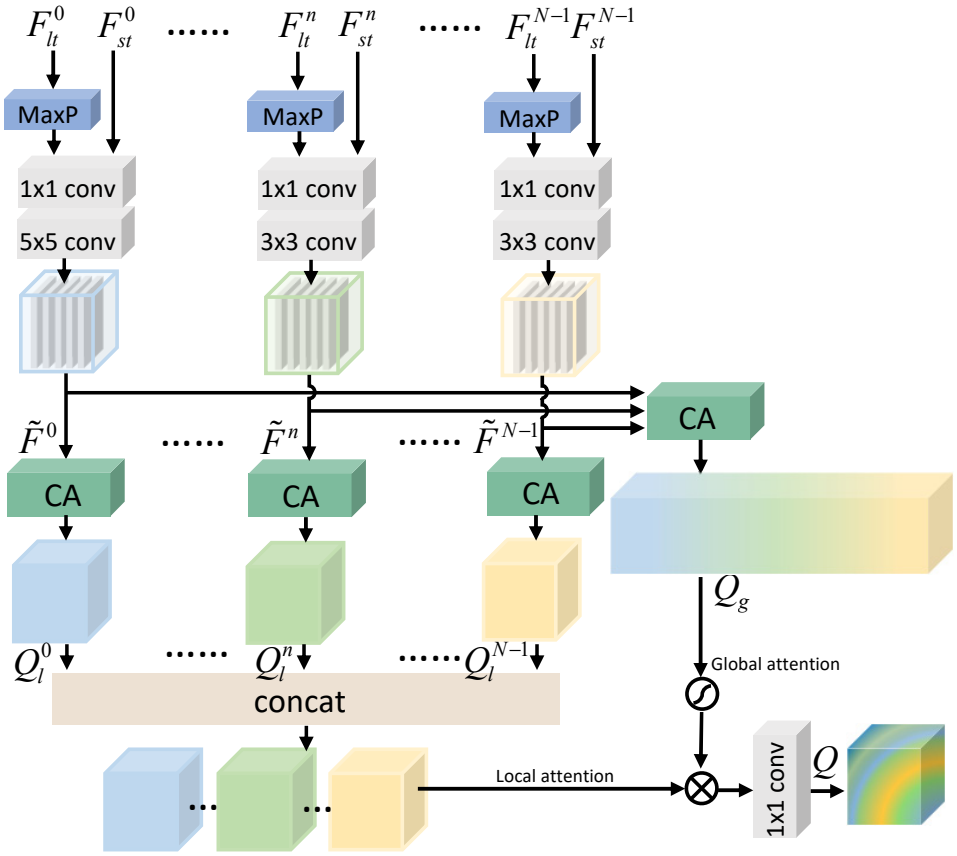


Figure 5.4: Multi-scale hierarchical feature distillation (MHFD).

combines the benefits of local attention and global attention. The local channel attention mechanism is tailored to focus on feature maps specific to certain channel locations. This allows the model to prioritize local patterns and textures that are essential for high-quality video reconstruction in each hierarchical feature branch. We utilize the channel attention [60] to generate the attention weight $\tilde{f}^n$, which can be obtained as:

$$H_{CA}(\cdot) = \sigma(Conv_{1\times1}(ReLU(Conv_{1\times1}(AvgP(\cdot))))) \tag{5.10}$$

$$\tilde{f}_l^n = H_{CA}(\tilde{F}^n) \tag{5.11}$$

where $H_{CA}(\cdot)$ denotes the channel attention operation, $\sigma(\cdot)$ is the sigmoid function, $ReLU(\cdot)$ is the ReLU [107] activation function, and $AvgP(\cdot)$ represents the average pooling operation. The attention weights $\tilde{f}^n$ indicate the sensitivity of different features in the

n^{th} feature transformation branch. Hence, local attention feature Q_l^n can be computed as:

$$Q_l^n = \tilde{f}_l^n \cdot \tilde{F}^n \tag{5.12}$$

On the other hand, the global channel attention mechanism offers a broader perspective by considering the entire channel extent of the feature maps. By assigning attention weights across different hierarchical feature branches, we can prevent the loss of high-frequency hierarchical features as the network depth increases. The synergy between local and global channel attention mechanisms facilitates a more dynamic and context-aware feature distillation. The global attention feature Q_g can be obtained as:

$$\tilde{f}_g = H_{CA}([\tilde{F}^0, \cdots, \tilde{F}^{N-1}]) \tag{5.13}$$

$$Q_g = \tilde{f}_g \cdot [\tilde{F}^0, \cdots, \tilde{F}^{N-1}] \tag{5.14}$$

where $\tilde{f}_g$ denotes the attention weight assigned for all hierarchical features.

Finally, the output feature map Q of the MHFD can be obtained by combining the local and global attention features:

$$Q = Conv_{1\times1}([Q_l^0, \cdots, Q_l^{N-1}] \otimes \sigma(Q_g)) \tag{5.15}$$

where $\otimes$ denotes elementwise multiplication. Here, the output Q of MHFD is the local-global attention-weighted feature which contains the refined information from each scale of long short-term feature extraction. After we obtain the distillated feature Q, we can fuse it with the short-term and long-term features, as depicted in Fig. 5.1, as:

$$Z_1 = Conv_{1\times1}([F_{st}^N, \bar{F}_{lt}, Q]) \tag{5.16}$$

where Z_1 is the output of the long short-term feature extraction module in Fig. 5.1, which contains the long short-term features and multi-scale hierarchical features. Increasing the depth of the network causes the extracted short-term and long-term features to diminish in prominence as they propagate through the model. Therefore, taking advantage of hierarchical features will greatly improve model performance.

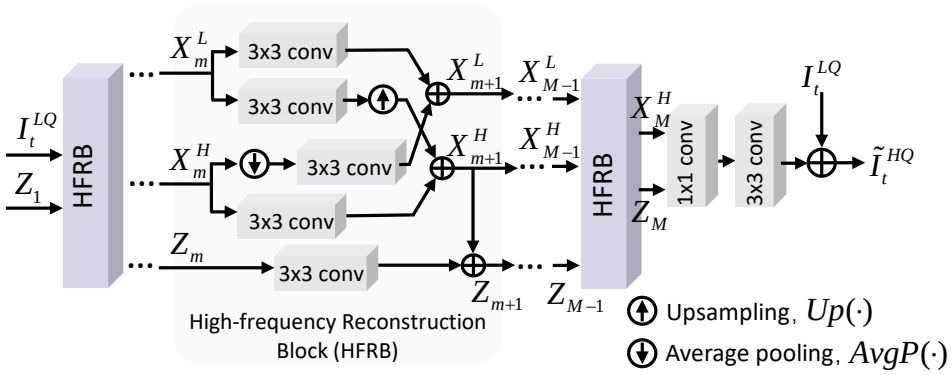


Figure 5.5: The structure of HFRB in the high-frequency reconstruction module.

5.1.5 High-frequency Reconstruction

Conventional reconstruction blocks using vanilla convolution typically focus on reconstructing the frame as a whole. However, this approach sometimes overlooks finer details that contribute significantly to the perceived sharpness and clarity of the screen content image. High-frequency components of this image, such as edges, textures, fonts, and fine structures, contain crucial details. By explicitly incorporating this information into the reconstruction block, we can restore fine details that are often lost during the compression process, which is vital for delivering an enhanced visual experience in screen content videos. To extract the high-frequency feature adaptively, we utilize the scale-space theory introduced in [110, 111] to factorize the feature map tensors into low- and high-frequency groups. The HFRB is then designed in this chapter to effectively integrate this high-frequency information into the reconstruction module. With the stacking of the HFRB in Fig. 5.5, the interaction between the extracted high-frequency features and the reconstructed features dynamically fine-tunes the high-frequency details in parallel. The synergistic effect of this parallel interaction not only aids in restoring the completeness of textures and edges but also enhances the overall quality of the reconstructed feature.

Different from the traditional reconstruction part using a single input from the previous layer, our HFRB in Fig. 5.5 combines features from the previous layer and extra features extracted from the target frame. Let us denote the input feature tensors to the m^{th} HFRB, where $m = 1, \cdots, M$, as:

1. $X_m^H \in \mathbb{R}^{(1-\alpha)c_f \times h \times w}$: high-frequency feature maps extracted from the target frame.

2. $X_m^L \in \mathbb{R}^{\alpha c_f \times \frac{h}{2} \times \frac{w}{2}}$: low-frequency feature maps extracted from the target frame.

3. $Z_m \in \mathbb{R}^{c \times h \times w}$: features from the long short-term extraction module and the high-frequency feature from the target frame excepting the input of the first HFRB, which is Z_1 in Eq. (5.16).

where h and w denote the spatial dimensions, and c and c_f denote the channel number where $c_f = 2c$ apart from the last HFRB. In the last HFRB, c_f is equal to c for the channel matching. Here, $\alpha \in [0, 1]$ denotes the ratio of channels allocated to the low-frequency part. The setting of the α will be introduced in Section 5.2.1.

By explicitly incorporating the target frame information into the reconstruction block, our model can more effectively restore fine details often lost during the training process. The output feature tensors of the m^{th} HFRB is denoted as $X_{m+1}^H \in \mathbb{R}^{(1-\alpha)c_f \times h \times w}$, $X_{m+1}^L \in \mathbb{R}^{\alpha c_f \times \frac{h}{2} \times \frac{w}{2}}$, and $Z_{m+1} \in \mathbb{R}^{c \times h \times w}$.

The process of HFRB is represented as:

$$\{X_{m+1}^H, X_{m+1}^L, Z_{m+1}\} = H_m^{HFRB}(X_m^H, X_m^L, Z_m) \tag{5.17}$$

where

$$\begin{cases} X_{m+1}^H = Conv_{3\times3}(X_m^H) + Up(Conv_{3\times3}(X_m^L)), \\ X_{m+1}^L = Conv_{3\times3}(X_m^L) + Conv_{3\times3}(AvgP(X_m^H)), \\ Z_{m+1} = Conv_{3\times3}(Z_m) + X_{m+1}^H \end{cases} \tag{5.18}$$

where $H_m^{HFRB}(\cdot)$ denotes the m^{th} HFRB and $Up(\cdot)$ denotes the upsampling operation by a scale factor of 2. $Up(\cdot)$ operation denotes the upsampling of the input by a scale factor of 2. The $Up(\cdot)$ and $AvgP(\cdot)$ operations are used for communication between the low-frequency and high-frequency feature groups, which helps adjust the feature dimensions. In the HFRB, the output feature Z_{m+1} encapsulates the high-frequency information from the target frame. This feature is subsequently fed into the next layer to extract deeper features. Concurrently, the high-frequency details from the target frame are fed into the subsequent layer for analysis and extraction of the most pertinent features. The HFRB's capability to handle multiple inputs allows for the parallel extraction and integration of high-frequency information. This enables the LSFM model to dynamically shift its focus

towards these crucial details as it progresses deeper into the network. This adaptive mechanism ensures that the essential high-frequency characteristics from the target frame are not overlooked but are instead emphasized throughout the reconstruction process.

Finally, the reconstructed frame can be represented as:

$$\tilde{I}_t^{HQ} = Conv_{3\times3}(Conv_{1\times1}([Z_M, X_M^H])) + I_t^{LQ} \tag{5.19}$$

where the Z_M and X_M^H are the output of the last HFRB. In the reconstruction module, this high-frequency information is progressively integrated with the major features in the HFRB. The parallel extraction and integration of high-frequency details enable the model to dynamically adjust its focus on these components as the network deepens. This newly adaptive mechanism ensures that the essential details from the target frame are not lost but rather emphasized, leading to improved frame quality.

5.1.6 Training Scheme

To effectively handle the high-frequency information and improve the performance, we adopt the robust Charbonnier loss function in [106, 115] to train our model in an end-to-end manner. The loss function L is represented as:

$$L = \sqrt{\left\| I_t^{HQ} - \tilde{I}_t^{HQ} \right\|^2 + \varepsilon^2} \tag{5.20}$$

where I_t^{HQ} is the ground truth frame at time t, $\tilde{I}_t^{HQ}$, represents the enhanced frame generated at time t by our model, and $\varepsilon = 10^{-3}$ is a constant value used across all experiments.

5.2 Experimental Results

5.2.1 Implementation Details

Our proposed LSFMD model mainly focuses on enhancing the video quality of screen content sequences. In our LSFMD framework, each convolutional layer, except for the final convolutional layer, is followed by a ReLU activation function [107] to introduce non-linearity into the model. Due to the limited number of available screen content se-

quences within the CTC [99], we gathered additional screen content sequences from other sources [100–102]. Our dataset consists of 41 video sequences with various resolutions, including 2560×1440, 1920×1080, and 1280×720. The lengths of these videos range from 300 to 600 frames, with frame rates varying between 20 and 60 fps. Among these sequences, 28 videos were adopted for training and the remaining 13 videos were for model testing. In the test set, 10 video sequences are provided from the CTC [99], that is a common dataset to exemplify various challenges in video quality enhancement. Notably, the CTC dataset contains only 3 videos characterized by frequent scene switches and dramatic motions. To make a robust assessment of the model's capabilities in handling real-world scenarios that feature rapid scene changes and motion complexities, we added 3 self-capture sequences to introduce more variations with scene transitions and dynamic motions. The video sequences were encoded using the HEVC reference software HM16.20-SCM8.8 under LDMS configuration as the network inputs, while the uncompressed raw video sequences were used as the ground-truths. We utilized four QPs of 22, 27, 32, and 37 for encoding the sequences and training a separate model for each QP. During training, only the luminance channel (Y channel) of each frame was considered as input. Model construction and training were implemented using PyTorch. The patch size of each input image and its corresponding ground truth was 128 × 128. To augment our dataset, we randomly selected 300 patches from one frame for each iteration. In our experiments, the learning rate was set to 0.0001 for all QPs. The Adam optimization method [108] was used to train the model for 300000 iterations. A computer equipped with Ubuntu 20.04 operating system, an Intel i9-10900K CPU, 64 GB RAM, and NVIDIA 3090Ti GPUs, was used to perform the model training.

In the LSFMD model, the number of MSRB and RB are set as 3 ($N = 3$) and the number of HFRB is also set as 3 ($M = 3$). The α in HFRB was set as 0.5 throughout the module for the channel matching between the extracted high-frequency feature and the reconstructed feature within the HFRB, apart from the first and the last HFRB. To convert a vanilla feature representation to a low-frequency and high-frequency feature representation, we set α in the first HFRB to 0. In this case, the low-frequency input of the first HFRB is disabled. To convert the low-frequency and high-frequency feature

representation back to vanilla feature representation, we set α in the last HFRB to 0, disabling the low-frequency output in the HFRB, and resulting in a single output.

5.2.2 Overall Performance

Objective Visual Quality Assessment: In this section, we compare the proposed LSFMD method with the state-of-the-art video quality enhancement methods, STDF-R3 [15], QECF [17], CAT [16], TGAF [38], STA [34], CF-STIF-M [30], and STDR [29]. To evaluate the quality enhancement performance of each quality enhancement method, the Peak Signal-to-Noise Ratio (PSNR) improvement (ΔPSNR) and the Structural Similarity Index (SSIM) improvement (ΔSSIM) are used. Table 5.1 shows the average ΔPSNR and the average ΔSSIM, respectively, over all frames of each test sequence. The best ΔPSNR/ΔSSIM is highlighted in bold. We can see that our proposed LSFMD outperforms other methods in most cases, highlighting the effectiveness of our approach. For instance, when using a QP of 37, our LSFMD achieves the highest ΔPSNR of 1.915 dB for the *paperpdf* sequence, which contains text and graphics. The average ΔPSNR of our LSFMD is 0.938 dB, which is 46.33% higher than that of CAT (0.641 dB), 52.52% higher than that of QECF (0.615 dB), 48.42% higher than that of STDF-R3 (0.632 dB), 16.09% higher than that of TGAF (0.808 dB), 8.31% higher than that of STA (0.866 dB), 17.25% higher than that of CF-STIF-M (0.800 dB), 20.72% higher than that of STDR (0.777 dB), and 21.19% higher than that of EAST-LITE (0.774 dB). For other QPs (22, 27, and 32), our LSFMD approach also outperforms other state-of-the-art video quality enhancement approaches. A similar trend can be found for ΔSSIM. This demonstrates that our LSFMD approach not only performs well in reducing pixel-level differences but also enhances the visual quality perceived by the human visual system. To further evaluate the performance, BD-rate [99] is used to indicate the bitrate savings achieved by these models under the equivalent PSNR. The experimental results are compared and tabulated in Table 5.2. Our LSFMD obtains an average BD-rate savings of 7.53%. For the test sequence *scSlideShow* with dramatic motion and scene switch, our LSFDM achieves up to 12.50% BD-rate saving for the Y component under LDMS configuration. We conjecture that our LSFMD effectively removes the artifacts and restores the high-frequency information, thereby enhancing the

Table 5.1: Overall ΔPSNR and ΔSSIM ($\times 10^{-3}$) of Different Models at QP=22,27,32,37

QP	Seq.	STDF-R3 [15]		QECF [17]		CAT [16]		TGAF [38]		STA [34]		CF-STIF-M [30]		STDR [29]		EAST-LITE [102]		Proposed LSFMD	
		ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM
37	1	0.327	3.18	0.325	3.23	0.318	4.19	0.432	4.11	0.477	4.92	0.369	3.98	0.408	4.33	0.411	**6.02**	**0.451**	4.59
	2	0.273	2.22	0.244	1.63	0.200	1.24	0.480	3.79	0.382	2.12	0.360	2.41	0.321	1.77	0.525	**5.55**	**0.529**	4.54
	3	0.867	2.78	0.770	2.7	0.951	3.27	1.101	3.63	1.134	3.62	1.243	3.41	1.266	4.01	0.991	**4.82**	**1.356**	4.25
	4	0.492	4.17	0.503	4.12	0.477	4.00	0.610	4.30	0.672	4.74	0.625	4.72	0.546	4.59	0.573	4.17	**0.679**	**5.12**
	5	1.281	2.87	1.225	2.67	1.421	3.08	1.728	3.18	1.771	3.38	1.718	3.28	1.718	3.31	1.500	3.36	**1.915**	**3.49**
	6	0.779	2.38	0.831	2.34	0.864	2.79	1.233	4.24	1.254	3.90	1.127	3.32	1.189	3.81	1.069	**4.56**	**1.299**	4.08
	7	0.301	3.46	0.365	3.51	0.329	3.05	0.516	4.13	0.538	3.85	0.278	3.85	0.306	4.16	0.528	3.07	**0.589**	**4.78**
	8	0.914	4.02	0.910	3.98	0.878	4.21	0.866	4.11	**1.166**	**4.62**	1.054	4.30	1.094	4.52	1.076	3.85	1.165	4.52
	9	0.453	5.71	0.373	3.53	0.416	6.26	0.463	6.44	0.529	6.61	0.526	5.97	0.408	5.04	0.476	4.35	**0.56**	**6.83**
	10	0.406	4.90	0.427	4.93	0.403	4.86	0.545	4.97	0.597	5.72	0.520	5.51	0.514	5.11	0.545	4.59	**0.635**	**5.77**
	11	1.008	3.28	0.907	3.38	0.969	3.56	1.137	3.78	1.292	4.02	1.286	3.93	1.046	3.72	1.107	**6.91**	**1.494**	4.23
	12	0.569	5.33	0.563	5.19	0.568	5.18	0.721	5.73	0.754	5.93	0.672	5.67	0.689	6.08	0.647	5.82	**0.799**	**6.67**
	13	0.545	5.06	0.551	4.96	0.535	4.98	0.677	5.38	0.698	5.55	0.625	5.25	0.591	5.51	0.612	3.69	**0.720**	**6.01**
	Avg.	0.632	3.80	0.615	3.55	0.641	3.90	0.808	4.45	0.866	4.54	0.800	4.28	0.777	4.30	0.774	4.67	**0.938**	**4.99**
32	Avg.	0.533	2.09	0.531	2.12	0.541	2.04	0.656	2.19	0.790	2.66	0.704	2.54	0.655	2.37	0.684	2.37	**0.798**	**2.67**
27	Avg.	0.467	0.91	0.495	1.07	0.429	0.91	0.586	0.99	0.626	1.15	0.608	1.22	0.588	1.10	0.548	1.12	**0.692**	**1.27**
22	Avg.	0.417	0.53	0.470	0.54	0.426	0.55	0.550	0.61	0.603	0.66	0.533	0.64	0.537	0.62	0.496	0.61	**0.611**	**0.68**

1: BigBuck(1920×1080, 404 frames, 60 fps) 2: ChineseEditing(1920×1080, 600 frames, 60 fps) 3: EnglishDocumentEditing(1920×1080, 300 frames, 30 fps) 4: MissionControlClip3(1920×1080, 600 frames, 60 fps) 5: Paperpdf(1920×1080, 300 frames, 60 fps) 6: Sephora(1920×1080, 300 frames, 60 fps) 7: mixvideo(1920×1080, 300 frames, 60 fps) 8: scSlideShow(1280×720, 500 frames, 20 fps) 9: scmap(1280×720, 600 frames, 60 fps) 10: scprogramming(1280×720, 600 frames, 60 fps) 11: scwebbrowsing(1280×720, 300 frames, 30 fps) 12: MissionControlClip1(2560×1440, 600 frames, 60 fps) 13: MissionControlClip2(2560×1440, 600 frames, 60 fps).

Table 5.2: Overall BD-rate(%) of Different Models at QP=22,27,32,37

Sequences	STDF-R3 [15]	QECF [17]	CAT [16]	TGAF [38]	Proposed LSFMD
BigBuck	-5.33	-6.09	-6.13	-7.24	**-7.90**
ChineseEditing	-1.39	-1.44	-1.25	-2.06	**-3.03**
EnglishDocumentEditing	-2.67	-2.59	-2.68	-3.53	**-4.33**
MissionControlClip3	-6.02	-6.24	-5.91	-7.17	**-8.21**
Paperpdf	-4.17	-4.53	-4.19	-5.81	**-7.03**
Sephora	-5.81	-6.07	-5.78	-7.25	**-10.07**
mixvideo	-2.05	-2.25	-2.00	-2.69	**-3.34**
scSlideShow	-9.94	-10.05	-9.85	-11.19	**-12.50**
scmap	-7.22	-6.42	-6.00	-7.58	**-8.56**
scprogramming	-5.81	-6.39	-6.33	-8.02	**-8.66**
scwebbrowsing	-3.14	-2.91	-2.90	**-3.67**	-3.66
MissionControlClip1	-7.27	-7.44	-7.44	-9.09	**-10.87**
MissionControlClip2	-7.25	-7.54	-7.19	-8.65	**-9.70**
Average	-5.24	-5.38	-5.20	-6.46	**-7.53**

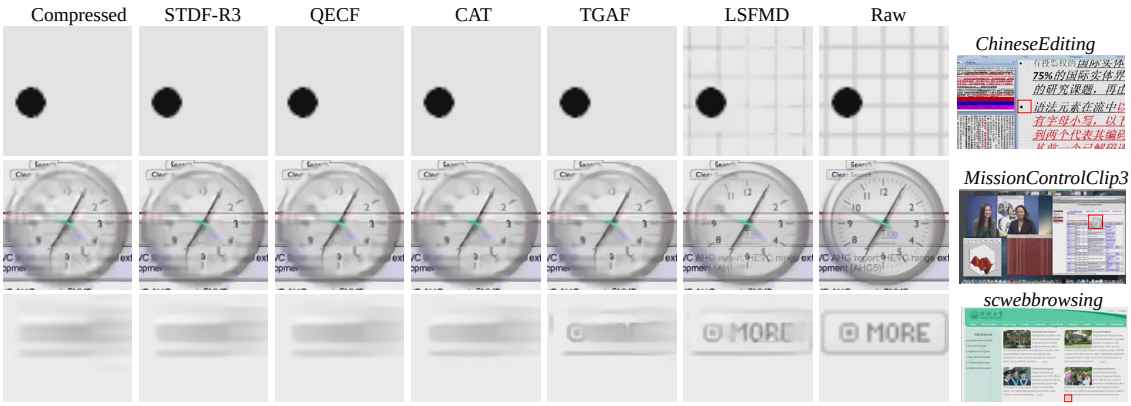


Figure 5.6: Subjective visual quality comparison at QP = 37 on *ChineseEditing*, *MissionControlClip3*, and *scwebbrowsing*.

quality of decoded frames and reducing the BD-rate.

Subjective Visual Quality Comparison: This section compares the subjective quality of different models. Fig. 5.6 shows the subjective visual quality performance of various models on the sequences *ChineseEditing*, *MissionControlClip3*, and *scwebbrowsing*, all encoded with QP = 37. From this figure, we can clearly see that the reconstructed frames of HM16.20-SCM8.8 exhibit noticeable compression artifacts and suffer from significant loss of high-frequency information details. These artifacts and details cannot be effectively restored by STDF-R3 [15], QECF [17],CAT [16], or TGAF [38]. As depicted in Fig. 5.6, our proposed LSFMD removes the artifacts and restores the content more effectively than the other models. Taking the *ChineseEditing* sequence as an example,

it can be observed that the edges of the background still disappear in other methods, but they are successfully restored by our LSFMD. For *MissionControlClip3*, the clock's numbers are blurry and the words in the background under the clock are unreadable. After being processed by our LSFMD, the details of the clock are restored clearly, and the content of the background under the clock is clear. For *scwebbrowsing*, we visualize the frame during the scene switch situation. There is a loss of high-frequency information, resulting in blurry text and icons. However, when applying our proposed approach, these elements become clearer. The examples presented in Fig. 5.6 collectively demonstrate the superiority of LSFMD over other models in terms of subjective visual quality. Once again, this showcases the ability of our LSFMD model to effectively restore high-frequency information and handle scenarios involving scene switches.

Quality Evaluation on Dramatic Motion and Scene Switches: To evaluate the capability of our proposed LSFMD in handling dramatic motion and scene switches, two different types of screen content videos were selected to compute the ΔPSNR curves for STDF-R3, QECF, CAT, TGAF, STA, CF-STIF-M, STDR, and our proposed method. The *scprogramming* sequence involves pop-up windows and window switching. These dynamic motions are commonly seen in daily life and can pose difficulties for video quality enhancement algorithms. Additionally, the *scSlideShow* sequence is composed of spliced videos from CTC [99], allowing us to evaluate the performance of our method in scenarios involving abrupt scene transitions. The results are shown in Fig. 5.7, where dashed lines indicate scene switch frames and gray shadow regions distinguish the frames exhibiting dynamic motion. The result in Fig. 5.7(a) demonstrates that our proposed LSFMD mostly outperforms the others from frame 38 to frame 63 in the *scprogramming* sequence. This shadow region encompasses window switches and a pop-up window. It can demonstrate that our proposed method can achieve significant ΔPSNR during periods of dramatic motion. While the STA only utilizes the single frame to handle the scene switch which does not perform well in dramatic motion. In Fig. 5.7(b), frame 28 and frame 44 represent the switch points between two PowerPoint slides in the *scSlideShow* sequence. Notably, our proposed method demonstrates an improvement during most of the transition points, highlighting its effectiveness in handling abrupt scene transitions. In summary, our approach

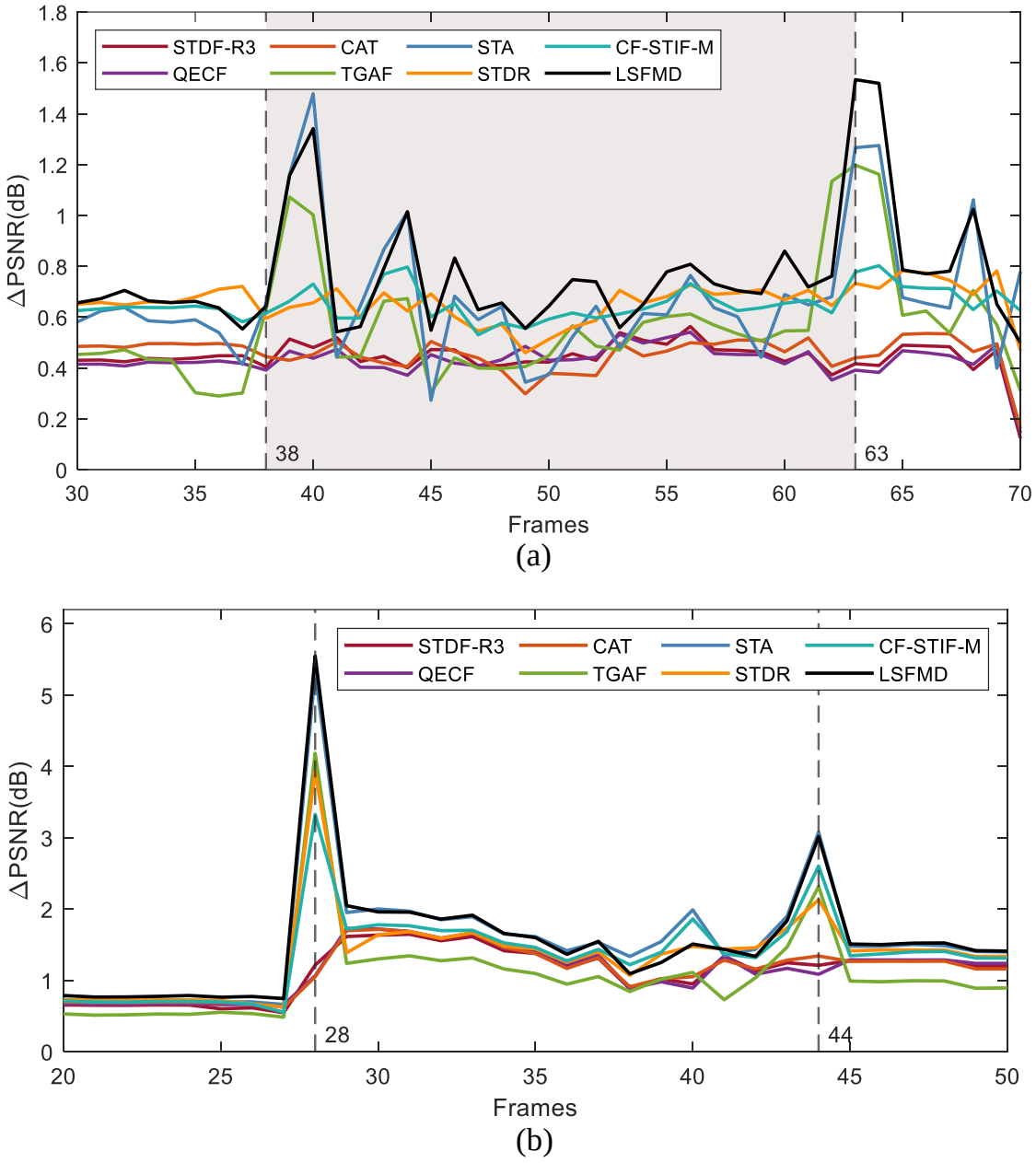


Figure 5.7: ΔPSNR curves of STDF-R3, QECF, CAT, TGAF, STA, STDR, CF-STIF-M and our LSFMD method for sequences, (a) *scprogramming* and (b) *scSlideShow*.

can take the balance between the performance in dynamic content and scene transitions but also proves effective in enhancing the quality of videos with slight motion. This robustness to screen content videos highlights the versatility and reliability of our method.

Model Size and Computational Complexity: Table 5.3 displays the average ΔPSNR in relation to the model parameters and floating point operations (FLOPs) for

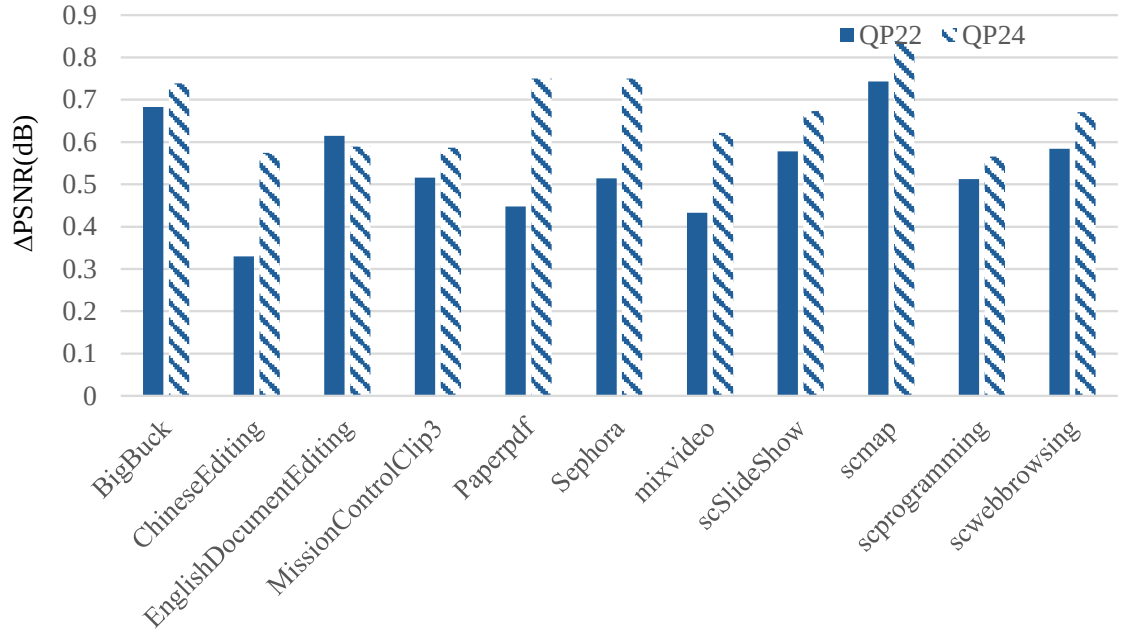
various methods including LSFMD, STDF-R3, QECF, CAT, and TGAF. These results are averaged over all test sequences. Our RB, MSRB, and MHFD modules in the LSFMD lead to increased consumption of FLOPs and require more parameters, as shown in Table 5.3. However, these modules are specifically designed to learn contextual information, capture high-frequency details, and efficiently remove redundant hierarchical features, respectively. This is further supported by the results of our ablation study, which will be discussed in the next section. As a result, the performance of LSFMD significantly surpasses other methods, as in Table 5.3. In addition, our LSFMD is a modular network, allowing for easy adjustment of the model size by varying the number of RB, MSRB, and HFRB blocks. Therefore, in applications with computational limitations, we can use a lightweight structure, such as LSFMD-N2M2, with fewer blocks ($N = 2$, $M = 2$). The LSFMD-N2M2 requires fewer model parameters than TGAF, as shown in Table 5.3, yet still achieves 0.883 dB ΔPSNR, which is 9.28% higher than that of TGAF (0.808 dB). This highlights the efficiency and effectiveness of our proposed method.

Table 5.3: Comparision of Model Size and Computational Complexity

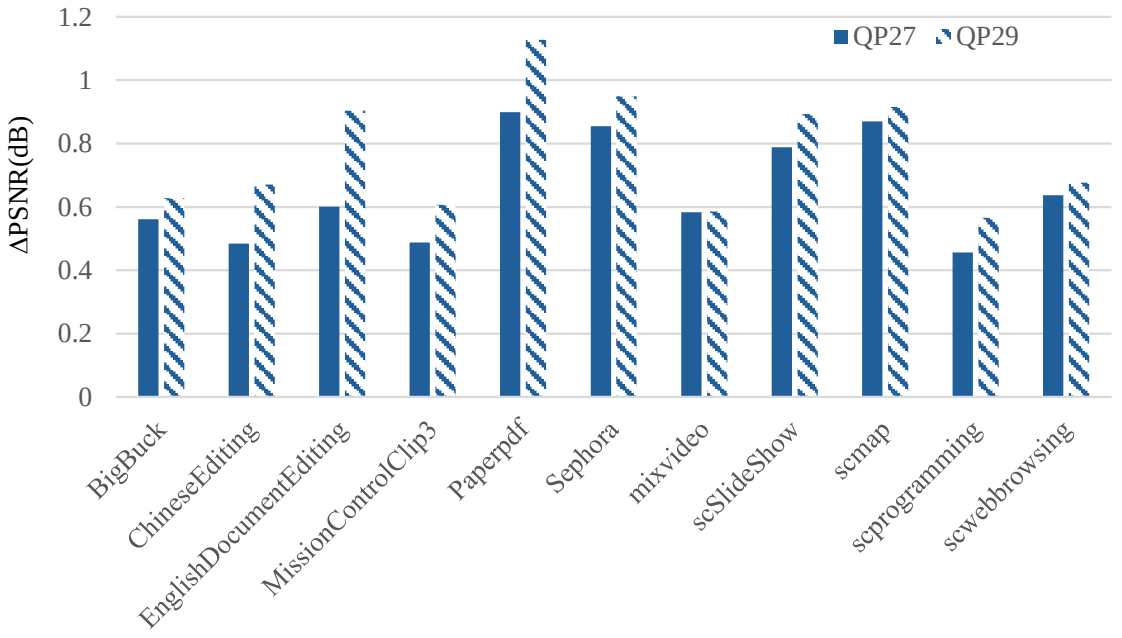
Model	STDF-R3	QECF	CAT	TGAF	LSFMD-N2M2	LSFMD
Δ PSNR (dB)	0.632	0.615	0.641	0.808	0.883	0.938
Parameters (KB)	364.51	773.313	848.546	1403.1	1244.029	1903.075
FLOPs (G)	3.856	3.292	7.541	20.344	36.383	54.449

Quality Enhancement at Different QPs: To verify the generalization ability of the LSFMD model across different QPs, we conducted additional encoding of all test sequences at QPs of 24, 29, 34, and 39, while training the model at different QPs: QP = 22, 27, 32, and 37.

The performance in terms of ΔPSNR is presented in Fig. 5.8. Fig. 5.8(a) shows the ΔPSNR of the model trained at QP = 22 and tested at QP = 22 and 24. In Fig. 5.8(b), the model is trained at QP = 27 and tested at QP = 27 and 29. Similarly, Fig. 5.8(c) and Fig. 5.8(d) show ΔPSNR of the model trained at QP =32 and 37, respectively, and tested at different QPs = 32 and 34, 37 and 39. As shown in this figure, each trained model can obtain good quality enhancement on decoded videos at adjacent QPs, thereby verifying the model's generalization ability at various QPs.

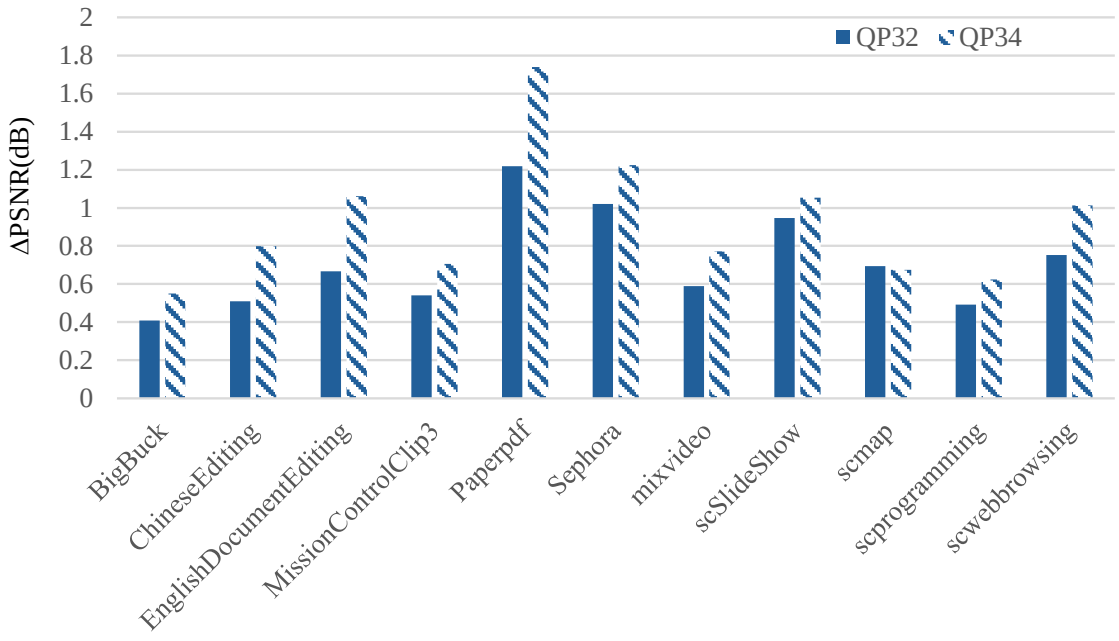


(a)

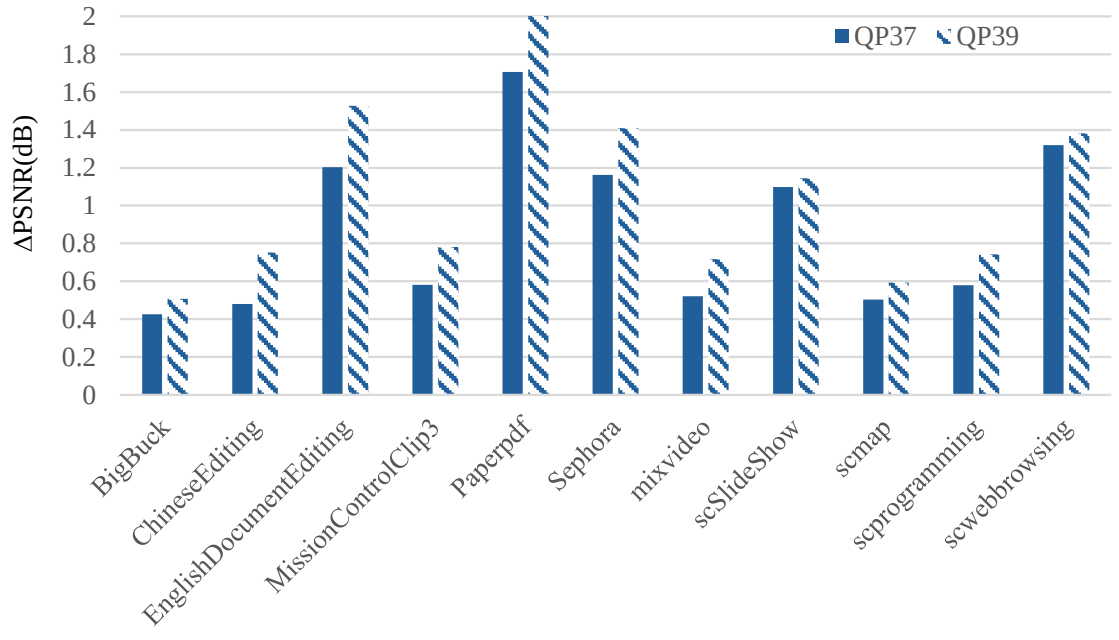


(b)

Figure 5.8: ΔPSNR of the model trained and tested at different QPs under LDMS configuration. (a) Trained at QP=22, Tested at QP=22 and 24, (b) Trained at QP=27, Tested at QP=27 and 29.



(c)



(d)

Figure 5.8: ΔPSNR of the model trained and tested at different QPs under LDMS configuration. (c) Trained at QP=32, Tested at QP=32 and 34, and (d) Trained at QP=37, Tested at QP=37 and 39.

5.2.3 Ablation Study

In this section, we conducted several ablation experiments on the LSFMD model to analyze its effectiveness in handling scene switches and reconstructing high-frequency details. To evaluate the performance, we present the ΔPSNR curve for frames affected by scene switches and visualize the frame that loses high-frequency information. Other ablation studies are evaluated by calculating the average PSNR improvement across all test sequences.

Study of Long Short-term Feature Extraction: As discussed in Section 5.1.2, the long short-term feature extraction consists of short-term feature extraction, long-term feature extraction, and MHFD. These components can adaptively handle scene switches to achieve better performance in screen content videos. To verify the effectiveness of these structures, we remove the short-term feature extraction stream, long-term feature extraction stream, or MHFD from LSFMD. The ablation results are shown in Table 5.4. We also compare the overall time consumption for enhancing a single frame at 1280×720 resolution using different structures in this table. When we remove the MHFD, as shown in Fig. 5.3(a), the "Structure A" column in Table 5.4 reveals a ΔPSNR loss of approximately 0.039 dB compared to our method. This indicates that the inclusion of MHFD improves the performance of our model. Furthermore, we also note that the inclusion of MHFD adds 38.911ms to the overall time consumption for enhancing a single frame at 1280×720 resolution, as shown in the "Time consumption" column of the table. This indicates that the local-global channel attention effectively balances time consumption and performance, further illustrating our model's efficiency. We also compare MHFD with the hierarchical feature utilization methods mentioned in Fig. 5.3(b) and presented the result in the "Structure B" column, demonstrating a ΔPSNR drop of about 0.039 dB. This suggests that distilling the useful hierarchical features makes our model pay more attention to the features of the target frame. The "Structure C" and "Structure D" demonstrate the results of using only short-term feature extraction and long-term feature extraction, respectively. We observe a significant drop in ΔPSNR, which clarifies the importance of the combination of these two streams.

To further validate that our modules meet their design objectives, we visualize the

Table 5.4: Comparisons of Different Structures in Our Proposed LSFMD at QP=37

Structure	A	B	C	D	E	**F**	G	H	I	J	K
SNFS	✓	✓	✓	✓	–	✓	✓	✓	✓	✓	✓
The number "N" of MSRBs in short term feature extraction	3	3	3	–	3	**3**	2	3	3	4	4
The number "N" of RBs in long term feature extraction	3	3	–	3	3	**3**	2	3	3	4	4
MHFD	–	–	–	–	✓	✓	✓	✓	✓	✓	✓
Hierarchical feature concatenation	–	✓	–	–	–	–	–	–	–	–	–
The number "M" of HFRBs in high-frequency reconstruction	3	3	3	3	3	**3**	2	2	4	3	4
ΔPSNR (dB)	0.899	0.899	0.839	0.727	0.930	0.938	0.883	0.903	0.897	0.945	0.933
Parameters (KB)	1783.825	1790.689	1780.177	569.521	1903.075	1903.075	1244.029	1799.347	2006.803	2458.969	2562.697
Time consumption (ms)	482.913	493.575	167.801	382.764	508.190	521.824	411.629	530.556	546.855	705.566	718.751

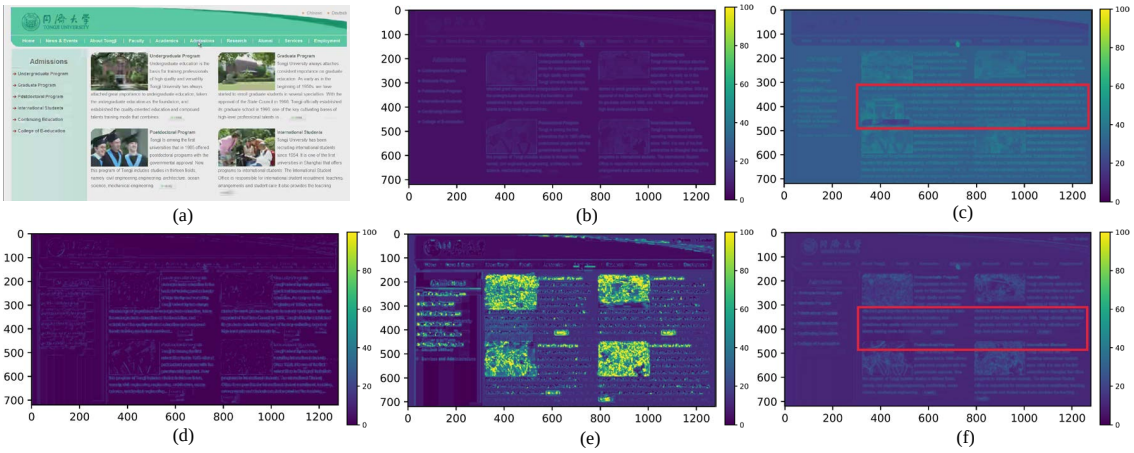


Figure 5.9: Visualization of feature maps produced by different modules of the proposed LSFMD. (a) Enhanced frame of our proposed LSFMD, (b) feature map F_{st}^0 in short-term feature extraction, (c) feature map F_{lt}^0 in long-term feature extraction, (d) feature map F_{st}^N in short-term feature extraction, (e) feature map F_{lt}^N in long-term feature extraction, and (f) feature map Q of MHFD.

extracted features from Fig. 5.9 (a) when different modules are adopted. The feature maps from the short-term feature extraction module, highlighted in Fig. 5.9 (b) and Fig. 5.9 (d), primarily focus on high-frequency information of the target frame. On the other hand, the long-term feature extraction module captures more extensive features from neighbor frames, as illustrated in Fig. 5.9 (c) and Fig. 5.9 (e). It is worth noting that in the region highlighted by the red rectangle in Fig. 5.9 (c), we can see the features from the neighbor frame are also introduced. This observation verifies that our SNFS in the short-term feature extraction stream ensures that short-term information is extracted from frames with similar content, enhancing the accuracy of the reconstruction. After the SNFS, the MSRB captures high-frequency details associated with text edges, preserving the sharpness and clarity of the text in the reconstructed frame, as we claim. Compared to the features extracted from the short-term stream, the long-term feature extraction integrates information from neighbor frames, enriching the feature set and maintaining the integrity of text and graphics across consecutive frames, as detailed in Section 5.1.3.

Therefore, our proposed MHFD effectively leverages these insights by combining the advantages of both long-term and short-term feature extractions. This successful integration is demonstrated in Fig. 5.9 (f), where the region corresponding to the red rectangle in

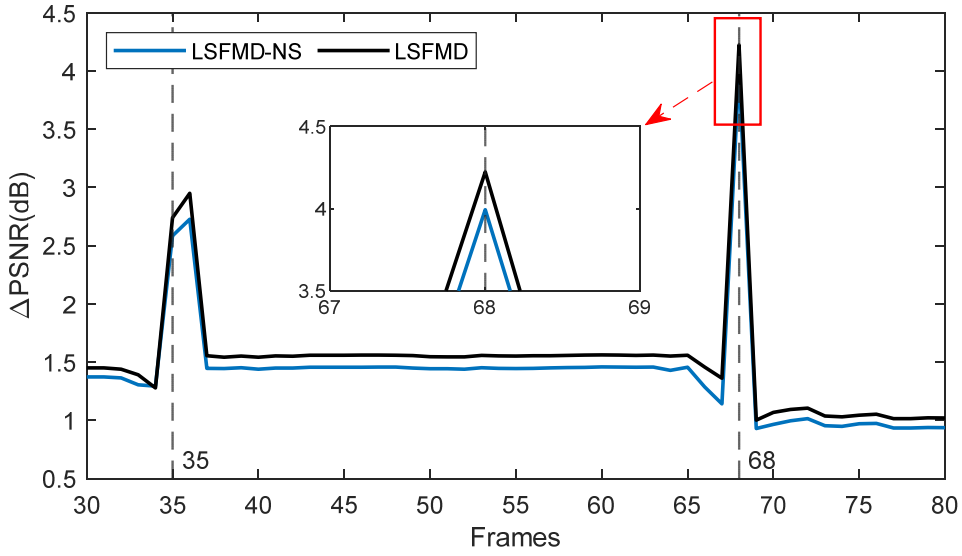


Figure 5.10: Δ PSNR curves of screen content video *scwebbrowsing*.

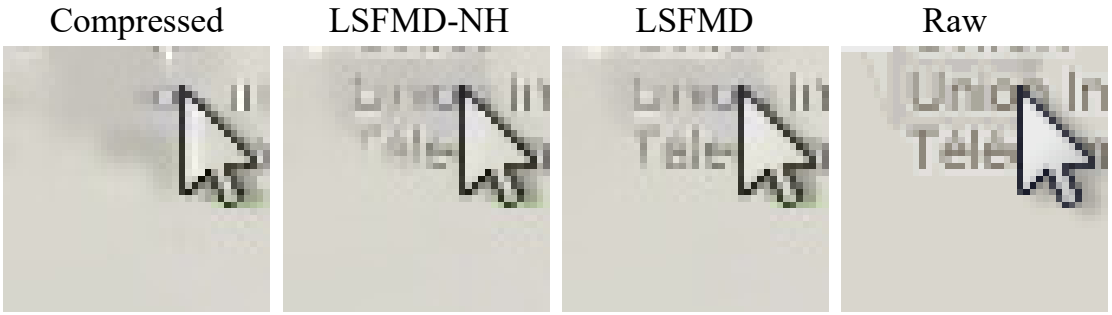


Figure 5.11: Subjective visual quality comparison at QP37 on *scmap*.

Fig. 5.9 (f) shows more relevance to the target frame than the correlated region in Fig. 5.9 (f). This observation further verifies that MHFD integrates both low and high-frequency features while filtering redundant features of neighbor frames. This balanced feature integration enhances the overall effectiveness of our approach to video quality enhancement.

In summary, the ablation study highlights the effectiveness of the long short-term feature extraction components, including short-term feature extraction, long-term feature extraction, and MHFD, in achieving better performance for screen content videos.

Study of SNFS: As discussed in Section 5.1.2, the SNFS enables the model to adaptively handle scene switches and enhances performance in screen content videos. To verify the effectiveness of the SNFS, we remove the SNFS from LSFMD. The ablation

results are shown in Table 5.4 and Fig. 5.10, where dashed lines indicate scene switch points. In the "Structure E" column of Table 5.4, it is evident that incorporating the SNFS adds an additional 13.634ms for improving each frame. However, this increase in processing time is considered acceptable given the improvement in quality it delivers. In Fig. 5.10, "LSFMD-NS" represents the LSFMD model without SNFS. The results reveal that the LSFMD-NS model experiences a slight decrease in PSNR during the scene switch compared to our proposed LSFMD model. This means that using the similarity frame to extract short-term information can further improve the quality of frames in the presence of scene switches. In other words, the SNFS component plays a crucial role in the LSFMD model's ability to adaptively handle scene switches and maintain high-quality performance. By using the similarity frame, the SNFS helps the model better extract short-term information, leading to improved reconstruction quality during scene transitions.

Study of High-frequency Reconstruction Block (HFRB): To verify the efficiency of our proposed HFRB in restoring high-frequency information in the target frame, we conducted a visual analysis of the high-frequency details in a frame from the "scmap" sequence. As illustrated in Fig. 5.11, the model labeled "LSFMD-NH" represents the LSFMD model without the incorporation of the HFRB. The absence of the HFRB in this model leads to a noticeable blurriness in the text, underscoring the importance of high-frequency detail preservation for maintaining text clarity and overall image sharpness. This comparison highlights the critical role of the HFRB in enhancing the visual quality of the reconstructed frames. The HFRB effectively restores the fine details that are often lost during the compression process, resulting in sharper and clearer images, especially in the text regions. To examine how the number of HFRB blocks affects performance, we varied the quantity of these blocks. The outcomes of these adjustments are detailed in columns "Structure F", "Structure H", and "Structure I" of Table 5.4. The results indicate that the optimal number of HFRB blocks is "3".

Influence of the Number of Blocks: The LSFMD features a modular network design that facilitates simple tuning of the model size through the adjustment of MSRB, RB, and HFRB block quantities. From columns "Structure F" to "Structure K" in Table

5.4, we observe that increasing the number of these blocks significantly enhances the PSNR gain. However, beyond a certain depth, the performance begins to decline. An excessive number of blocks not only hampers training but also leads to the loss of useful information. To strike a balance between performance and model size, we set $N = 3$ and $M = 3$ in the final LSFMD model.

This modular architecture enables fine-tuning of network complexity to achieve the desired performance-complexity trade-off. For instance, in Table 5.3, the LSFMD with fewer blocks ($N = 2$, $M = 2$) requires fewer model parameters than TGAF, but still outperforms TGAF in terms of ΔPSNR. This design flexibility ensures that the LSFMD can be optimized for different application scenarios and computational constraints, making it a versatile and adaptable solution for screen content video reconstruction.

5.3 Summary

In this chapter, we propose a novel method tailored for handling scene switches and reconstructing high-frequency information in screen content videos. Our approach includes a long short-term feature extraction module, consisting of three components: the long-term feature extraction stream, which learns contextual information; the short-term feature extraction stream, which selects relevant features from shorter inputs to better manage fast motion and scene switches; and the multi-scale feature distillation mechanism, which adaptive fuse the short-term and long-term features. Meanwhile, we introduce the SNFS into the short-term feature stream to further enhance the quality of scene switch frames. In the reconstruction phase, we propose the HFRB, which guides the model to focus on restoring high-frequency components. This is crucial for preserving the sharpness and clarity of text and other fine details in screen content videos. The novel contributions of our work, including the modular feature extraction module, the SNFS mechanism, and the HFRB, have collectively led to substantial improvements in screen content video reconstruction quality. Experimental results demonstrate that our proposed LSFMD significantly enhances the quality of compressed videos, surpassing the current state-of-the-art methods. Moreover, we conduct thorough ablation studies to verify the effectiveness of the designed network structure and its individual components.

Chapter 6

Conclusion and Future Work

6.1 Conclusions

In this thesis, we conducted an in-depth study on screen content video quality enhancement (SCVQE) using deep neural networks. In particular, we observed the characteristic of SCV in spatial and temporal domains, which is related to the artifact. Along these observations, this thesis has developed a series of methods, which are presented in Chapters 3,4,5. Specifically, the proposed EAST in Chapters 4 and the proposed LSFMD in Chapters 5 are codec-independent. The effectiveness of the proposed methods has been validated by extensive experimental results and ablation studies on the public benchmarks, including CTC [99], JCT-VC [103], and a newly created dataset, i.e., PolyUSCC [102].

Chapter 3 developed a MICNN to further improve the quality of screen content sequences at the decoder side. To identify natural content and screen content such that our MICNN can effectively enhance the reconstruction quality of different contents, it can be guided by the coding mode. Therefore, to exploit the side information from the bitstream, we proposed to use three binary mode masks devised by different coding modes, IBC, INTRA, and PLT, which are dedicated to screen content. Besides, we established a large-scale dataset containing 9810 frames for screen content videos. This dataset is publicly available to facilitate further research.

Chapter 4 designed a new approach called the EAST. In our EAST approach, the STFE is designed to capture slow motion or continuous motion, and adapt to potential scene switches and dramatic motion. It extracts features from different groups of input frames, utilizing both spatial and temporal domains. This design effectively addresses the

challenges posed by scene switches in screen content videos. After extracting the feature associated with the target frame, we also developed a novel edge-aware block that focuses on extracting high-frequency information from the target frame and then incorporates to our model. The incorporation of high-frequency information into our model is critical as it assists the network in restoring sharp edges of the compressed frame and provides valuable supplementary information during frame freezing. To adaptively enhance target frames in scenarios involving scene switches and frame freezing, we introduced a novel CSAB, which consists of a channel attention module and a spatial attention module, in the spatio-temporal feature fusion. It fuses the obtained features in a manner that optimally enhances the target frame, taking into account the specific context and requirements of each scenario.

Chapter 5 proposed LSFMD to extract and fuse the long short-term features in the corresponding frames to improve frame quality during scene switches and restore the high-frequency detail. Unlike conventional methods that use a fixed set of neighbor frames to enhance the target frame, we proposed SNFS to dynamically identify and select the most relevant frames based on content similarity. This adaptive frame selection mechanism minimizes the disturbance from unrelated frames, enhancing the accuracy of the reconstruction. To avoid the loss of features with the depth of the network, we designed the MHFD to capture the correlation of hierarchical features between short-term and long-term feature extraction streams to distillate the useful information related to the target frame, making the reconstructed frame more high-quality. Furthermore, different from the conventional reconstruction part using vanilla convolution, we adopted the HFRB to parallelly reuse the high-frequency information of the target frame to adaptively restore the high-frequency details of the reconstructed frame.

6.2 Future Work

Currently, the high computational demands of our proposed method may limit its suitability for deployment on resource-constrained devices such as smartphones and IoT devices. Addressing this limitation, we plan to explore the implementation of teacher-student techniques [116,117] to develop a more streamlined version of the model. This strategy involves

a sophisticated, computationally heavy "teacher" model imparting crucial knowledge to a lighter "student" model. The student model learns to mimic the teacher's output but with substantially reduced computational complexity. This distillation process effectively condenses the essential knowledge into a simpler form, thus decreasing the necessary computational resources without significantly compromising performance.

Further efforts to enhance computational efficiency will involve advanced model compression techniques such as pruning and quantization [118]. Pruning reduces the model size by eliminating unnecessary model parameters that contribute minimally to output performance, while quantization reduces the precision of the numerical values in the model, thereby speeding up computation and reducing memory usage. These techniques not only streamline the model but also maintain a balance between efficiency and performance.

Additionally, to bolster the model's generalizability, we plan to expand our dataset to encompass a wider variety of screen content videos. This expansion will include videos from dynamic scenarios with varied noise patterns and different types of content, ensuring that our model is robust across diverse real-world conditions.

To tailor the model for real-time applications, a key area of focus will be on minimizing frame dependencies, which can delay processing time. We plan to leverage parallel processing technologies, such as GPUs and FPGAs, which are adept at handling multiple operations simultaneously, thus speeding up the overall computational time. Furthermore, we will explore the implementation of adaptive complexity mechanisms. These mechanisms dynamically adjust the computational load of the model based on the available hardware specifications and the complexity of the content being processed, ensuring optimal performance without overburdening the device.

To further optimize the model for real-time applications on resource-constrained devices, we are considering the integration of a lookup table (LUT) [119, 120] strategy. A lookup table can significantly accelerate the processing speed by precomputing and storing the results of complex computations. Instead of recalculating these results every time they are needed, the model can simply retrieve them from the LUT, drastically reducing the computational load during runtime. This approach is particularly beneficial for functions or operations that are computationally intensive and recur frequently within our model. By deploying a LUT, we can bypass these heavy computations, which is cru-

cial for maintaining high performance without extensive computational resources. For instance, in the context of video processing, transformations such as color adjustments, gamma corrections, and certain filtering operations can be precomputed and stored in a LUT. This not only speeds up the processing but also ensures consistency in the execution of these operations. In addition to the existing strategies of model distillation and compression, the LUT method would also complement our efforts to minimize frame dependencies. By pre-storing outcomes of certain frame-related computations, the model can reduce its reliance on sequential frame processing, thereby enabling more parallel and independent frame handling. This is especially advantageous in scenarios involving rapid scene changes or high motion content, where dependency on previous frames can hinder the model's responsiveness and efficiency.

By integrating these strategies, our goal is to significantly enhance the model's applicability for real-time applications, thereby broadening its practical utility across a range of smart devices. This would make our model not only more versatile but also more accessible for various applications, from mobile video streaming to real-time video analysis in security systems, where rapid and efficient processing is crucial.

Bibliography

[1] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging hevc screen content coding extension," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 50–62, 2015.

[2] J. Chen, J. Ou, H. Zeng, and C. Cai, "A fast algorithm based on gray level co-occurrence matrix and gabor feature for hevc screen content coding," *Journal of Visual Communication and Image Representation*, vol. 78, p. 103128, 2021.

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[4] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the hevc standard," *IEEE transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1792–1801, 2012.

[5] Z. Ma, W. Wang, M. Xu, and H. Yu, "Advanced screen content coding using color table and index map," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4399–4412, 2014.

[6] X. Xu, S. Liu, T.-D. Chuang, Y.-W. Huang, S.-M. Lei, K. Rapaka, C. Pang, V. Seregin, Y.-K. Wang, and M. Karczewicz, "Intra block copy in hevc screen content coding extensions," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 409–419, 2016.

[7] W.-S. Park and M. Kim, "CNN-based in-loop filtering for coding efficiency improvement," in *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*. IEEE, 2016, pp. 1–5.

[8] Y. Dai, D. Liu, and F. Wu, "A convolutional neural network approach for post-processing in HEVC intra coding," in *MultiMedia Modeling: 23rd International Conference, MMM 2017, Reykjavik, Iceland, January 4-6, 2017, Proceedings, Part I 23*. Springer, 2017, pp. 28–39.

[9] T. Wang, M. Chen, and H. Chao, "A novel deep learning-based method of improving coding efficiency from the decoder-end for HEVC," in *2017 data compression conference (DCC)*. IEEE, 2017, pp. 410–419.

[10] S. Kuanar, C. Conly, and K. Rao, "Deep learning based HEVC in-loop filtering for decoder quality enhancement," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 164–168.

[11] R. Yang, M. Xu, and Z. Wang, "Decoder-side HEVC quality enhancement with scalable convolutional neural network," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 817–822.

[12] S. Hu, H. Wang, and S. Kwong, "Adaptive quantization-parameter clip scheme for smooth quality in h. 264/AVC," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1911–1919, 2011.

[13] R. Yang, M. Xu, Z. Wang, and T. Li, "Multi-frame quality enhancement for compressed video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6664–6673.

[14] Z. Guan, Q. Xing, M. Xu, R. Yang, T. Liu, and Z. Wang, "Mfqe 2.0: A new approach for multi-frame quality enhancement on compressed video," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 3, pp. 949–963, 2019.

[15] J. Deng, L. Wang, S. Pu, and C. Zhuo, "Spatio-temporal deformable convolution for compressed video quality enhancement," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 10 696–10 703.

[16] Y. Liu, M. Ye, Y. Gao, S. Li, Y. Zhao, and X. Li, "Content adaptive compressed screen content video quality enhancement," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 01–06.

[17] J. Huang, J. Cui, M. Ye, S. Li, and Y. Zhao, "Quality enhancement of compressed screen content video by cross-frame information fusion," *Neurocomputing*, vol. 493, pp. 486–496, 2022.

[18] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 576–584.

[19] K. Li, B. Bare, and B. Yan, "An efficient deep convolutional neural networks model for compressed image deblocking," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 1320–1325.

[20] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[21] H. Chen, X. He, L. Qing, S. Xiong, and T. Q. Nguyen, "Dpw-sdnet: Dual pixel-wavelet domain deep cnns for soft decoding of jpeg-compressed images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 711–720.

[22] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Deep generative adversarial compression artifact removal," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4826–4835.

[23] J. Guo and H. Chao, "One-to-many network for visually pleasing compression artifacts reduction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3038–3047.

[24] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," *Advances in neural information processing systems*, vol. 31, 2018.

[25] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," *arXiv preprint arXiv:1903.10082*, 2019.

[26] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4539–4547.

[27] R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, "Enhancing quality for hevc compressed videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 7, pp. 2039–2054, 2018.

[28] W. Lin, X. He, X. Han, D. Liu, J. See, J. Zou, H. Xiong, and F. Wu, "Partition-aware adaptive switching neural networks for post-processing in HEVC," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2749–2763, 2019.

[29] D. Luo, M. Ye, S. Li, C. Zhu, and X. Li, "Spatio-temporal detail information retrieval for compressed video quality enhancement," *IEEE Transactions on Multimedia*, vol. 25, pp. 6808–6820, 2022.

[30] D. Luo, M. Ye, S. Li, and X. Li, "Coarse-to-fine spatio-temporal information fusion for compressed video quality enhancement," *IEEE Signal Processing Letters*, vol. 29, pp. 543–547, 2022.

[31] Q. Zhu, J. Hao, Y. Ding, Y. Liu, Q. Mo, M. Sun, C. Zhou, and S. Zhu, "Cpga: Coding priors-guided aggregation network for compressed video quality enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2964–2974.

[32] M. Zhao, Y. Xu, and S. Zhou, "Recursive fusion and deformable spatiotemporal attention for video compression artifact reduction," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 5646–5654.

[33] Z. Wang, M. Ye, S. Li, and X. Li, "Multi-frame compressed video quality enhancement by spatio-temporal information balance," *IEEE Signal Processing Letters*, vol. 30, pp. 105–109, 2023.

[34] C. Shu, M. Ye, H. Guo, and X. Li, "Spatial-temporal adaptive compressed screen content video quality enhancement," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2024.

[35] X. Meng, X. Deng, S. Zhu, X. Zhang, and B. Zeng, "A robust quality enhancement method based on joint spatial-temporal priors for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2401–2414, 2020.

[36] W. Xiao, H. He, T. Wang, and H. Chao, "The interpretable fast multi-scale deep decoder for the standard hevc bitstreams," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1680–1691, 2020.

[37] K. Min and J. J. Corso, "Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2394–2403.

[38] Q. Zhu, Y. Qiu, Y. Liu, S. Zhu, and B. Zeng, "Compressed video quality enhancement with temporal group alignment and fusion," *IEEE Signal Processing Letters*, vol. 31, pp. 1565–1569, 2024.

[39] F. Fang, J. Li, and T. Zeng, "Soft-edge assisted network for single image super-resolution," *IEEE Transactions on Image Processing*, vol. 29, pp. 4656–4668, 2020.

[40] R. Quan, X. Yu, Y. Liang, and Y. Yang, "Removing raindrops and rain streaks in one go," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9147–9156.

[41] R. Quan, L. Zhu, Y. Wu, and Y. Yang, "Holistic lstm for pedestrian trajectory prediction," *IEEE transactions on image processing*, vol. 30, pp. 3229–3239, 2021.

[42] C. Tian, Y. Xu, W. Zuo, B. Du, C.-W. Lin, and D. Zhang, "Designing and training of a dual cnn for image denoising," *Knowledge-Based Systems*, vol. 226, p. 106949, 2021.

[43] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in neural information processing systems*, vol. 27, 2014.

[44] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[45] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4768–4777.

[46] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4305–4314.

[47] Y. Peng, Y. Zhao, and J. Zhang, "Two-stream collaborative learning with spatial-temporal attention for video classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 773–786, 2018.

[48] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2740–2755, 2018.

[49] H. Zhang, D. Liu, and Z. Xiong, "Two-stream action recognition-oriented video super-resolution," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8799–8808.

[50] L. Wang, W. Li, W. Li, and L. Van Gool, "Appearance-and-relation networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1430–1439.

[51] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "Stm: Spatiotemporal and motion encoding for action recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2000–2009.

[52] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[53] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6202–6211.

[54] X. He, Q. Hu, X. Zhang, C. Zhang, W. Lin, and X. Han, "Enhancing HEVC compressed videos with a partition-masked convolutional neural network," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 216–220.

[55] T. M. Hoang and J. Zhou, "B-drrn: A block information constrained deep recursive residual network for video compression artifact reduction."

[56] W. Sun, X. He, H. Chen, R. E. Sheriff, and S. Xiong, "A quality enhancement framework with noise distribution characteristics for high efficiency video coding," *Neurocomputing*, vol. 411, pp. 428–441, 2020.

[57] C. Ma, Y. Rao, Y. Cheng, C. Chen, J. Lu, and J. Zhou, "Structure-preserving super resolution with gradient guidance," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7769–7778.

[58] M. K. Ng, H. Shen, E. Y. Lam, and L. Zhang, "A total variation regularization based super-resolution reconstruction algorithm for digital video," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–16, 2007.

[59] Y. Wang, W. Yin, and Y. Zhang, "A fast algorithm for image deblurring with total variation regularization," *Rice University CAAM Technical Report TR07-10*, pp. 1–19, 2007.

[60] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[61] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[62] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[63] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[64] Y. Yuan and J. Wang, "Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.

[65] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146–3154.

[66] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua, "Neural aggregation network for video face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4362–4371.

[67] Q. Wang, T. Wu, H. Zheng, and G. Guo, "Hierarchical pyramid diverse attention networks for face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8326–8335.

[68] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2285–2294.

[69] B. Chen, W. Deng, and J. Hu, "Mixed high-order attention network for person re-identification," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 371–381.

[70] W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1347–1360, 2017.

[71] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1487–1500, 2017.

[72] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3047–3055.

[73] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[74] Q. Guan, Y. Huang, Z. Zhong, Z. Zheng, L. Zheng, and Y. Yang, "Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification," *arXiv preprint arXiv:1801.09927*, 2018.

[75] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *International conference on machine learning*. PMLR, 2015, pp. 1462–1471.

[76] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.

[77] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1831–1840.

[78] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 065–11 074.

[79] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 286–301.

[80] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," *Advances in neural information processing systems*, vol. 27, 2014.

[81] A. Show, "Tell: Neural image caption generation with visual attention kelvin xu," *Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio arXiv (2015-02-10) https://arxiv.org/abs/1502.03044 v3*, 2015.

[82] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[83] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9308–9316.

[84] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 534–11 542.

[85] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[86] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational lin-*

guistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.

[87] T. Liu, R. Zhao, J. Xiao, and K.-M. Lam, "Progressive motion representation distillation with two-branch networks for egocentric activity recognition," *IEEE Signal Processing Letters*, vol. 27, pp. 1320–1324, 2020.

[88] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[89] D. Huang, P. Chen, R. Zeng, Q. Du, M. Tan, and C. Gan, "Location-aware graph convolutional networks for video question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 021–11 028.

[90] T. Miyanishi, T. Maekawa, and M. Kawanabe, "Two-stream spatiotemporal compositional attention network for videoqa." in *BMVC*, 2020.

[91] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles, "Action genome: Actions as compositions of spatio-temporal scene graphs," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 236–10 247.

[92] M. Lu, D. Liao, and Z.-N. Li, "Learning spatiotemporal attention for egocentric action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[93] H. Li, Y. Cai, and W.-S. Zheng, "Deep dual relation modeling for egocentric interaction recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7932–7941.

[94] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng, "Eleatt-rnn: Adding attentiveness to neurons in recurrent neural networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 1061–1073, 2019.

[95] S. Sudhakaran, S. Escalera, and O. Lanz, "Lsta: Long short-term attention for egocentric action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9954–9963.

[96] X. Wang, Y. Wu, L. Zhu, and Y. Yang, "Symbiotic attention with privileged information for egocentric action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 249–12 256.

[97] S. Sudhakaran and O. Lanz, "Attention is all we need: Nailing down object-centric attention for egocentric activity recognition," *arXiv preprint arXiv:1807.11794*, 2018.

[98] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[99] H. Yu, R. Cohen, K. Rapaka, and J. Xu, "Common test conditions for screen content coding, document jctvc-x1015," *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG*, vol. 16, 2015.

[100] JCT-VC, "Screen content sequences," 2015. [Online]. Available: ftp://mpeg.tnt.uni-hannover.de/testsequences/

[101] S.-H. Tsang, Y.-L. Chan, and W. Kuang, "Mode skipping for HEVC screen content coding via random forest," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2433–2446, 2019.

[102] Z. Huang, "Polyuscc," 2023. [Online]. Available: https://github.com/HUANGZiyin1/PolyUSCCv2

[103] JCT-VC, "Screen content sequences provided by JCT-VC," 2022. [Online]. Available: ftp://mpeg.tnt.uni-hannover.de/testsequences/

[104] K. R. Castleman, *Digital image processing*. Prentice Hall Press, 1996.

[105] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[106] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2599–2613, 2018.

[107] K. Fukushima, "Visual feature extraction by a multilayered network of analog threshold elements," *IEEE Transactions on Systems Science and Cybernetics*, vol. 5, no. 4, pp. 322–333, 1969.

[108] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[109] Z. Ni, H. Zeng, L. Ma, J. Hou, J. Chen, and K.-K. Ma, "A gabor feature-based quality assessment model for the screen content images," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4516–4528, 2018.

[110] T. Lindeberg, *Scale-space theory in computer vision*. Springer Science & Business Media, 2013, vol. 256.

[111] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng, "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3435–3444.

[112] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 517–532.

[113] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," *Noise reduction in speech processing*, pp. 1–4, 2009.

[114] Y. Liu, J. Wu, L. Li, W. Dong, J. Zhang, and G. Shi, "Spatiotemporal representation learning for blind video quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3500–3513, 2022.

[115] P. Yi, Z. Wang, K. Jiang, Z. Shao, and J. Ma, "Multi-temporal ultra dense memory network for video super-resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2503–2516, 2020.

[116] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3048–3068, 2021.

[117] J. Zhou, B. Zhang, D. Zhang, G. Vivone, and Q. Jiang, "Dtkd-net: Dual-teacher knowledge distillation lightweight network for water-related optics image enhancement," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–13, 2024.

[118] A. Kuzmin, M. Nagel, M. Van Baalen, A. Behboodi, and T. Blankevoort, "Pruning vs quantization: Which is better?" *Advances in neural information processing systems*, vol. 36, pp. 62 414–62 427, 2023.

[119] G. Yin, Z. Qu, X. Jiang, S. Jiang, Z. Han, N. Zheng, H. Yang, X. Liu, Y. Yang, D. Li *et al.*, "Online streaming video super-resolution with convolutional look-up table," *IEEE Transactions on Image Processing*, vol. 33, pp. 2305–2317, 2024.

[120] G. He, G. Quan, C. Wu, S. Wang, D. Zhou, and Y. Li, "Multi-frame deformable look-up table for compressed video quality enhancement," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 3, 2025, pp. 3392–3400.