Low-Complexity Intra Prediction for Screen Content Coding by Convolutional Neural Network

***Abstract*—Screen content coding (SCC) is developed to encode screen content videos, and it is an extension of High Efficiency Video Coding (HEVC). Since screen content videos contain computer-generated content that shows special characteristics, SCC adopts the new Intra Block Copy mode and Palette mode besides the HEVC based Intra mode to improve the coding efficiency. However, the exhaustive mode searching process makes the SCC encoder computational expensive. In this paper, a low-complexity intra prediction algorithm is proposed by the convolutional neural network (CNN). The proposed network skips unnecessary coding units (CUs) and mode candidates by imitating the behavior of the original SCC encoder. The network first decides if a CU size should be checked by analyzing global features, and it decides which mode should be checked by analyzing the local features. Experimental results show that the proposed algorithm achieves 53.44% computational complexity reduction on average with 1.94% Bjøntegaard delta bitrate loss under All Intra configuration.**

Keywords—Screen Content Coding (SCC), High Efficiency Video Coding (HEVC), Convolutional Neural Network, Fast Algorithm.

I. INTRODUCTION

Screen content videos are videos captured from the display screens of electronic devices. With the fast development of the Internet and wireless communication, screen content videos have been applied to many screen-sharing-based applications, such as cloud-mobile computing, remote education, video conference with document sharing, wireless or Wi-Fi screen mirroring [1]. As shown in Fig. 1, a screen content frame usually contains both the traditional natural image blocks (NIBs) and the new computer-generated screen content blocks (SCBs). Compared with NIBs, SCBs show different characteristics such as many repeated patterns within a frame, limited colors and sharp edges. Since High Efficiency Video Coding (HEVC) only considers the characteristics of NIBs, the Joint Collaborative Team on Video Coding (JCT-VC) started a Screen Content Coding (SCC) extension [2] based on HEVC to explore new coding tools for SCBs.

As an extension of HEVC, SCC adopts the same quadtree-based coding tree unit (CTU) partitioning structure as HEVC. To predict the traditional NIBs in screen content videos, SCC directly inherits the Intra mode from HEVC. Then, to predict the new SCBs that have different characteristics, SCC adopts two new coding modes, which are Intra Block Copy (IBC) [4] and Palette (PLT) [5]. With the adoption of the new mode candidates, SCC achieves over 50% Bjøntegaard delta bitrate (BDBR) [6] reduction compared with HEVC. However, the quadtree-based CTU partitioning structure and the exhaustive mode checking strategy make the SCC encoder computational expensive.

To reduce the computational burden of the SCC encoder, various algorithms have been proposed, and they can be divided into three categories. The first category is to reduce the mode candidates to be checked [7], [8]. In [7], learning frames are first extracted and encoded by the original encoder to build Bayesian classifiers, and then the classifiers are

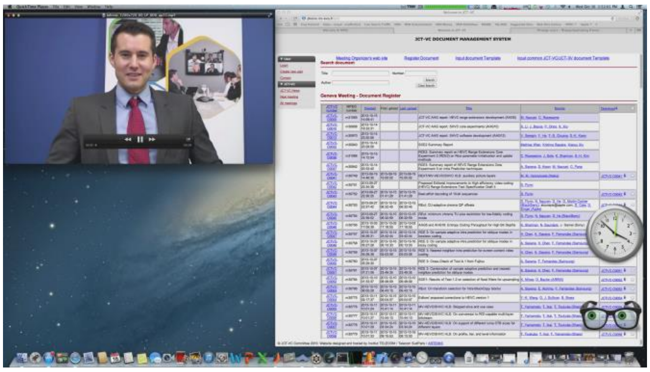

Fig. 1. A frame in a screen content sequence "MissionControlClip2".

applied to the following frames to adaptively skip unnecessary mode candidates. In [8], a fast searching approach was proposed for IBC mode by calculating a hash value, where only blocks with the same hash value as the current coding unit (CU) are searched by IBC mode.

The second category is to make a fast CU partitioning decision [9], [10]. In [9], various features describing the CU statistics and sub-CU homogeneity are extracted to train neural network-based classifiers. The simplified SCC encoder can adaptively skip unnecessary CU sizes according to the output of the classifiers. In [10], a convolutional neural network (CNN) based classifier was proposed to early terminate the CU partition by taking the raw samples as the input. Since CNN contains much more training parameters than the traditional machine-learning-based method, it significantly reduces the encoding time.

The third category is to simplify both the mode decision and the CU partitioning decision [11]–[13]. Decision tree-based classifiers were proposed in [11], [12]. First, CUs are classified into SCBs and NIBs by analyzing various features. Then, only IBC and PLT modes are checked for SCBs while only Intra mode is checked for NIBs. Besides, another set of decision trees are trained to early terminate the CU partition. In [13], CUs are also classified into NIBs and SCBs by extracting limited hand-crafted rules. Only Intra mode is checked for NIBs but all modes are checked for SCBs because of the low classification accuracy. Then, spatial and temporal correlation, as well as the coding bits of the current CU are analyzed to make early CU partitioning decisions.

Compared with the traditional fast prediction algorithms [7]–[9], [11]–[13] that rely on the limited number of hand-crafted rules or hand-crafted features, CNN based classifiers show further improvement as they contain much more trainable parameters. However, the CNN based classifiers in [10] only consider the CU partitioning decision, which leaves rooms for further improvement. In this paper, we propose a low-complexity intra prediction algorithm by CNN. First, the proposed network predicts whether a CU size should be checked or not by extracting global features from a CTU. Second, it a CU size is decided to be checked, the proposed network decides which mode should be checked by extracting

The following publication W. Kuang, Y. -L. Chan and S. -H. Tsang, "Low-Complexity Intra Prediction for Screen Content Coding by Convolutional Neural Network," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), 2020, pp. 1-5 is available at https://dx.doi.org/10.1109/ISCAS45731.2020.9180754

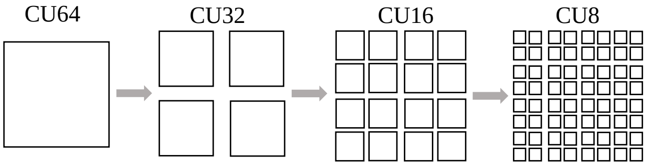

Fig. 2. Quad-tree based CTU partitioning structure.

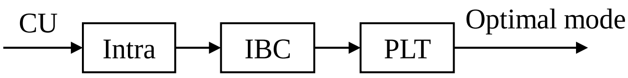

Fig. 3. Exhaustive mode searching strategy for a CU.

local features from a CU. To reduce the computational overhead of the CNN inference time, we integrate these two parts to the same network by sharing feature maps. Since a screen content video contains many stationary areas, the temporal correlation of the CU size decision and the mode decision is also utilized for the stationary CTUs. The differences between our contributions and the related schemes can be summarized as 1) Unlike the traditional fast prediction algorithms [7]–[9], [11]–[13] that rely on the limited number of hand-crafted rules or hand-crafted features, the proposed algorithm utilizes CNN as the classifier. As a result, the proposed algorithm automatically learns useful features from the raw samples, and it avoids the risk that humans may ignore some important features during feature extraction. 2). Unlike [10] only utilizes CNN based classifiers that simplify the CU partitioning decision, the proposed algorithm additionally considers the mode decision. Therefore, it achieves further performance improvement.

The rest of this paper is organized as follows. Section II briefly reviews the CU size decision and mode decision in SCC. Section III presents the proposed low-complexity intra prediction algorithm by CNN. The experimental results are presented in Section IV to verify the performance of the proposed algorithm. Finally, section V concludes the paper.

II. Review of the CU Size and Mode Decision in SCC

SCC inherits the same quad-tree based CTU partitioning structure from HEVC. As shown in Fig. 2, a CTU of 64×64 pixels can be partitioned to four CUs of 32×32 pixels, and then each CU can be further partitioned into four smaller CUs recursively until the smallest CUs of 8×8 pixels are reached. In this paper, we call them CU64, CU32, CU16, and CU8, respectively. To find the optimal mode of a CU, an exhaustive mode searching strategy is adopted.

A CU in SCC has three mode candidates, which are Intra, IBC, and PLT. The Intra mode from HEVC includes 33 directional modes, plus planar and DC modes. Since a NIB usually has a dominant direction, Intra mode predicts the content in the NIB by copying the boundary samples along this direction. IBC is a block-matching-based technique. Since a SCB has many repeated patterns within the same frame, IBC performs motion estimation in the reconstructed areas of the current frame, and it uses a block vector to denote the position of the best-matched block. PLT is designed based on the characteristics that a SCB contains limited colors. It encodes a CU by selecting several representative colors in the CU, and it denotes the position of each representative color by using an index map. In the original encoder, the exhaustive mode checking strategy checks all modes of Intra, IBC, and PLT, as shown in Fig. 3. For each mode, the SCC encoder calculates the Lagrange rate-distortion (RD) cost J_{mode}

$$J_{mode} = D_{mode} + \lambda \times R_{mode} \quad (1)$$

where $mode \in \{\text{Intra, IBC, PLT}\}$, λ is a Lagrange multiplier, D_{mode} is the distortion and R_{mode} is the bit cost of the CU when selecting *mode*. The mode with the smallest value of *mode* is selected as the optimal mode for the CU. Therefore, the optimal mode selection of a CU is only related to the local content of the CU.

After selecting the optimal mode for all CUs, the RD costs are compared across different CU sizes. If the RD cost of a CU is smaller than the sum of the RD costs of its four sub-CUs, the content is coded by the current CU size. Otherwise, the current CU continues partitioning, and the content is coded by its four sub-CUs. Finally, the CTU partitioning structure is decided as the one with the smallest sum of RD cost. Therefore, the CU size decision is not decided by the local content of a CU but the global content of a CTU.

TABLE I. Details of the proposed network

Layer	Output size	Channel number	Kernel size	Stride
Input	64×64	1	2×2	2×2
Conv1	32×32	8	2×2	2×2
Conv2	16×16	16	2×2	2×2
Conv3	8×8	32	2×2	2×2
Conv4	4×4	64	2×2	2×2
Conv5	2×2	128	2×2	2×2
Conv6	1×1	256	2×2	2×2
Deconv1	2×2	128	2×2	2×2
Deconv2	4×4	64	2×2	2×2
Deconv3	8×8	32	2×2	2×2

III. Proposed Network for Low-Complexity Intra Prediction

To make a low-complexity Intra prediction for SCC, we adopt CNN as the classifier. Since it contains extensive trainable parameters, it automatically learns useful features from the raw samples, which avoids the risk that humans may ignore some important features during feature extraction.

A. *Description of the CNN Based Classifier*

The structure of the proposed network is shown in Fig. 4. The input is the luminance component of a CTU, and it is preprocessed by mean removal before fed to the network. The proposed network can output all mode decisions and CU size decisions of a CTU in a single test. In total, the network contains 6 convolutional layers (Conv1–Conv6) and 3 deconvolutional layers (Deconv1–Deconv3). The details of the proposed network are shown in Table I. Each convolutional or deconvolutional layer is followed by a rectified linear unit (ReLU) activation function.

To make fast mode decision, the luminance component of a CTU goes through the convolutional layers (Conv1–Conv6). For each CU, the network outputs the probability of selecting each mode, i.e., $P(mode)$, $mode \in \{Intra,\ IBC,\ PLT\}$. As shown in Fig. 2, the width/height of a CU is reduced by half when it goes through a partition. To imitate the heavier of the original SCC encoder, we set the kernel sizes of the conv1–

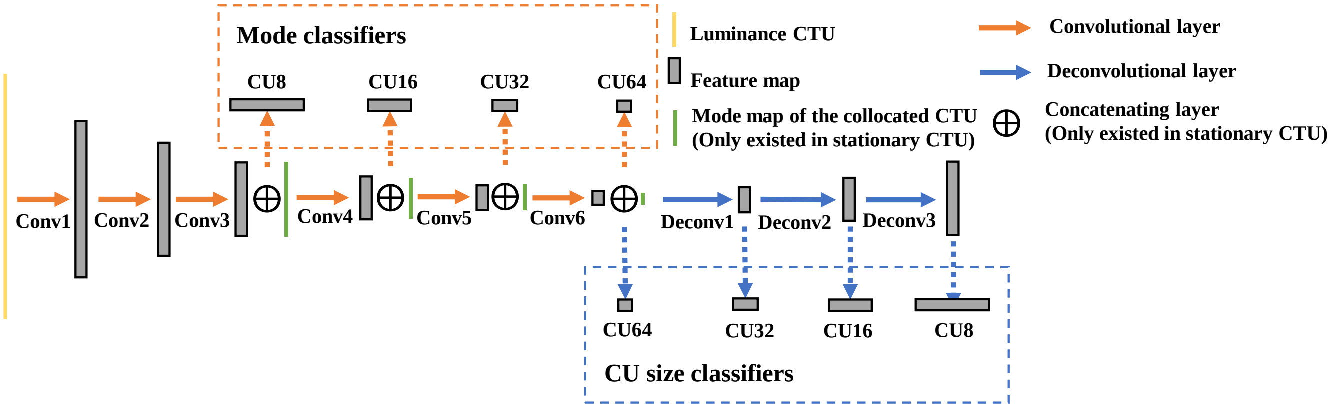

Fig. 4. Structure of the proposed network.

TABLE II. TRAINING SEQUENCES

Sequences	Resolution	No. of Frame
ClearTypeSpreadsheet	1920×1080	300
PptDocXls	1920×1080	200
RealTimeData	1920×1080	600
WordEditing	1920×1080	600
VideoConferencingDoc Sharing	1280×720	300
BigBuck	1920×1080	400
KristenAndSaraScreen	1920×1080	600
MissionControlClip1	2560×1440	600
Viking	1280×720	300
EBULupoCandlelight	1920×1080	250
Seeking	1920×1080	250
ParkScene	1920×1080	240

conv6 to 2×2, and their strides are set to the widths of the kernels to perform non-overlapping convolutions, in accordance with the non-overlapping CU partitioning structure. By using this strategy, the receptive field of a node in feature maps of Conv3–Conv6 is a CU8, CU16, CU32, and CU64, respectively. Therefore, Conv3–Conv6 can extract the local features of CU8, CU16, CU32, and CU64 without introducing the influence from other CUs. Since the mode decision of a CU is only related to its local content, four mode classifiers are designed by using the local features from Conv3–Conv6. Each mode classifier contains a convolutional layer with a kernel size of 1×1, and it is followed by a softmax function to output *P*(*mode*). For example, the feature maps of Conv3 are used to predict the mode decision of CU8. Since the feature map of Conv3 has the size of 8×8, the mode classifiers for CU8 output 64 *P*(*mode*), in accordance with the 64 CU8, as shown in Fig. 2.

On the other hand, the CU size decision in a CTU is not only related to the content of a single CU but also related to the content from other CUs. Therefore, we use three deconvolutional layers (Deconv1–Deconv3) to enlarge the feature maps while extracting global features. Since the receptive field of each node in the feature maps of Conv6 is the entire CTU, the receptive field of each node in the feature maps of Deconv1–Deconv3 also becomes the entire CTU. By using this strategy, the feature maps of Deconv1–Deconv3 introduce the influence of other CUs. By utilizing the global features of Conv6 and Deconv1–Deconv3, four CU size classifiers are designed. For each CU, the network outputs the probability of selecting this CU size, i.e., *P*(*CU*). Similar to the mode classifiers, each CU size classifier also contains a convolutional layer with a kernel size of 1×1, and it is followed by a softmax function to output *P*(*CU*). For example, the feature maps of Deconv3 are used to predict the CU size decision for CU8. Since the feature map of Deconv3 has the size of 8×8, the CU size classifier for CU8 output 64 *P*(*CU*), in accordance with the 64 CU8, as shown in Fig. 2.

Specifically, two CNN models are trained for dynamic CTUs and stationary CTUs, respectively. For dynamic CTUs where the sum of absolute differences (SAD) between the current CTU and its collocated CTU is not 0, only the current CTU is input to the proposed network. For stationary CTUs where the SAD between the current CTU and its collocated CTU is 0, the temporal correlation is additionally utilized as the input. Four mode maps of the collocated CTUs are extracted, and they represent the optimal modes of the collocated CU8, CU16, CU32, and CU64, respectively. The mode maps are concatenated to the corresponding feature maps of Conv3–Conv6, as shown in Fig. 4.

B. *Training of the CNN Based Classifier*

To generate the training data of the proposed network, 12 sequences [14]–[18] are selected to cover various video content, and they are shown in Table II. For each training sequence, 50 frames were extracted with an equal interval, and they were encoded by the original encoder to get the ground-truth labels. Unlike other machine-learning-based algorithms that train a new model for each quantization parameter (QP), we train a single model by using training data from a wide range of QPs, which are QPs of 22, 27, 32, and 37, as defined in the common test conditions (CTC) [19]. Therefore, it can be applied to testing sequences under a wide range of QPs.

The training of the network was implemented in Caffe, and a GPU of GeForce GTX 1080 Ti was used to accelerate the training process. To train the network, the “msra” filter is adopted for parameter initialization, and the “adam” optimizer is adopted to update the trainable parameters. The learning rate policy of “poly” is used to gradually reduce the learning rate, with the base learning rate of 0.01, *power* of 0.9, and the maximum iteration of 100,000. To alleviate the overfitting problem, a weight decay of 0.005 is applied. The cross-entropy function is utilized to calculate the loss of the proposed network, and it is represented as

$$f(\omega,\hat{\omega}) = -\sum_i y(\omega = \omega_i)\,log(P(\hat{\omega} = \omega_i)) \quad (2)$$

TABLE III. PERFORMANCE OF THE PROPOSED ALGORITHM

Categories	Sequences	Proposed	
		BDBR (%)	ΔTime (%)
TGM	ChineseEditing	1.44	-64.02
	Console	3.20	-50.38
	Desktop	3.39	-62.25
	FlyingGraphics	1.34	-26.26
	Map	1.90	-48.72
	Programming	1.17	-48.13
	SlideShow	2.67	-62.07
	WebBrowsing	2.56	-66.46
M	BasketballScreen	2.10	-56.01
	MissionControlClip2	2.27	-56.6
	MissionControlClip3	2.07	-52.5
A	Robot	1.79	-37.1
CC	EBURainFruits	0.51	-41.85
	Kimono1	0.69	-75.82
Average (ALL)		1.94	-53.44

where ω_i denotes the i-class. $y(\omega_i = \omega)$ is 1 if the ground truth class ω is ω_i. Otherwise, it is set to 0. $P(\omega_i = \widehat{\omega})$ denotes the probability that the predicted class $\widehat{\omega}$ is ω_i. In this paper, the loss function of a training sample is defined as the summation of the cross-entropy over all predicted labels in the mode classifiers and the CU size classifiers.

C. Testing of the CNN Based Classifier

After the training of the proposed network, it is invoked by the modified SCC encoder to skip unnecessary CU sizes and mode candidates. Before checking a CU size, the encoder checks its probability $P(CU)$, and the CU size is skipped if

$$P(CU) < \alpha \tag{3}$$

where α is a confidence threshold for CU size decision. Otherwise, if a CU is decided to be checked, the encoder further checks the probability of $P(mode)$, $mode \in \{Intra, IBC, PLT\}$, and a mode is skipped by the encoder if

$$P(mode) < \beta \tag{4}$$

There exists spatial correlation in the CU size decision and mode decision, where a CU tends to have the same size and the same optimal mode as its neighbor CUs. Therefore, the spatial correlation is utilized to adaptively adjust the values of α and β. For the CU that has the same size as its left or top CU, the value of α is reduced by half so that the CU has a larger chance to checked. Similarly, for the mode that is the same as the optimal mode of its left or top CU, the value of β is reduced by half so that the mode has a larger chance to be checked.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed fast mode decision algorithm has been implemented in the SCC reference software, SCM-8.3 [20], and the proposed network is invoked by the DNN tool of OpenCV 3.4.1 to make the low-complexity prediction. To evaluate the performance of the proposed algorithm, the change in encoding time ΔTime and BDBR with QPs of 22, 27, 32, and 37 have been compared with those of the original SCM-8.3 in percentage (%) under All Intra configuration. It should be noted that a negative value of BDBR or ΔTime denotes decrement in percentage as compared with SCM-8.3. The testing sequences are the 14 sequences in CTC [19]. They are divided into four categories according to the video content, where TGM represents text and graphics with motion, M represents mixed content, A represents animation and CC represents camera-captured content.

TABLE IV. PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS

Algorithms	Proposed	
	BDBR (%)	ΔTime (%)
Duanmu *et al*. [9]	3.00	-36.80
Zhao *et al*. [10]	2.67	-53.21
Yang *et al*. [11]	2.72	-49.89
Duanmu *et al*. [12]	3.65	-52.00
Lei *et al*. [13]	2.01	-44.92
Proposed	1.94	-53.44

To skip the unnecessary CU size and mode candidates, the proposed network adopts two confidence thresholds, α and β. Their values control the trade-off between BDBR and ΔTime. If larger values of α and β are used, more encoding time reduction can be achieved. However, it also brings more increase in BDBR. In this paper, the values of α and β are experimentally set to 0.15 and 0.07, respectively. The performance of the proposed network is shown in Table III. It is observed that the proposed algorithm achieves 53.44% encoding time reduction with 1.94% negligible increase in BDBR.

Furthermore, Table IV shows the performance comparison with the existing fast SCC encoding algorithms [9]–[13] in the literature. It is observed that they provide 36.80%–53.21% encoding time reduction with BDBR increased by 2.01%–3.65%. Comparatively, the proposed algorithm outperforms them by achieving the largest encoding time reduction with the smallest increase in BDBR. Specially, Zhao *et al.* [10] is also a CNN based algorithm that only optimizes the CU size decision. However, the proposed CNN based algorithm optimizer both the CU size decision and the mode decision, and it further outperforms Zhao *et al.* [10] by provides a similar encoding time reduction with 0.73% less increase in BDBR. Besides, Zhao *et al.* [10] trains four models for four different QPs of 22, 27, 32, and 37. Comparatively, the proposed method only trained a single model by using mixed training data from the four QPs. Therefore, the memory overhead of the proposed network is also smaller than Zhao *et al.* [10].

V. CONCLUSION

In this paper, a low-complexity intra prediction algorithm was proposed for SCC. CNN is utilized as the classification tool because it contains extensive trainable parameters. First, the proposed network decides if the current CU size should be checked or not by extracting global features from a CTU. If a CU size is decided to be checked, the proposed network further decides the mode candidates of Intra, IBC, and PLT should be checked or not by extracting local features from the CU. Since screen content videos contain both dynamic CTUs and stationary CTUs, two different models are trained for them, respectively. For dynamic CTUs, only the luminance component of a CTU is utilized as the input. For stationary CTUs, the mode maps of the collocated CTUs are additionally utilized as the input. Experimental results show that the proposed network provides an average computational complexity reduction of 53.44% with a negligible increase in BDBR of 1.94% for typical screen content sequences under All Intra configuration.

REFERENCES

[1] Y. Lu, S. Li, and H. Shen, "Virtualized screen: A third element for cloud-mobile convergence," *IEEE Multimedia*, vol. 18, no. 2, pp. 4–11, February. 2011.

[2] 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.

[3] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] X. Xu *et al.*, "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] 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.

[6] G. Bjontegaard, "Calculation of average PSNR differences between rd-curves," document VCEG-M33, VCEG, Austin, Texas, USA, Mar. 2001.

[7] W. Kuang, S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Fast Mode Decision Algorithm for HEVC Screen Content Intra Coding," in *Proc. of Int. Conf. on Image Process.*, pp. 2473-2477, Beijing, China, Sept. 2017.

[8] S.-H. Tsang, W. Kuang, Y.-L. Chan and W.-C. Siu, "Fast HEVC screen content coding by skipping unnecessary checking of intra block copy mode based on CU activity and gradient," in *Proc. APSIPA ASC*, Jeju, Korea, Dec. 2016, pp.1–5.

[9] F. Duanmu, Z. Ma, and Y. Wang, "Fast CU partition decision using machine learning for screen content compression," in *Proc. IEEE Int. Conf. Image Process.*, Quebec, QC, Canada, Sep. 2015, pp. 4972–4976.

[10] L. Zhao, Z. Wei, W. Cai, W. Wang, L. Zeng and J. Chen, "Efficient screen content coding based on convolutional neural network guided by a large-scale database", in *Proc. of Int. Conf. on Image Process.*, pp. 2656-2660, Taiwan, Sept. 2019.

[11] H. Yang, L. Shen, and P. An, "An efficient intra coding algorithm based on statistical learning for screen content coding", in *Proc. IEEE Int. Conf. Image Process.*, Beijing, China, Sep. 2017, pp. 2468–2472.

[12] F. Duanmu, Z. Ma, and Y. Wang, "Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension," *IEEE J. Emerg. Sel. Topic Circuits Syst.*, vol. 6, no. 4, pp.517–531, Dec. 2016.

[13] J. Lei, D. Li, Z, Pan, Z. Sun, S. Kwong, and C. Hou, "Fast intra prediction based on content property analysis for low complexity HEVC-based screen content coding," *IEEE Trans. Broadcast*., vol. 63, no.1, pp.48–58, Mar. 2017.

[14] R. Cohen, "AHG8: 4:4:4 game content sequences for HEVC range extensions development", 14th JCT-VC meeting, document JCTVC-N0294, Vienna, Austria, Aug. 2013.

[15] A. M. Tourapis, D. Singer, and K. Kolarov, "New test sequences for screen content coding", 15th JCT-VC meeting, document JCTVC-O0222, Geneva, Switzerland, Nov. 2013.

[16] H. -P. Yu, W. Wang, X. Wang, J. Ye, and Z. Ma, "AHG8: New 4:4:4 test sequences with screen content", 15th JCT-VC meeting, document JCTVC- O0256, Geneva, Switzerland, Nov. 2013.

[17] J. Guo, L. Zhao, and T. Lin, "Response to B1002 Call for test materials: Five test sequences for screen content video coding", 3th JVET meeting, document JVET-C0044, Geneva, Switzerland, May. 2016.

[18] K. Sharman, and K. Suehring, "Common test conditions", 24th JCT-VC meeting, document JCTVC-X1100, Geneva, Switzerland, May. 2016.

[19] H.-P. Yu, R. Cohen, K. Rapaka, and J. -Z Xu, "Common test conditions for screen content coding", 21th JCT-VC meeting, document JCTVC-U1015-r2, Warsaw, Poland, Jun. 2015.

[20] HM-16.12+SCM-8.3, HEVC test model version 16.12 screen content model version 8.3, [Online], available at: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.12+SCM-8.3/