IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 January 2023, accepted 26 January 2023, date of publication 6 February 2023, date of current version 14 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3242673

RESEARCH ARTICLE

Mode Information Guided CNN for Quality Enhancement of Screen Content Coding

ZIYIN HUANG[1], YUI-LAM CHAN[1], (Member, IEEE), SIK-HO TSANG[2], AND KIN-MAN LAM[1], (Senior Member, IEEE)

[1]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, SAR
[2]Centre for Advances in Reliability and Safety Ltd. (CAiRS), Hong Kong Science Park, New Territories, Hong Kong, SAR

Corresponding author: Yui-Lam Chan (enylchan@polyu.edu.hk)

This work was supported in part by the Hong Kong Research Grants Council (RGC) under Grant PolyU 152069/18E, and in part by RGC Research Impact Fund (RIF) under Grant R5001-18.

ABSTRACT Video quality enhancement methods are of great significance in reducing the artifacts of decoded videos in the High Efficiency Video Coding (HEVC). However, existing methods mainly focus on improving the quality of natural sequences, not for screen content sequences that have drawn more attention than ever due to the demands of remote desktops and online meetings. Different from the natural sequences encoded by HEVC, the screen content sequences are encoded by Screen Content Coding (SCC), an extension tool of HEVC. Therefore, we propose a Mode Information guided CNN (MICNN) to further improve the quality of screen content sequences at the decoder side. To exploit the characteristics of the screen content sequences, we extract the mode information from the bitstream as the input of MICNN. Furthermore, due to the limited number of screen content sequences, we establish a large-scale dataset to train and validate our MICNN. Experimental results show that our proposed MICNN can achieve 3.41% BD-rate saving on average. In addition, our MICNN method consumes acceptable computational time compared with the other video quality enhancement methods.

INDEX TERMS Convolutional neural network, deep learning, HEVC, quality enhancement, SCC.

I. INTRODUCTION

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, etc., greatly increase the scope of video applications. Especially due to the recent spread of COVID-19, the demand for online education and virtual conferences is rapidly increasing, with Screen Content Coding (SCC) [1] playing a critical role. Unlike the natural video sequence, as shown in the example of Fig. 1(a), captured by a camera, the screen content sequence as in Fig. 1(b) can be generated from different mobile terminals directly. It is composed of many static or moving computer-generated images and texts. It often contains many uniform and flat areas, repeated patterns and limited pixel colors, etc. By making use of these screen content characteristics, SCC [1] is proposed as an extension of High Efficiency Video Coding (HEVC) [2] to increase the coding efficiency. In addition to the conventional HEVC intra (INTRA) mode [3], the SCC standard adopts two dedicated coding modes, Intra Block Copy (IBC) and palette (PLT) [4], [5], [6]. IBC [4] uses the reconstructed block of the current frame as the prediction block. IBC performs motion compensation for the current Coding Unit (CU) in the reconstructed region of the current frame, which can improve the compression efficiency of screen content video by more than 30% in [5]. 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 [5].

Although the coding efficiency can be improved by introducing the coding tools, screen content videos still contain compression artifacts corresponding to the dedicated tools in the SCC standard.

The associate editor coordinating the review of this manuscript and approving it for publication was Zahid Akhtar.

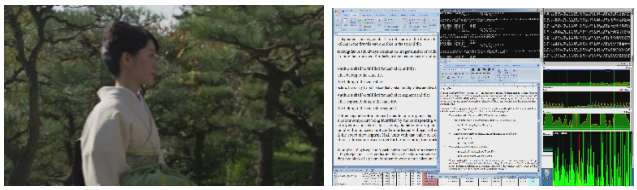


FIGURE 1. (a) Original natural frame, (b) original screen content frame.

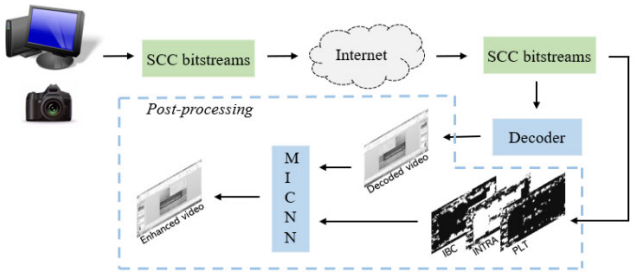


FIGURE 2. Overall framework of our proposed method.

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 new progress in this field and has achieved impressive performance in video enhancement. A series of neural network architectures were proposed to remove the artifacts in reconstructed videos. Examples include an In-loop Filter using the Convolutional Neural Networks (IFCNN) [7], a Variable-filter-size Residue-learning CNN (VRCNN) [8], a Deep CNN based Auto Decoder (DCAD) [9], a Multi-layered Deep CNN (MDCNN) [10], and a Decoder-side Scalable CNN (DS-CNN) [11]. Unlike other architectures that replace the in-loop filter, DCAD and DS-CNN were designed to improve the video quality at the decoder side. The advantage of these post-processing methods is that there is no need to modify the HEVC codec inside. Hence, the structure proposed in this paper focuses on video post-processing at the decoded side, as depicted in Fig.2.

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 [12] 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 [12] 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. With the use of the partition mask, the blocking effect is eliminated. However, this approach is 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.

In summary, the novelty and contributions of our work are twofold:

- We propose 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.
- We establish a large-scale dataset containing 9810 frames for screen content videos. This dataset will be publicly available to facilitate further research.

The remainder of the paper is organized as follows. The related works are provided in Section II. In Section III, we describe the generation of the proposed mode mask and the details of our proposed network architecture. Experiments and ablation studies are brought in Section IV. Section V concludes this paper.

II. RELATED WORKS

A. DEEP LEARNING-BASED VIDEO QUALITY ENHANCEMENT

In recent years, deep learning has been successfully applied to computer vision tasks. Many works have been applied to improve the visual quality of HEVC videos. They are divided into two major approaches. One is to modify the internal module within the codec, such as in-loop filtering for visual quality enhancement [8]. Another approach uses post-processing techniques to improve the video quality after the decoder [9], [11]. For the former one, the HEVC standard specifies an in-loop filter [2], which comprises a deblocking filter and a sample adaptive offset (SAO). The in-loop filter is embedded in the encoding and decoding loops, after inverse quantization and before saving the decoded picture in the decoded picture buffer to improve image quality. The work in [7] suggested a new in-loop filtering technology in HEVC using convolutional neural networks (CNN), namely IFCNN, to replace the SAO filter for coding efficiency and subjective visual quality improvement. Inspired by Deep Convolution Networks for Compression Artifacts Reduction (ARCNN), Dai et al. [8] proposed a Variable-filter-size Residue-learning CNN (VRCNN), which can improve the visual quality of HEVC videos without increasing the bit rate compared to the original in-loop filter in HEVC. However, the above networks cannot be directly applied to compressed videos, as they were designed as a part of the HEVC encoder. For the latter approach, a Deep CNN-based Auto Decoder (DCAD) [9] was developed to improve the visual quality through deep learning only at the decoder side. Later, a Decoder-side Scalable CNN (DS-CNN) was proposed by Yang et al. [11] wherein there are two subnetworks, DSCNN-I and DSCNN-B, to reduce the artifacts of intra-coded and inter-coded frames, respectively. In [13], QE-CNN-I and QE-CNN-P were also proposed to enhance the intra-coded frames and inter-coded frames, respectively. In [14], Huang proposed a cross feature fusion framework to enhance the gaming video in the decoder side. Notably, these works only focus on the decoded frame as the input of the network. They do not consider the information extracted from the bitstream.

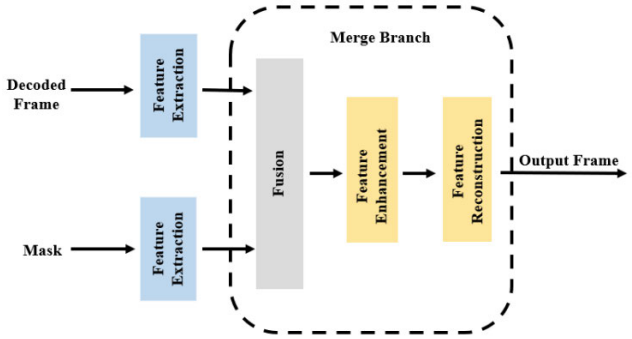


FIGURE 3. Dual-input network model structure.

FIGURE 4. Examples of (a) boundary mask and (b) mean mask.

B. DUAL-INPUT CNN ON VIDEO QUALITY ENHANCEMENT

Recently, a dual-input network has been proposed for visual quality enhancement on natural videos. The beginning of the dual-input network in Fig. 3 is composed of two branches –the main branch and the mask branch. The mask branch utilizes the compressed information extracted from the bitstream as side information for the neural network. The main branch and the mask branch in Fig. 3 are fused at certain position within the network. The post-processed frame with reduced artifacts is finally obtained. For instance, a partition-aware convolution neural network was proposed in [15], which uses the partition information produced by the encoder to assist post-processing at the decoder side. In particular, it adopts the boundary mask and the mean mask to guide the neural network. In Fig. 4(a), the boundary mask represents CU partition information by setting the CU boundary region as 255 and the non-boundary CU region as 0. The mean mask, as shown in Fig. 4(b), represents CU partition information by filling each CU with the mean value of all pixel values within each CU. Either one of these two masks can be input into the model in Fig. 3 as a grayscale image. Inspired by He et al. [12], another dual-input model proposed by Hoang and Zhou [16], a Deep Recursive Residual Network with Block information (B-DRRN), also employs the mean mask as side information. However, these dual input networks only focus on natural videos and do not consider specific features of screen content. In contrast, this paper proposes 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.

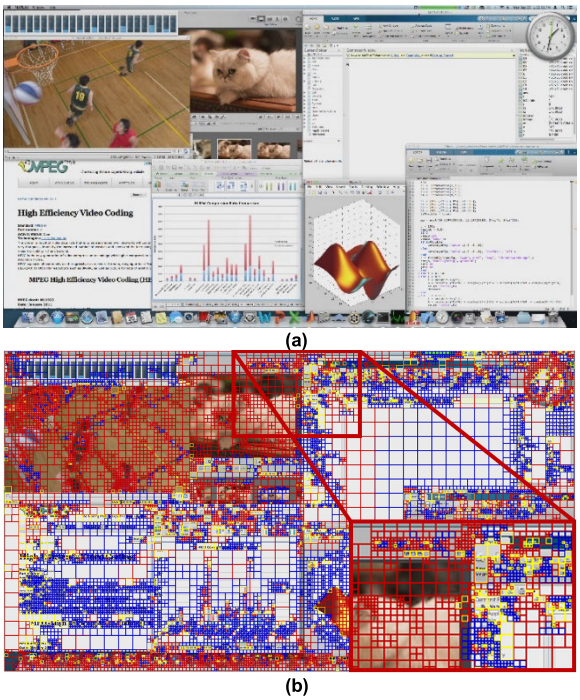


FIGURE 5. (a) Original frame, and (b) associated coding modes (red: INTRA, yellow: PLT, blue: IBC).

III. PROPOSED MULTI-INPUT CNN FOR VIDEO ENHANCEMENT

In this section, we describe our network architecture in detail. The framework of the proposed MICNN is shown in Fig. 2. To exploit the side information from the bitstream, we propose three binary masks dedicated to screen content videos. This is the first work to enhance the SC quality using deep learning with the help of mode information as the binary mask input into the deep network.

A. MOTIVATION

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. Recently, the works in [12] and [15] have proven that using the mean mask or the boundary mask can achieve better performance in the post-processing method.

However, 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. 5 explains the relationship between content type and coding mode. Fig. 5(a) shows a frame with mixed content, and

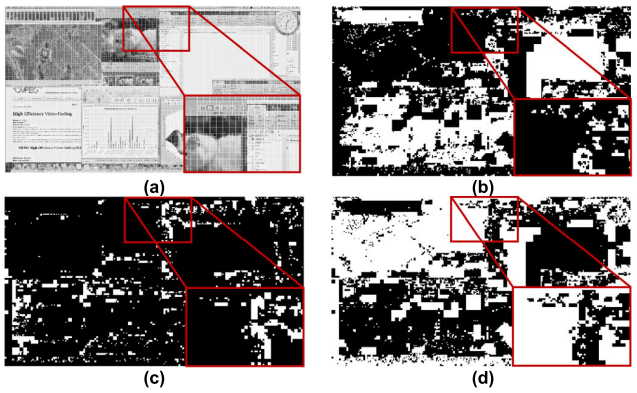


FIGURE 6. Examples of three binary mode masks. (a) Original frame with CU partition, (b) IBC binary mask, (c) PLT binary mask, and (d) INTRA binary mask.

Fig. 5(b) illustrates the associated coding modes, highlighted by different colors. As shown in Fig. 5(b), red, yellow, and blue boxes are used to represent INTRA, PLT, and IBC, respectively. 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.

B. BINARY MODE MASKS

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:

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

$$M_{PLT}\ (x, y) = \begin{cases} 1, \ if\ Mode(CU(x, y)) \epsilon PLT \\ 0, \ if\ Mode(CU(x, y)) \notin PLT \end{cases} \quad (2)$$

$$M_{INTRA}\ (x, y) = \begin{cases} 1, \ if\ Mode(CU(x, y)) \epsilon INTRA \\ 0, \ if\ Mode(CU(x, y)) \notin INTRA \end{cases} \quad (3)$$

Figure 6 shows the examples of the IBC binary mode mask, PLT binary mode mask, and INTRA binary mode mask based on the assigned values using (1)-(3).

C. PROPOSED MODE INFORMATION GUIDED CNN (MICNN)

The baseline CNN architecture is shown in Fig.7(a), where our proposed mode information guided CNN (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 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. 7(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. 7(c) are stacked as the main branch of the proposed MICNN. As shown in Fig. 7(c), the RDB contains three groups of convolutional layers that are in dense connection [17]. 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 back-propagation. Compared with the original residual block as shown in Figure 7(d), RDB uses dense connection which can exploit hierarchical features.

To formulate the MICNN model proposed in Fig. 7(b), it is assumed that the decoded and enhanced frames are represented by $\tilde{D}$ and $\tilde{Y}$, respectively. 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

$$\tilde{y} = H_{RDB}(H_{cr}(H_{RDB}(H_{cr}(H_{RDB}(H_{cr}(\tilde{D})))))) \quad (4)$$

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

$$\tilde{m}_{ibc} = H_{cr}\ (M_{IBC}) \quad (5)$$

$$\tilde{m}_{intra} = H_{cr}\ (M_{INTRA}) \quad (6)$$

$$\tilde{m}_{plt} = H_{cr}\ (M_{PLT}) \quad (7)$$

where $\tilde{m}_{ibc}$, $\tilde{m}_{intra}$, and $\tilde{m}_{plt}$ are defined as the feature maps of the IBC mode mask, INTRA mode mask, and PLT mode

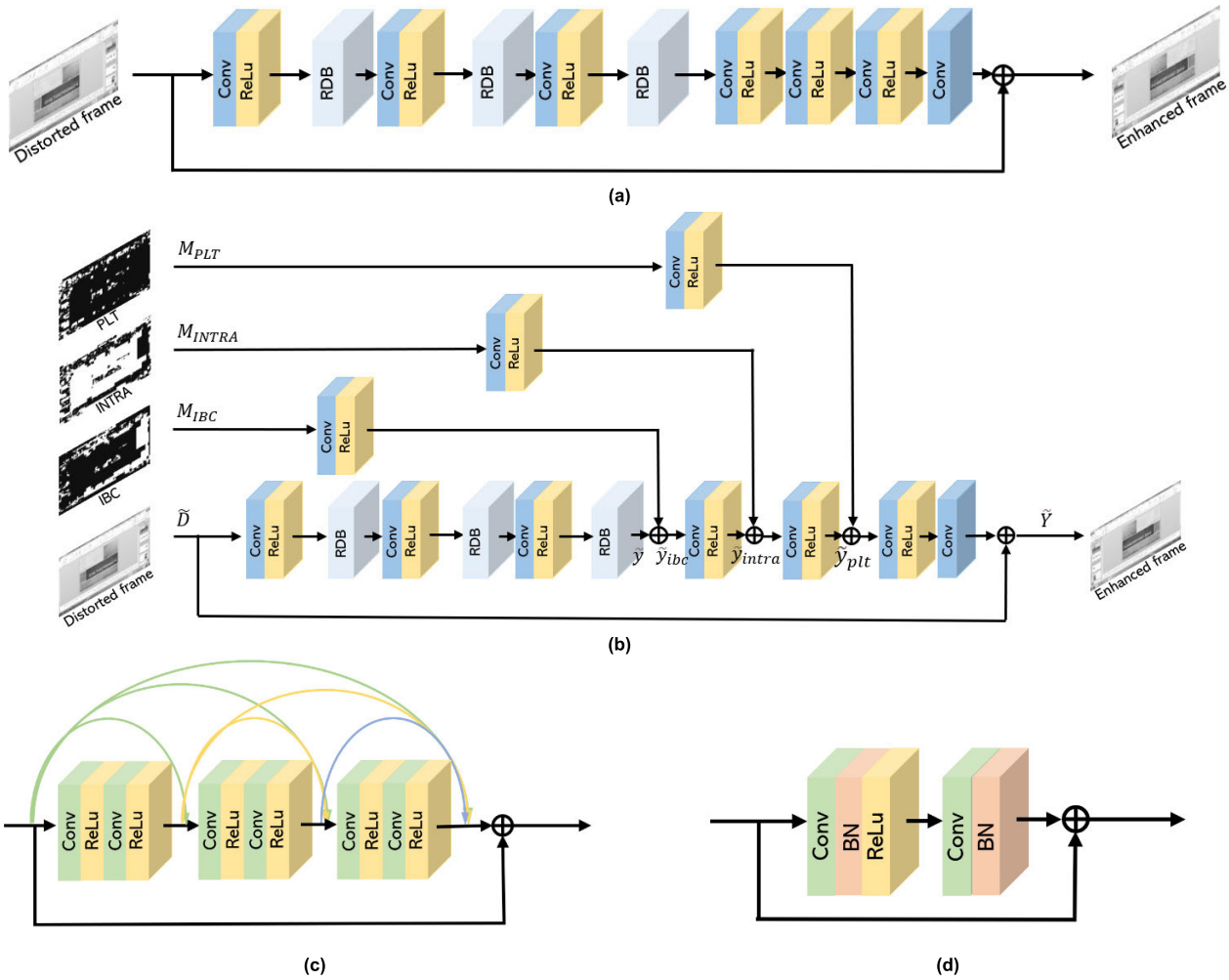


FIGURE 7. (a) The baseline CNN structure without binary mode masks, (b) the proposed MICNN structure, (c) Residual Dense Block (RDB), and (d) Traditional Residual Block.

mask, respectively. These feature maps are then integrated into the main branch in the feature fusion stage, which can be formulated as:

$$\tilde{y}_{ibc} = \tilde{y} + \tilde{m}_{ibc} \tag{8}$$

$$\tilde{y}_{intra} = H_{cr}\left(\tilde{y}_{ibc}\right) + \tilde{m}_{intra} \tag{9}$$

$$\tilde{y}_{plt} = H_{cr}\left(\tilde{y}_{intra}\right) + \tilde{m}_{plt} \tag{10}$$

where $\tilde{y}_{ibc}$, $\tilde{y}_{intra}$, and $\tilde{y}_{plt}$ 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:

$$\tilde{Y} = H_c\left(H_{cr}\left(\tilde{y}_{plt}\right)\right) + \tilde{D} \tag{11}$$

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. Given a training set $\left\{\tilde{D}_i, M_{IBC,i}, M_{INTRA,i}, M_{PLT,i}, Y_i\right\}_{i=1}^{N}$, where N is the number of patches in the training set. Here, Y_i is the ground truth patch of the decoded patch $\tilde{D}_i$ and the set $\left\{M_{IBC,i}, M_{INTRA,i}, M_{PLT,i}\right\}_{i=1}^{N}$ are the patches of mode information. The loss function can be formulated as:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left\|H\left(\tilde{D}_i, M_{IBC,i}, M_{INTRA,i}, M_{PLT,i}\right) - Y_i\right\|_2^2 \tag{12}$$

where $H(\cdot)$ denotes our proposed network and θ denotes all the parameters.

IV. PROPOSED POLYUSCC DATASET

The work of this paper 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) [19], SC sequences were gathered from other sources [18], [20], or self- capture [21] 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)

TABLE 1. Dataset.

Dataset	Ref.	Sequence	Frame	Type	Resolution
Training set	[21]	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
	[20]	MsStore	100	M	1680×1050
		NewsBrowse	100	M	1680×1050
		PaperPdf	100	TGM	1680×1050
		VisualStudio	100	M	1680×1050
	[18]	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	[18]	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
	[20]	YouTube	100	M	1680×1050
	[21]	consolenew2	300	TGM	1920×1080

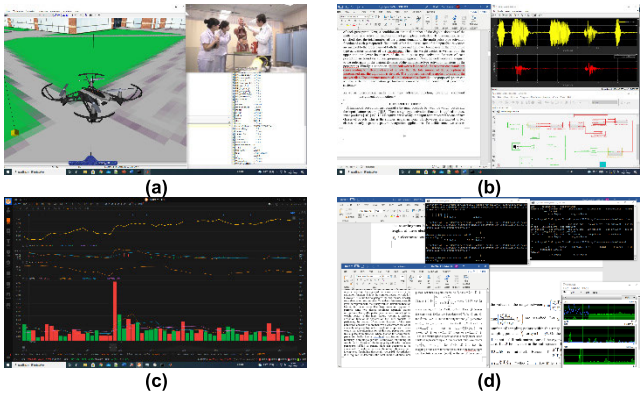


FIGURE 8. Examples of self-captured sequences. (a) *airplanevideocmd*, (b) *consoledocument*, (c) *consolenew*, and (d) *cmd*3.

are adopted, as shown in Table 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 [18] but not included in CTC [19]. Second, there are 5 SC

TABLE 2. 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

sequences from Tsang et al. [20]. To enrich the text and graphics with motion content and mixed content sequences, we further capture 14 video sequences by ourselves. Some examples of our self-captured videos are represented in Fig. 8. Our self-captured sequences will be published on website [21]. 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 1.

V. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETTING

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 [2] 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. 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 [19], none of which is the same as the training set and validation set. This is essential to avoid overfitting issue.

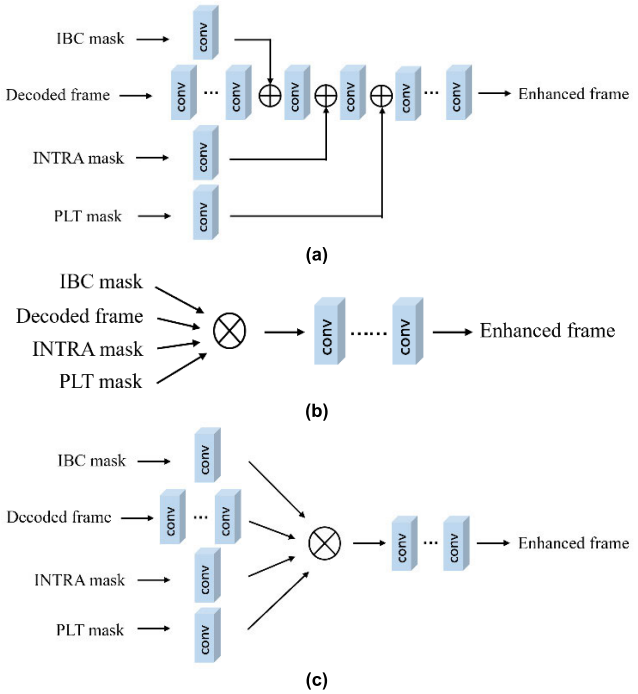


FIGURE 9. (a) Proposed fusion strategy, (b) Early Fusion by Concatenation (EFC), and (c) Late Fusion by Concatenation (LFC).

TABLE 3. Different fusion strategies 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**

B. ABLATION STUDY

As mentioned in Section III, 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 2 : *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. 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 [15] into the baseline model in Fig. 7(a) with the same number of layers and the same training process. The PSNR improvement

TABLE 4. Different masks 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 5. Different baselines at QP=37.

Seq.	1	2	3
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**

1: Mode+Residual Block 2: Mode+Dense Block 3: Proposed MICNN

of various combinations on the validation set under AI configuration is shown in Table 2. It can be seen that *ibc-intra-plt* can achieve the highest PSNR improvement (0.629 dB) over the SCC baseline at QP=37. So, we will use the order of *ibc- intra-plt* to compare other enhancement algorithms in the following discussions. To further verify the efficiency of our proposed fusion approach as in Fig. 9(a), we also evaluated two different fusion strategies - Early Fusion by Concatenation (EFC) and Late Fusion Concatenation (LFC), as shown in Fig. 9(b) and Fig.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. 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 4, 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 traditional Residual Block as shown in Figure 7(d) and the traditional Dense Block [17] instead of the RDB for compression. The results are shown in Table 5, the RDB can achieve the highest PSNR performance. Combining residual block and dense connection can help to extract the

TABLE 6. Overall ΔPSNR of different models at qp = 22, 27, 32, 37 under ai configuration.

Seq.	QECNN[13]				DCAD[9]				Partition-aware CNN[15]				QECF				Proposed baseline				Proposed MICNN			
	QP				QP				QP				QP				QP				QP			
	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37
1	0.02	0.11	0.19	0.31	0.17	0.30	0.34	0.40	-0.08	0.09	0.19	0.38	0.01	0.04	0.24	0.38	0.18	0.34	0.38	0.45	**0.24**	**0.37**	**0.43**	**0.46**
2	0.02	0.08	0.14	0.21	0.17	0.22	0.30	0.37	-0.41	-0.05	0.02	0.37	-0.01	-0.01	0.2	0.22	0.24	0.27	0.42	**0.48**	**0.32**	**0.29**	**0.42**	0.43
3	0.04	0.10	0.15	0.31	0.22	0.26	0.31	0.42	-0.09	0.07	0.19	0.44	0.01	0.03	0.24	0.37	0.25	0.32	0.38	0.48	**0.29**	**0.33**	**0.42**	**0.55**
4	0.05	0.13	0.20	0.35	0.28	0.32	0.38	0.45	-0.12	0.07	0.20	0.48	-0.01	0.04	0.27	0.43	0.32	0.39	0.44	0.49	**0.37**	**0.41**	**0.48**	**0.58**
5	0.01	-0.07	-0.33	-0.31	0.04	0.04	0.12	-0.10	-2.32	-0.69	-0.82	-0.47	-0.04	-0.02	-0.11	-0.17	0.03	-0.05	0.10	0.01	**0.08**	**0.16**	**0.20**	**0.17**
6	0.01	0.08	-0.05	0.27	0.29	0.36	0.30	0.54	-1.32	-0.32	-0.37	0.39	0.02	0	-0.05	-0.03	0.36	0.40	0.39	**0.79**	**0.50**	**0.59**	**0.54**	0.76
7	0.02	0.11	0.23	0.48	0.15	0.33	0.49	0.66	-0.27	-0.01	0.19	0.69	0.00	-0.18	-0.08	-0.43	0.20	0.34	0.51	0.81	**0.32**	**0.38**	**0.54**	**0.88**
8	0.1	0.32	0.23	0.20	0.40	0.58	0.42	0.33	-0.06	0.31	0.26	0.37	0.02	0.02	0.23	0.19	0.47	0.67	0.46	0.40	**0.62**	**0.71**	**0.59**	**0.42**
9	0.01	0.02	0.10	0.29	0.13	0.19	0.27	0.40	-0.31	-0.02	0.08	0.39	-0.01	0	0.13	0.19	0.16	0.24	0.29	0.53	**0.30**	**0.31**	**0.37**	**0.54**
10	-0.03	-0.01	0.02	0.06	0.00	0.04	0.05	0.09	-0.03	-0.01	-0.01	0.08	0.00	-0.01	**0.12**	**0.24**	0.01	0.04	0.07	0.12	**0.04**	**0.07**	0.10	0.14
11	0.05	0.22	0.33	0.55	0.19	0.37	0.57	0.71	-0.24	0.07	0.28	0.64	-0.02	0.03	0.22	0.26	0.20	0.43	0.63	0.82	**0.26**	**0.43**	**0.64**	**0.87**
12	0.11	0.06	0.25	0.94	0.22	0.22	0.43	1.03	-0.41	-0.13	0.16	1.03	-0.05	0	0.12	0.34	0.23	0.01	0.39	1.19	**0.30**	**0.27**	**0.40**	**1.20**
Avg.	0.03	0.10	0.12	0.31	0.19	0.27	0.33	0.44	-0.47	-0.05	0.03	0.40	-0.01	-0.01	0.13	0.17	0.22	0.28	0.37	0.55	**0.30**	**0.36**	**0.43**	**0.58**

1: BasketballScreen 2: ChineseEditing 3: MissionControlClip2 4: MissionControlClip3 5: scconsole 6: scdesktop 7: scflyingGraphics 8: scmap 9: scprogramming 10: scrobot 11: scSlideShow 12: scwebbrowsing

TABLE 7. Overall ΔSSIM(10^{-3}) of different models at qp = 22, 27, 32, 37 under ai configuration.

Seq.	QECNN[13]				DCAD[9]				Partition-aware CNN[15]				QECF[14]				Proposed baseline				Proposed MICNN			
	QP				QP				QP				QP				QP				QP			
	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37
1	0.00	0.17	0.88	2.17	0.16	0.70	1.29	2.81	-0.05	0.29	0.87	2.50	0.04	0.25	**1.75**	2.88	0.14	0.76	1.46	3.37	**0.26**	**0.89**	1.71	**3.47**
2	0.00	0.23	0.79	1.93	0.14	0.43	1.18	3.00	-0.09	0.15	0.65	3.39	-0.01	0.02	1.03	1.80	0.2	**0.61**	1.63	3.92	**0.25**	0.56	**1.78**	**4.11**
3	-0.03	-0.09	0.48	1.59	0.17	0.20	1.00	2.24	-0.04	0.17	0.46	2.09	0.03	0.02	1.41	2.06	0.29	**0.38**	1.22	2.85	**0.31**	0.34	**1.43**	**3.16**
4	0.04	0.13	0.60	1.75	0.3	0.52	1.28	2.61	-0.03	0.18	0.59	2.27	0.03	0.15	1.56	2.51	0.37	0.67	1.60	3.09	**0.41**	**0.70**	**1.74**	**3.56**
5	0.00	0.00	-0.02	-0.09	0.01	0.01	0.19	0.01	-0.1	-0.06	-0.30	-0.27	0	0	0.01	-0.15	0.01	0.04	0.22	0.19	**0.02**	**0.04**	**0.22**	**0.67**
6	0.02	0.10	0.25	0.56	0.09	0.22	0.45	0.82	-0.06	0.07	-0.01	0.50	0.01	0.03	0.15	-0.11	0.10	0.28	0.55	1.13	**0.12**	**0.31**	**0.64**	**1.32**
7	0.01	0.03	0.12	0.69	0.04	0.18	**0.56**	1.53	-0.07	0.01	-0.03	1.32	0	-0.22	-1.08	-7.39	0.04	0.18	0.48	2.32	**0.06**	**0.18**	0.54	**2.39**
8	0.09	1.03	1.23	0.15	0.42	1.83	2.69	3.11	0.01	1.21	1.68	4.22	0.03	0	1.13	0.99	0.51	2.18	3.07	3.93	**0.66**	**2.3**	**4.71**	**4.70**
9	0.03	-0.10	0.27	1.05	0.12	0.27	0.98	2.12	-0.10	-0.07	0.15	1.61	0.02	0.11	0.66	1.09	0.19	0.58	1.22	2.81	**0.28**	**0.63**	**1.39**	**2.99**
10	-0.29	-0.73	-1.47	-0.64	-0.07	0.12	-1.30	0.01	-0.20	-0.49	-2.50	-0.79	-0.08	-0.64	0.15	**4.99**	-0.04	0.00	-0.49	0.93	**0.14**	**0.33**	**0.62**	1.20
11	0.03	0.18	0.36	0.99	0.06	0.28	0.69	1.50	-0.11	0.07	0.35	1.18	-0.02	0.1	0.61	0.67	0.06	**0.32**	0.79	1.78	**0.09**	0.31	**0.85**	**1.90**
12	0.03	0.08	0.12	0.84	0.06	0.17	0.38	1.30	-0.05	0.00	-0.11	0.58	0.02	0.01	0.27	0.29	0.06	0.18	0.38	1.60	**0.07**	**0.21**	**0.43**	**1.80**
Avg.	-0.01	0.09	0.30	0.92	0.13	0.41	0.78	1.76	-0.07	0.13	0.15	1.55	0.01	-0.01	0.64	0.80	0.16	0.52	1.01	2.33	**0.22**	**0.57**	**1.34**	**2.61**

1: BasketballScreen 2: ChineseEditing 3: MissionControlClip2 4: MissionControlClip3 5: scconsole 6: scdesktop 7: scflyingGraphics 8: scmap 9: scprogramming 10: scrobot 11: scSlideShow 12: scwebbrowsing

TABLE 8. Overall Bd-rate(%) of different models at qp = 22, 27, 32, 37 under ai configuration.

Sequences	QECNN[13]	DCAD[9]	Partition-aware CNN[15]	QECF	Proposed baseline	Proposed 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**

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.

C. OVERALL PERFORMANCE

1) OBJECTIVE VISUAL QUALITY ASSESSMENT

In this section, we compare QECNN [13], DCAD [9], Partition-aware CNN [15], and QECF [14] with our proposed

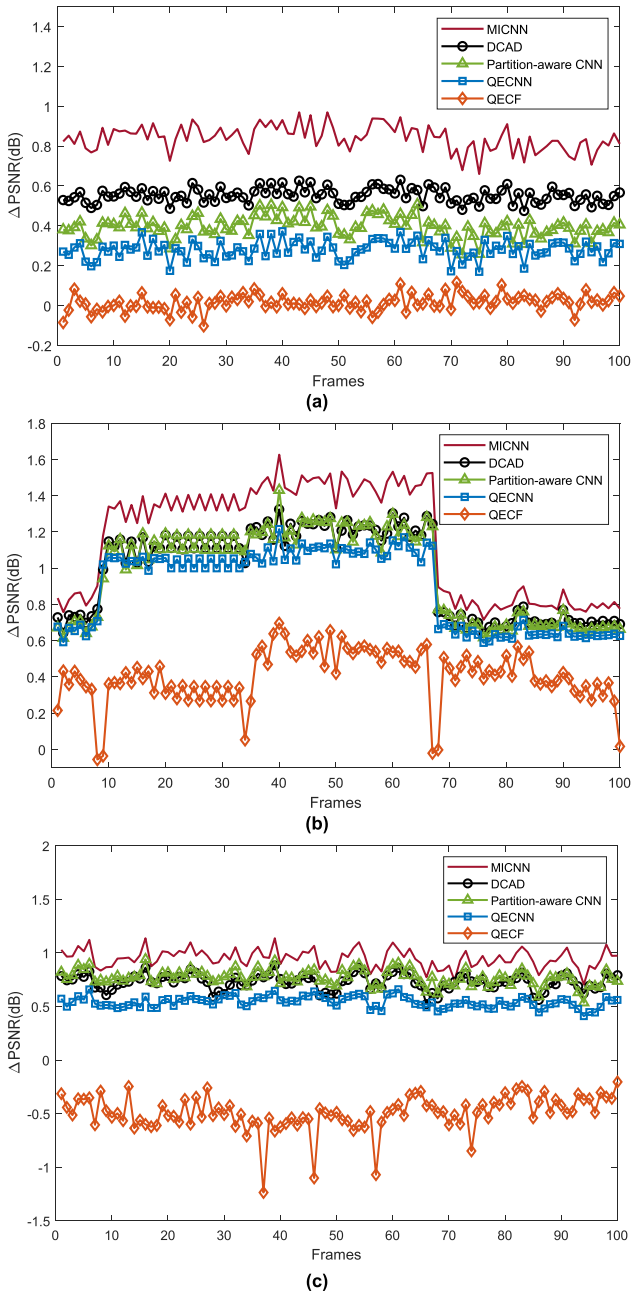


FIGURE 10. PSNR improvement curves of partition-aware CNN, DCAD, QECNN, QECF and our MICNN method for sequences, (a) *scdesktop*, (b) *scwebbrowsing*, and (c) *scflyingGraphics*.

MICNN. Table 6 and Table 7 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 PSNR/SSIM improvement is highlighted in bold and the underline number is the second-best PSNR/SSIM improvement. 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 PSNR improvement of our-MICNN approach reaches 1.20 dB, i.e., for sequence scweb-browsing. 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 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. 10. 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 [22] is used to indicate the bitrate savings of these models under the equivalent PSNR. Experimental results are compared and tabulated in Table 8. 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.

2) SUBJECTIVE VISUAL QUALITY COMPARISON

This section compares the subjective quality of different models. Fig. 11 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. 11, 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 becomes 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. 11 show that MICNN is superior to the other models in terms of subjective visual quality. There are no uneven regions at the

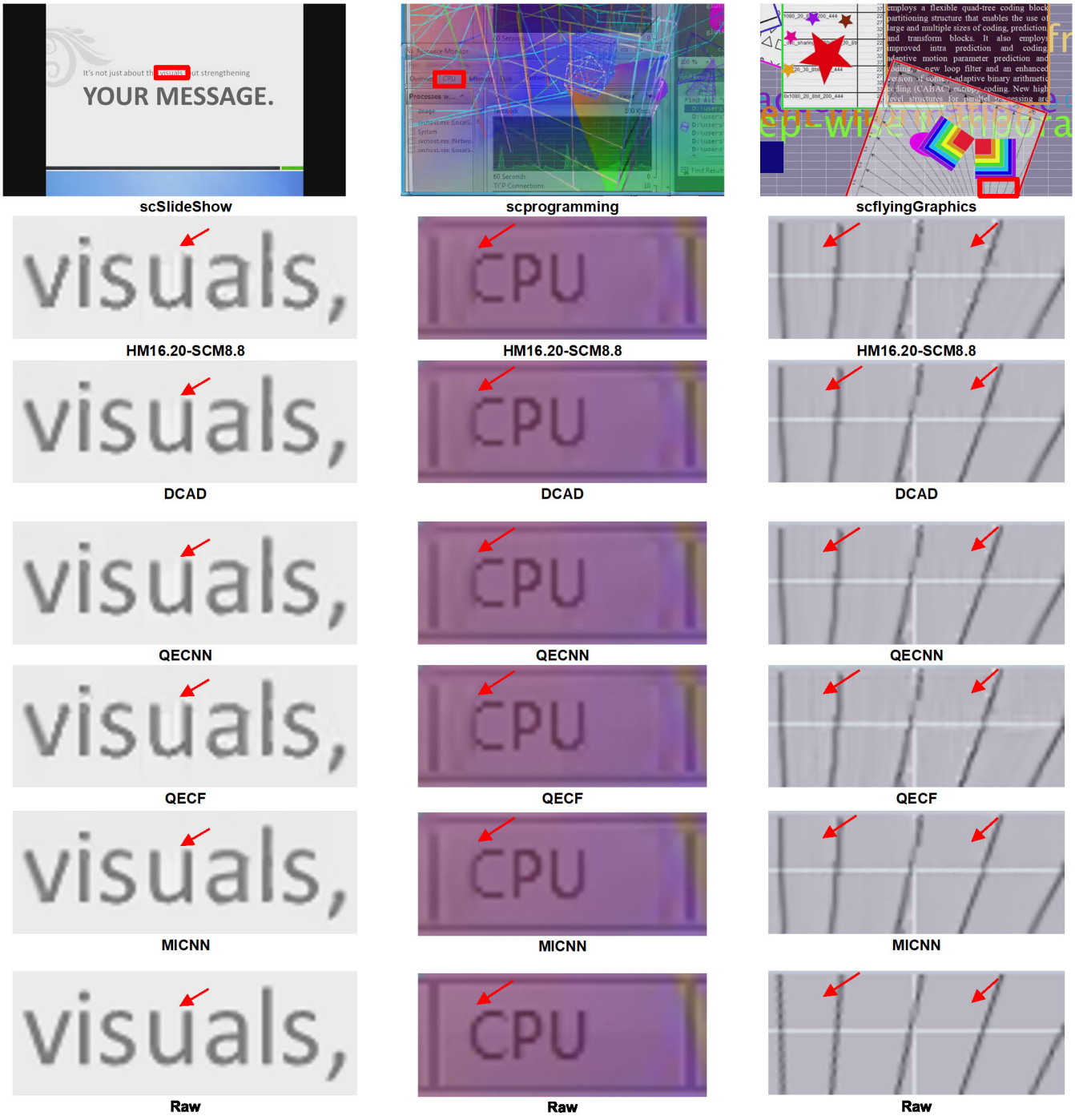


FIGURE 11. Subjective visual quality comparison at QP = 37 on (a) scSlideShow, (b) scprogramming,and (c) scflyingGraphics.

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) 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. 12. Fig. 12(a) shows the PSNR improvement of the model trained at QP = 22 and tested at QP = 22 and 24. In Fig. 12(b), the model is trained at QP = 27 and tested at QP = 27 and 29. Similarly, Fig. 12(c) and Fig. 12(d) show ΔPSNR of the model trained at QP =32 and 37 and tested at different QPs = 32 and 34, 37 and 39,

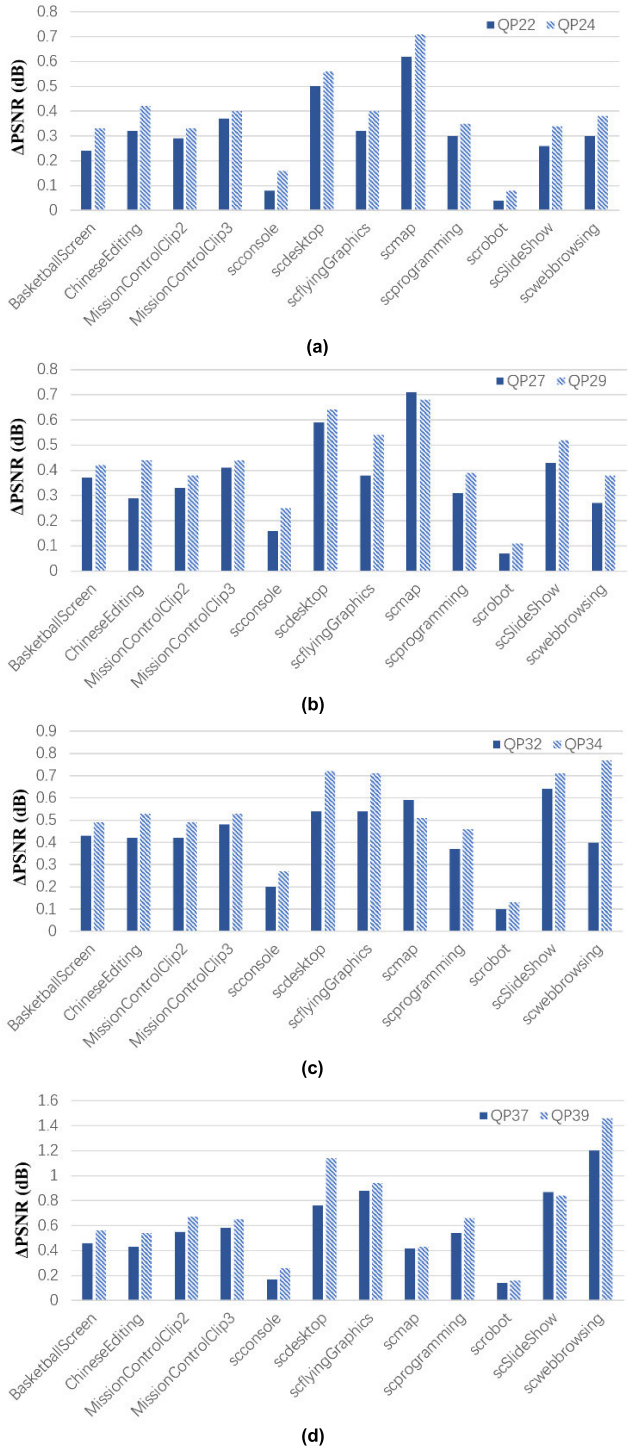


FIGURE 12. ΔPSNR of the model trained and tested at different QPs under AI configuration. (a) Trained at QP=22, Tested at QP=22 and 24, (b) Trained at QP=27, Tested at QP=27 and 29, (c) Trained at QP=32, Tested at QP=32 and 34, and (d) Trained at QP=37, Tested at QP=37 and 39.

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.

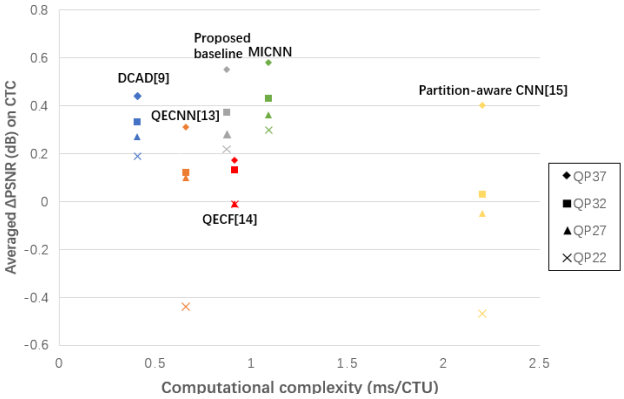


FIGURE 13. Average ΔPSNR against computational complexity of different methods in the decoder side.

TABLE 9. Comparison of running time per frame.

Frame Size	QECNN [13]	DCAD [9]	Partition-aware CNN [15]	QECF [14]	Proposed baseline	MICNN
1280x720	53.94ms	50.38ms	442.39ms	163.57ms	150.73ms	164.57ms
1920x1080	117.94ms	109.69ms	1009.81ms	378.82ms	346.17ms	423.38ms
2560x1440	203.94ms	193.49ms	1779.68ms	703.98ms	611.50ms	718.61ms

TABLE 10. Comparison of model size.

Model	QECNN [13]	DCAD [9]	Partition-aware CNN [15]	QECF [14]	Proposed baseline	MICNN
Model size (KB)	451.78	296.64	3114.31	764.067	1268.16	1269.89

4) COMPARISONS ON COMPUTATIONAL COMPLEXITY IN DECODER

To evaluate the computational complexity of various models, we follow the measurement metric of other post-processing algorithms [11], [15] 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. 13 shows the average ΔPSNR against running time per CTU for MICNN, DCAD [9], QECNN [13], QECF [14], and partition-aware CNN [15] methods. The results shown in this figure are calculated over all the test sequences on average. In Fig. 13, 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 PSNR improvement over other models. In Table 9, we also compare the overall time consumption in enhancing one frame in different resolutions of different methods. From Table 9 and Table 10, 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.

5) MODEL SIZE

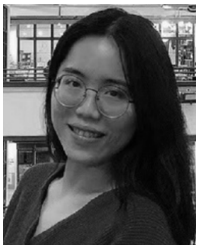
Model complexity in terms of model size for various models is also evaluated in Table 10. Model size reflects the number of network parameters. Compared to our baseline model, MICNN adds sub-branches to improve performance without significantly affecting model size. Besides, the proposed MICNN can achieve 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.

VI. CONCLUSION

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. In the future, we will move to create a real-time model, which is essential for the further development of real-time applications.

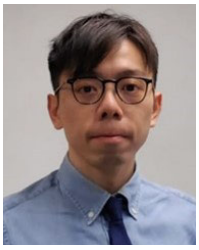
REFERENCES

[1] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging HEVC screen content coding extension," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 50–62, Jan. 2016.
[2] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Sep. 2012.
[3] J. Lainema, F. Bossen, W. Han, J. Min, and K. Ugur, "Intra Coding of the HEVC Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.
[4] X. Xu, "Intra block copy in HEVC screen content coding extensions," *IEEE J. Emerg. Sel. Topic Circuits Syst.*, vol. 6, no. 4, pp. 409–419, Dec. 2016.
[5] *IBC+Palette Realizes Screen Content Coding Optimization*. Accessed: Jun. 2022. [Online]. Available: https://cloud.tencent.com/developer/article/1453587
[6] Z. Ma, W. Wang, M. Xu, and H. Yu, "Advanced screen content coding using color table and index map," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4399–4412, Oct. 2014.
[7] W.-S. Park and M. Kim, "CNN-based in-loop filtering for coding efficiency improvement," in *Proc. IEEE 12th Image, Video, Multidimensional Signal Process. Workshop (IVMSP)*, Jul. 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 *Proc. Int. Conf. Multimedia Model. (MMM)*, 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 *Proc. Data Compress. Conf. (DCC)*, Apr. 2017, pp. 410–419.
[10] S. Kuanar, C. Conly, and K. R. Rao, "Deep learning based HEVC in-loop filtering for decoder quality enhancement," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 164–168.
[11] R. Yang, M. Xu, and Z. Wang, "Decoder-side HEVC quality enhancement with scalable convolutional neural network," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2017, pp. 817–822.
[12] 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 *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 216–220.
[13] R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, "Enhancing quality for HEVC compressed videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 2039–2054, Jul. 2019.
[14] 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, Jul. 2022.
[15] W. Lin, "Partition-aware adaptive switching neural networks for post-processing in HEVC," *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 2749–2763, Nov. 2020.
[16] T. M. Hoang and J. Zhou, "B-DRRN: A block information constrained deep recursive residual network for video compression artifacts reduction," in *Proc. Picture Coding Symp. (PCS)*, Nov. 2019, pp. 1–5.
[17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[18] *Screen Content Sequences Provided by JCT-VC*. Accessed: Apr. 2022. [Online]. Available: https://mpeg.tnt.uni-hannover.de/testsequences/
[19] *Common Test Conditions for Screen Content Coding*, document JCT-VC, JCTVC-X1015, May/Jun. 2016, pp. 1–6. Accessed: Apr. 2022.
[20] S.-H. Tsang, Y.-L. Chan, and W. Kuang, "Mode skipping for HEVC screen content coding via random forest," *IEEE Trans. Multimedia*, vol. 21, no. 10, pp. 2433–2446, Oct. 2019.
[21] *POLYUSCC Provided by Ziyin Huang*. [Online]. Available: https://github.com/HUANGZiyin1/PolyuSCC
[22] K. Sharman and K. Suehring, *Common Test Conditions*, document JCTVC-Z1100, 26th Meeting, Geneva, Switzerland, Jan. 2017.



ZIYIN HUANG received the M.Sc. degree from the Guangdong University of Technology, China, in 2020. She is currently pursuing the Ph.D. degree with the Digital Signal Processing Laboratory, The Hong Kong Polytechnic University, Hong Kong. Her current research interests include deep learning and video enhancement.

YUI-LAM CHAN (Member, IEEE) received the B.Eng. (Hons.) and Ph.D. degrees from The Hong Kong Polytechnic University, Hong Kong, in 1993 and 1997, respectively. He joined The Hong Kong Polytechnic University, in 1997, where he is currently an Associate Professor with the Department of Electronic and Information Engineering. He is actively involved in professional activities. He has authored over 140 research papers in various international journals and conferences. His research interests include multimedia technologies, signal processing, image and video compression, video streaming, video transcoding, video conferencing, digital TV/HDTV, 3DTV/3DV, multiview video coding, machine learning for video coding, and future video coding standards, including screen content coding, light-field video coding, and 360-degree omnidirectional video coding. He was the Secretary of the 2010 IEEE International Conference on Image Processing. He was also the Special and Demo Sessions Co-Chair of IEEE International Conference on Visual Communications and Image Processing and the Publication Chair of the IEEE International Conference on Multimedia and Expo. He served as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING.



SIK-HO TSANG received the Ph.D. degree from The Hong Kong Polytechnic University (PolyU), Hong Kong, in 2013. He was a Research Fellow at PolyU, where he is currently a Postdoctoral Fellow at the Centre for Advances in Reliability and Safety (CAiRS). His research interests include video compression, such as HEVC and VVC, image and video processing, and imaging sensor techniques, such as video quality assessment and blur detection, using deep learning. He is a Reviewer of international journals, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON BROADCASTING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and IEEE ACCESS.

KIN-MAN LAM (Senior Member, IEEE) received the Associate degree (Hons.) in electronic engineering from The Hong Kong Polytechnic University (formerly Hong Kong Polytechnic), in 1986, the M.Sc. degree in communication engineering from the Department of Electrical Engineering, Imperial College, U.K., in 1987, and the Ph.D. degree from the Department of Electrical Engineering, The University of Sydney, Australia, in 1996. From 1990 to 1993, he was a Lecturer at the Department of Electronic Engineering, The Hong Kong Polytechnic University. He joined the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, as an Assistant Professor, in October 1996, where he became an Associate Professor, in 1999, and has been a Professor, since 2010. He is currently an Associate Dean with the Faculty of Engineering. He was actively involved in professional activities. His current research interests include image and video processing, computer vision, and human face analysis and recognition. He has been a member of the organizing committee or program committee of many international conferences. He was the Chairman of the IEEE Hong Kong Chapter of Signal Processing, from 2006 to 2008. He was the Director-Student Services and the Director-Membership Services of the IEEE SPS, from 2012 to 2014 and from 2015 to 2017, respectively. He was also the VP-Member Relations and Development and the VP-Publications of the Asia–Pacific Signal and Information Processing Association (APSIPA), from 2014 to 2017 and from 2017 to 2021, respectively. He is currently the IEEE SPS VP-Membership and the Member-at-Large of APSIPA. He was an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, from 2009 to 2014, and *Digital Signal Processing*, from 2014 to 2018. He was also an Editor of *HKIE Transactions*, from 2013 to 2018, and an Area Editor of the *IEEE Signal Processing Magazine*, from 2015 to 2017. He also serves as a Senior Editorial Board Member for *APSIPA Transactions on Signal and Information Processing* and an Associate Editor for *EURASIP International Journal on Image and Video Processing*.