Journal of Multimedia Information System VOL. 8, NO. 3, September 2021 (pp. 147-158): ISSN 2383-7632 (Online)
http://doi.org/10.33851/JMIS.2021.8.3.147

CNN-based Fast Split Mode Decision Algorithm for Versatile Video Coding (VVC) Inter Prediction

Woon-Ha Yeo[1], Byung-Gyu Kim [1*]

Abstract

Versatile Video Coding (VVC) is the latest video coding standard developed by Joint Video Exploration Team (JVET). In VVC, the quadtree plus multi-type tree (QT+MTT) structure of coding unit (CU) partition is adopted, and its computational complexity is considerably high due to the brute-force search for recursive rate-distortion (RD) optimization. In this paper, we aim to reduce the time complexity of inter-picture prediction mode since the inter prediction accounts for a large portion of the total encoding time. The problem can be defined as classifying the split mode of each CU. To classify the split mode effectively, a novel convolutional neural network (CNN) called multi-level tree (MLT-CNN) architecture is introduced. For boosting classification performance, we utilize additional information including inter-picture information while training the CNN. The overall algorithm including the MLT-CNN inference process is implemented on VVC Test Model (VTM) 11.0. The CUs of size 128×128 can be the inputs of the CNN. The sequences are encoded at the random access (RA) configuration with five QP values {22, 27, 32, 37, 42}. The experimental results show that the proposed algorithm can reduce the computational complexity by 11.53% on average, and 26.14% for the maximum with an average 1.01% of the increase in Bjøntegaard delta bit rate (BDBR). Especially, the proposed method shows higher performance on the sequences of the A and B classes, reducing 9.81%~26.14% of encoding time with 0.95%~3.28% of the BDBR increase.

Key Words: Versatile Video Coding (VVC), Inter Prediction, Fast algorithm, Convolutional Neural Network (CNN), Deep learning.

I. INTRODUCTION

As multimedia technology advances, new types of video formats such as ultra-high-definition (UHD), virtual reality (VR), and 360-degree video has emerged. Accordingly, the needs of novel video coding standard that can support higher resolution videos and better coding efficiency are increasing. Versatile Video Coding (VVC) was developed by the Joint Video Exploration Team (JVET), a collaboration between VCEG and MPEG [1]. It was finalized in July 2020. As the latest video coding standard, the VVC adopts several new coding schemes and tools, such as coding tree unit (CTU) with a maximum size of 128×128, quad-tree plus multi-type tree (QT+MTT) structure of coding unit (CU) partition and affine motion compensation prediction. These new techniques achieve about 50% gain over the HEVC standard in terms of bit rate reduction. However, the computational complexity of both encoding and decoding has also increased sharply.

n [2], the time complexity of VVC Test Model (VTM) was analyzed compared to HEVC Test Model (HM). Figure 1 compares the normalized total average complexity of the HEVC and VVC encoder with all 720p and 1080p test sequences [3]**.** The VVC encoder takes 5, 7, and 31 times as much time as the encoding time of the HEVC under Low Delay (LD), Random Access (RA), and All Intra (AI) configurations, respectively.

In Figure 2, the complexity breakdown of the VVC encoder is presented. The total complexity is broken down into six categories of Inter-prediction (Inter), Intra prediction (Intra), Transform and Quantization (T/Q), Entropy Coding (EC), Loop Filters (LF), and Memory (Mem) operations. Out of all the encoding tools, *inter-coding* accounts for the highest portion of the total encoding time in LD and RA. In addition, the QT+MTT structure causes much more recursive calls to coding tool functions than that of the quad-tree structure of the HEVC.

A new approach is needed for the complexity optimization for the VVC inter-coding and the QT+MTT partitioning. With QT+MTT partitioning structure, a CU can be split among quad-tree (QT), binary-tree (BT), ternary-tree (TT). Also, horizontal (H) and vertical (V) direction split can be applied in BT and TT. Therefore, total of 6 split modes (Non-split, QT, BT_H, BT_V, TT_H, TT_V) are available for a CU. Figure 3 shows the possible split modes, and the Figure 4 represents an example of CTU that is partitioned by the QT+MTT structure. To be more specific, a CTU is first partitioned by the QT structure. Then, the QT leaf nodes that are CUs can be further partitioned by the QT or MTT structure.

For the inter prediction, the VVC encoder takes advantage of the redundancy that exists between pictures

Manuscript received July 30, 2021; Revised August 08, 2021; Accepted August 11, 2021. (ID No. JMIS-21M-07-025)
Corresponding Author (*): Byung-Gyu Kim, Dept. of IT Engineering, Sookmyung Women's University, Seoul, Korea, +82-2-2077-7293, bg.kim@sookmyung.ac.kr.
[1] Dept. of IT Engineering, Sookmyung Women's University, Seoul, Korea, wh.yeo@ivpl.sookmyung.ac.kr, bg.kim@sookmyung.ac.kr

(inter pictures). After partitioned into blocks, motion compensation is applied for each block. Inter prediction mode has mainly two coding methods: Advanced motion vector prediction (AMVP) mode and Merge mode. In AMVP mode, the optimal values of multiple motion vector candidates, the motion vector difference value, the reference picture number, and the uni-/bi-directional prediction mode are encoded. In merge mode, only the optimal value of multiple motion vector candidates is encoded. The AMVP mode has the advantage of freely determining and coding parameters, while the number of bits required for coding the parameter is high, and requires a complex coding process, motion estimation. For the Merge mode, the number of bits required for coding is very small, but the prediction value is inaccurate.

In this study, we define the problem as deciding a split mode for a coding tree unit (CTU) by using a convolutional neural network (CNN).

We propose a CNN architecture called multi-level tree CNN (MLT-CNN). The MLT-CNN is used during the encoding process and is implemented in VVC Test Model (VTM) 11.0. In addition, some additional information including inter-picture information is used to boost the training performance of CNN.

The remainder of this paper is divided as follows: In Section 2, we give a summarization of the related works in HEVC and VVC. In Section 3, we first observe which information can be useful to train the MLT-CNN model, present the MLT-CNN architecture, and then outline the overall algorithm. The experimental results are shown in Section 4. Lastly, the paper closes with a conclusion and a preview of future work in Section 5.

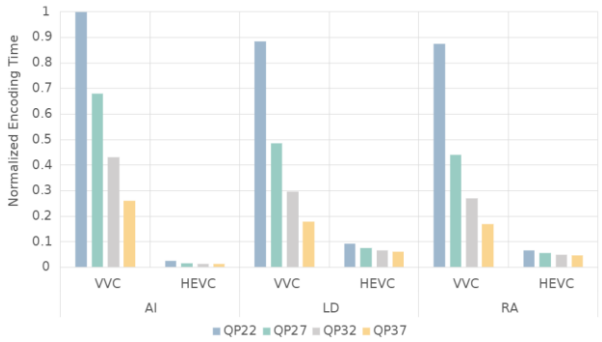

Fig. 1. Normalized average time complexity of the VVC and HEVC encoders [2].

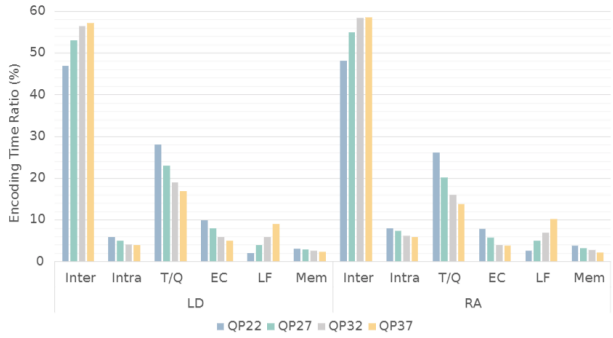

Fig. 2. Complexity analysis for each coding tool in the VVC encoder [2].

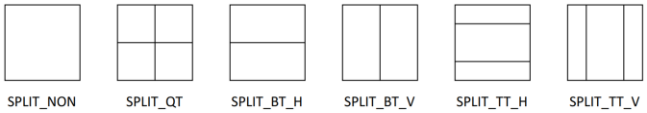

Fig. 3. Possible split types of a CU in the QT+MTT structure.

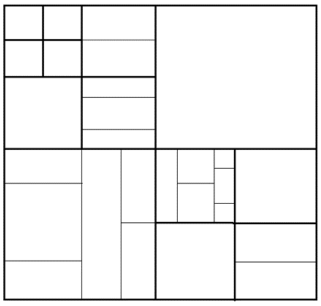

Fig. 4. Example of partitioning with the QT+MTT in VVC [4].

II. RELATED WORKS

2.1. Statistical Approach

Kim et al. proposed an inter mode decision algorithm based on temporal correlation in H.264/AVC [5]. In [6], the RD cost distribution and spatial correlation were jointly adopted for fast CU mode decision in HEVC intra coding. Lim et al. utilized the RD cost as the key feature to skip PU early and terminate early based on Bayes decision rule [7]. In [8], an early skip detection method for HEVC was proposed based on the identification of motionless and regions with homogeneous texture in a video sequence. A fast inter-mode decision algorithm for HEVC was proposed by exploiting both the spatio-temporal correlations and the inter CU correlation of quad-tree structure among neighboring CUs, in which the prediction mode, MV, and RD cost were found strongly correlated [9].

Zhang et al. proposed a scheme based on the relationship between impossible modes and distribution of distortion to accelerate the inter coding in HEVC [10]. Xiong et al. proposed a fast inter CU decision algorithm for HEVC based on the latent sum of absolute difference (SAD) estimation by defining the concepts of two-layer motion estimation and motion compensation RD cost [11]. Lee et al. proposed a PU decision algorithm based on correlation and block motion complexity (BMC) for HEVC [12]. Goswami et al. utilized a Bayesian classifier for skip detection and coding unit termination in HEVC [13].

The advantages of these algorithms are simple, easy to implement, and hardware friendly. Also, they are usually efficient due to the limited complexity overhead. However, these approaches have some limits: Only a small number of critical features can be exploited in each algorithm, and the thresholds of these algorithms are usually determined based on the statistical analyses on a small set.

2.2. Deep Learning Based Approach

For the deep learning-based approach, Jin et al. used a convolutional neural network (CNN) to predict the range of CU depth in each 32×32 CU, skipping the rate-distortion optimization (RDO) search of unused CUs at intra-mode [14]. Another CNN-based approach is to predict CU depth range at inter-prediction mode, which uses a residual CU as the CNN input since the partition relies on the correlation between the current frame and the reference frames [15]. Considering that various CU partition results may satisfy the same depth range, the models in [14], [15] can hardly predict the exact CU partition. Thus, they are limited in reducing the complexity of the VVC.

Subsequently, Galpin et al. [16] suggested a scheme deciding the CU partition directly by predicting all possible CU boundaries between adjacent 4×4 blocks using ResNet model [17]. But the bottom-up decision causes unnecessary calculation when a CTU is non-split or split into only a few large CUs in Kim et al. adopted CNN to predict split or non-split for CU depth decision both inter and intra-coding in the HEVC [18]. Lee et al. have improved visual quality for HEVC using CNN [19]. They constructed a little simple network model for intra prediction mode. In [20], a CNN based fast CU mode decision algorithm is devised for HEVC inter-prediction. The CNN takes the features of the integer motion estimation (IME) and then determines the partition modes in advance. Li et al. proposed a method for VVC intra-coding, where multiple CNNs are trained for various CU sizes to decide whether the CU should be split by which kind of partitioning [21].

Some studies for VVC intra-coding have been done; however, few studies yet consider the characteristics of inter-prediction. As some studies show, CNN based approach is suitable for dealing with images. The most recent work proposed in [15] takes quad-tree plus binary-tree (QTBT) structure. Also, this method predicts each coding unit (CU) depth, therefore it needs more RDO process.

As far as we know, no previous research has studied complexity reduction for inter-prediction within QT+MTT structure using CNN. Thus, we propose a fast split mode decision method that handles with QT+MTT structure by utilizing a CNN to decide CU split type efficiently in the inter-coding process. In this paper, a new CNN architecture that fits QT+MTT structure of the VVC to predict the split mode of each square CU is proposed. In addition, we present a fast decision algorithm using additional temporal information to reduce time complexity for inter-coding in the VVC encoder.

III. PROPOSED METHOD

3.1. Observation and Analysis

In this section, observation and analysis are made on block shapes for various conditions to design a CNN architecture suited to QT+MTT structure.

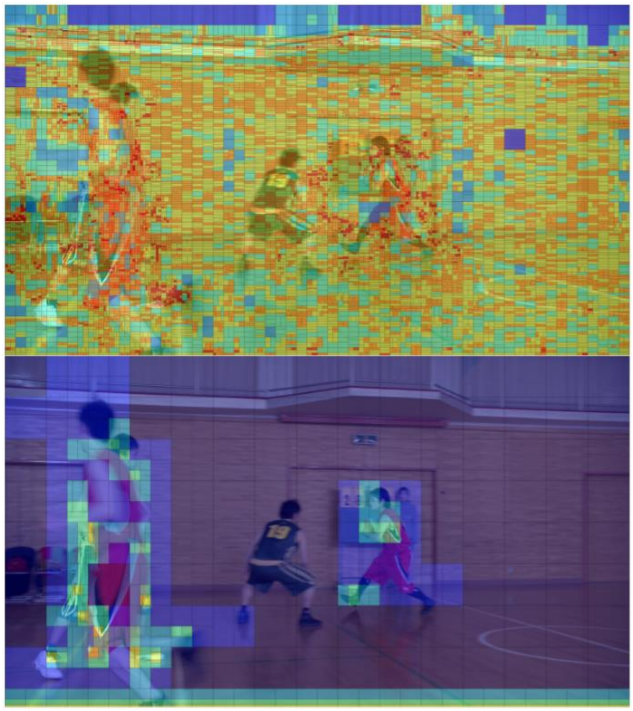

Fig. 5. Top: one frame encoded with all intra (AI) configuration. Bottom: one frame encoded with random access (RA) configuration (POC=1, BasketballDrive sequence).

Figure 5 shows an equivalent frame encoded with all intra (AI) and random access (RA) configuration (picture order count (POC)=1, *BasketballDrive* from VVC test sequences). This represents split tendency difference between intra-picture prediction (intra prediction) and inter-picture prediction (inter prediction). Each block is split finely when encoded with all intra prediction than when encoded with inter prediction mode. During the intra prediction process, a prediction block is constructed using neighboring pixels, and a residual block is obtained by subtracting the original block and the prediction block pixel by pixel. Meanwhile, the prediction block is found by referencing other pictures during the inter-mode prediction process. Therefore, it is not enough to characterize a split mode of a CU with only original CU image in the inter prediction. Furthermore, this leads to why we should make use of the residual image as well as the original image of CU for CNN based inter mode CU partitioning decision.

In this work, we target square-shaped CUs as CNN input. Specifically, 128×128 blocks are used for training. In addition, since square-shaped CU which is split by QT can be further split by QT or MTT, it has more cases to be split.

This means that QT leaf node, e.g., square-shaped CU, has more opportunity to reduce time compared to other shapes through early termination.

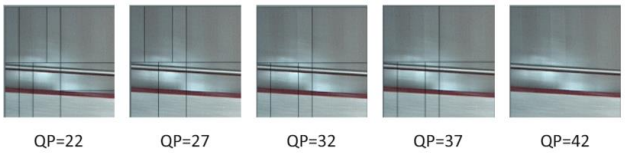

Fig. 6. CTU partitioning difference by QP.

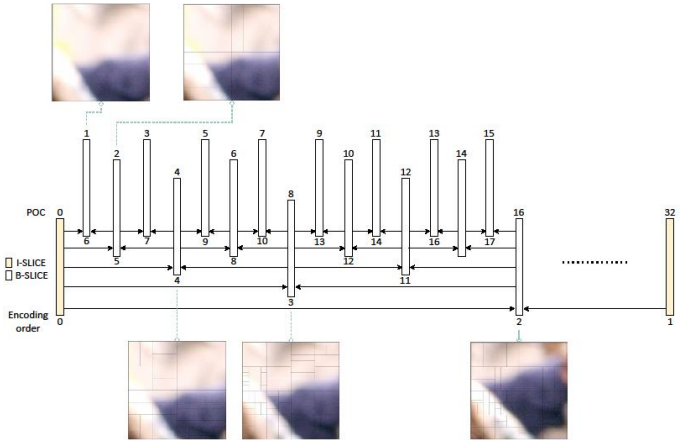

Fig. 7. Picture coding structure of Random Access (RA) configuration.

As shown in Figure 6 and Figure 7, split types are different from quantization parameter (QP) value and POC value. In Figure 6, there shows CTUs in (0, 0) position in *BasketballDrive* sequence encoded with each of five QPs (22, 27, 32, 37, and 42). The QP reflects the compression of spatial details. If QP is large, some details are lost and the image quality is reduced. Therefore, the larger the QP value for the same block is, the shallower the split depth is. Additionally, Figure 6 illustrates CTU in (128, 0) position in *Tango2* sequence with different POC under QP=22 and RA configuration. One can notice that the CTU with a deeper temporal layer tends to be split further. As shown in Figure 7, the first picture in group of pictures (GOP) is encoded as intra picture and all the other pictures within the GOP are encoded as B or GPB pictures. Encoding order in the first GOP is as follows: the picture with POC 8 refers POC 0, and then the picture with POC 4 is encoded referring POC 0 and 8. As such, the temporal layer gets deeper according to the referred pictures.

Figure 8 shows the encoding time ratio of three inter prediction modes (Affine MERGE, MERGE, and Inter ME mode) for all CTUs. Encoding time was measured with *Campfire* sequence from CTC sequences with QP={22, 27, 32, 37, 42}. It shows that two merge modes take much less time than that of the Inter ME process. Therefore, it is entirely reasonable to predict the split mode of CU using a CNN after performing merge modes to decide whether to skip Inter ME and split directly. In this stage, the residual image is obtained from the current best CU for training CNN.

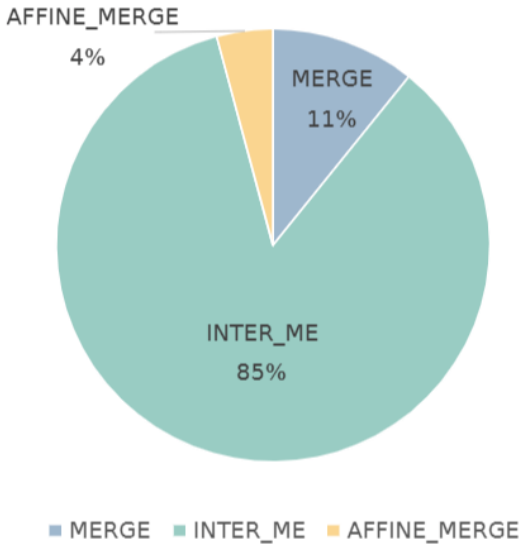

Fig. 8. CTU level encoding time ratio for each inter-prediction mode.

3.2. MLT-CNN

In this study, we propose a multi-level tree CNN (MLT-CNN) architecture that fits QT+MTT structure of the VVC standard. The architecture is modified from Branch-CNN (B-CNN) [22]. The B-CNN is suitable for training labels with a hierarchical structure because it predicts labels in a coarse-to-fine manner. The major difference between MLT-CNN and the B-CNN is that the additional feature vector is used in each level to improve training performance and we use the residual block [23] as a basic block. The MLT-CNN network predicts a split mode among four split modes for CUs with 128×128, since the ternary-tree split is restricted for CTU level. The key feature of the network is that it predicts split mode per each level similar to the split mode tree structure described in Figure 9.

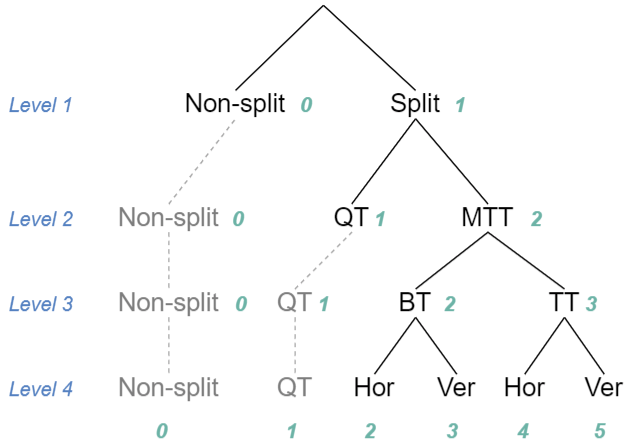

Fig. 9. Split mode tree.

The equation of the cross entropy loss is denoted by:

$$CE = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \quad (1)$$

where y_i and $\hat{y}_i$ are the ground-truth and the predicted score for each class i, and C is the number of classes. Also, $\hat{y}_i$ is calculated by:

$$\hat{y}_i = \frac{e^{s_i}}{\sum_j^C e^{s_i}} \quad (2)$$

The weighted categorical cross-entropy loss is used for training the MLT-CNN:

$$WCE = -\sum_{l=1}^{L} W_l CE_l, \quad (3)$$

where L denotes the number of levels in CNN. The weight W_l changes by iteration so that the loss of each level can be reflected effectively. In this experiment, W_l changes every 150k iterations; W_l=[0.97, 0.02, 0.01] for 0-150k, [0.97, 0.02, 0.01] for 150k-300k, [0.1, 0.1, 0.8] for 300k-450k, and [0, 0, 1] for the last 150k iterations. By setting weights different from iteration, CNN can learn the characteristics of the hierarchical split mode tree. In the early stages of training, the CNN model learns the labels of level 1 (split or non-split) by giving more weights on the loss of level 1. As learning progresses, more weight is given to the lower level. It helps to solve complex problem more effectively than learning from scratch.

Figure 10 shows the architecture of the MLT-CNN. The network consists of four ResBlocks with three levels. First, the 128×128 original and residual image which can be obtained after performing MERGE modes are concatenated as a CNN input. Then, 3×3 kernels at the first convolutional layer is used to extract the low-level features. Before each level prediction, feature maps are further convoluted with residual blocks (ResBlocks).

The detail of ResBlock is depicted in Figure 11. For each level, feature maps are flattened and concatenated with information vector which includes POC, and the CU-level QP. If each of them is not available, it is zeroed. These additional data help improve the performance of training. Finally, the concatenated tensor goes through one fully connected (FC) layer. The effects of using additional information will be shown in Section 4.

3.3. Overall Algorithm

The overall process of the proposed algorithm using the MLT-CNN is shown in Figure 12. The VVC test model, VTM-11.0, was used as the baseline and additional implementation was carried out. After performing MERGE mode in the encoding process, the inference is performed using original, residual image, and POC, and the CU QP.

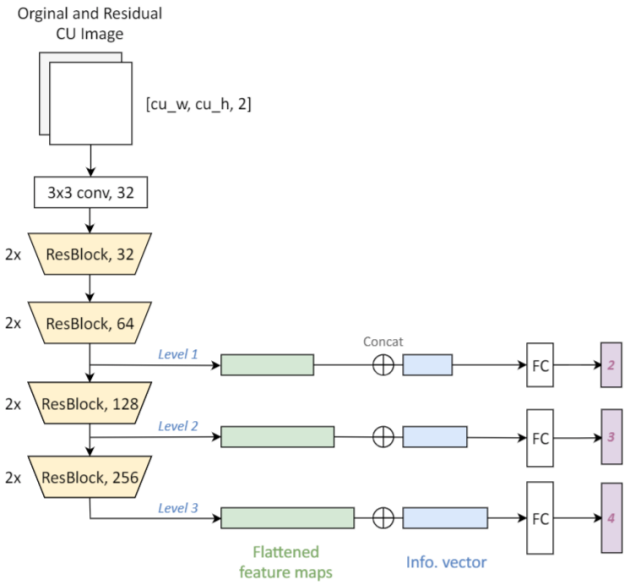

Fig. 10. MLT-CNN architecture for split mode decision.

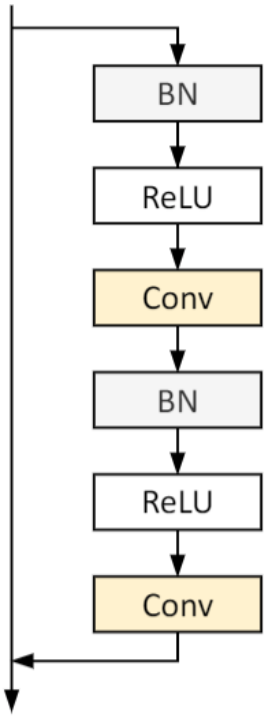

Fig. 11. Residual block [10] used in MLT-CNN architecture.

If the predicted split mode is 0 (Non-split), then it keeps performing other modes such as inter motion estimation and intra prediction, and then no further split occurs. If the predicted split mode is bigger than 0, it means CU needs to be directly split and no other split mode for the current depth is performed. Split modes 1, 2, 3, 4, and 5 represent QT, BT_H, BT_V, TT_H, and TT_V respectively.

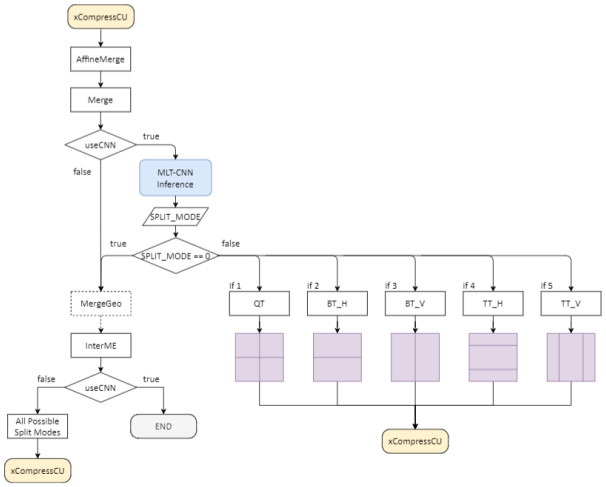

Fig. 12. Overall structure of the proposed scheme.

IV. EXPERIMENTS AND DISCUSSION

4.1. Dataset Construction

A training database for deep video compression called BVI-DVC [24] is used in this paper. All sequences in the BVI-DVC are progressive-scanned at a spatial resolution of 3840×2160, with frame rates ranging from 24 fps to 120 fps, a bit depth of 10 bit, and in YUV420 format. Also, each sequence has total of 64 frames. There are 200 kinds of video clips and they are augmented with 3 resolutions: 1920×1080, 960×540, 480×270. Therefore, the resulting number of sequences is 800 with four different resolutions.

We classified all the sequences into three groups according to the motion degree of each video. Among the whole sequences, 23 kinds of sequences were selected: 9, 7, and 7 sequences in the order of high motion degree. 20 sequences used for training CNN are *AmericanFootballS3Harmonics, BasketballGoalScoredS2Videvo, BricksTiltingBVITexture, BuildingRoofS3IRIS, CharactersYonseiUniversity, ColourfulRugsMoroccoVidevo, FerrisWheelTurningVidevo, FireS18Mitch, HamsterBVIHFR, HongKongMarket4S1Videvo, LakeYonseiUniversity, ManStandinginProduceTruckVidevo, MoroccanCeramicsShopVidevo, MuralPaintingVievo, PillowsTransBVITexture, RunnersSJTU, SquareS1IRIS, StreetDancerS3IRIS, TraditionalIndonesianKecakVidevo, and WatPhoTempleVidevo*, and 3 sequences for validating CNN during training are *CostaRicaS3Harmonics, FireS21Mitch, and ResidentialBuildingSJTU*. Each kind of sequence can have four different resolutions, so 92 sequences were used for building the dataset.

In VVC Test Model (VTM) 11.0 encoder, the residual image is acquired right after Merge mode as the best CU at that point under the Random Access (RA) configuration. The other information such as picture order count (POC), and CU-level QP value are obtained from the decoding process. Table 1 shows the number of train and validation data of 128×128 size.

Table 1. The number of 128×128 images of each class in training dataset.

128×128	NON	QT	BT_H	BT_V	Total
Train	1,136,495	521,929	115,699	129,277	1,903,400
Validation	6,702	14,364	1,387	1,847	24,300
total	1,143,197	536,293	117,086	131,124	1,927,700

4.2. Training Details

All models were trained using the PyTorch deep learning framework [25] with a single GPU for training. Table 2 shows the specifications of the experimental environment. In all experiments, we use Adam optimizer [26] with $\beta_1 = 0.9$, $\beta_2 = 0.99$; initial learning rate 0.0004 that decays with the cosine annealing schedule [27]. To assess the training performance, the validation accuracy over the validation dataset was measured.

To prove the performance with using additional information including temporal features such as residual image and POC helps improve training performance, we compare the validation accuracy between the ResNet trained only with the original CU images (ResNet_O), ResNet trained with the original and residual images (ResNet_OR), and ResNet trained with the original, residual images and the information vector (ResNet_ORI). As shown in Figure 13 and Figure 14, the ResNet_ORI shows the highest validation accuracy, and the ResNet_OR shows higher validation accuracy and lower training loss than those of ResNet_O.

Table 2. Simulation Condition for Training CNN models.

OS	Ubuntu 16.04
CPU	Intel Xeon Processor (Skylake, IBRS) (2.6GHz)
GPU	NVIDIA Tesla V100 (32GB)
Mem	103GB

For training the ResNet, the categorical cross-entropy loss was used as the loss function. The weighted categorical cross-entropy loss which was mentioned in Section 3.2 was used for training the MLT-CNN. Figure 15 shows that the proposed MLT-CNN outperforms the ResNet model in terms of validation accuracy. As the MLT-CNN can learn from multiple levels, it can be trained the complex structure of split mode tree.

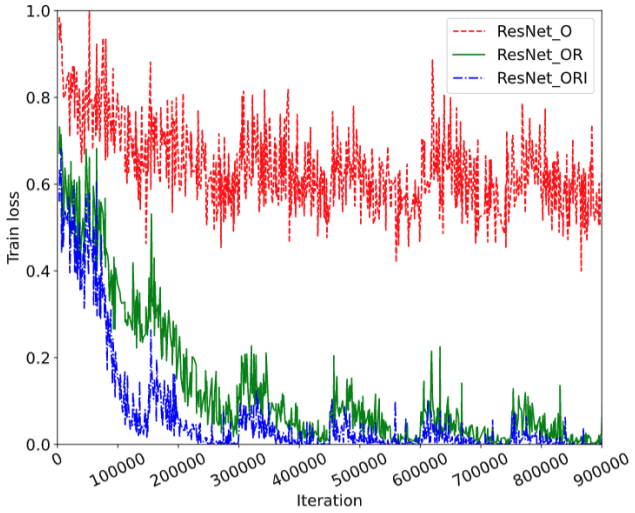

Fig. 13. Train loss of 128×128 ResNet models (ResNet_O, ResNet_OR, and ResNet_ORI).

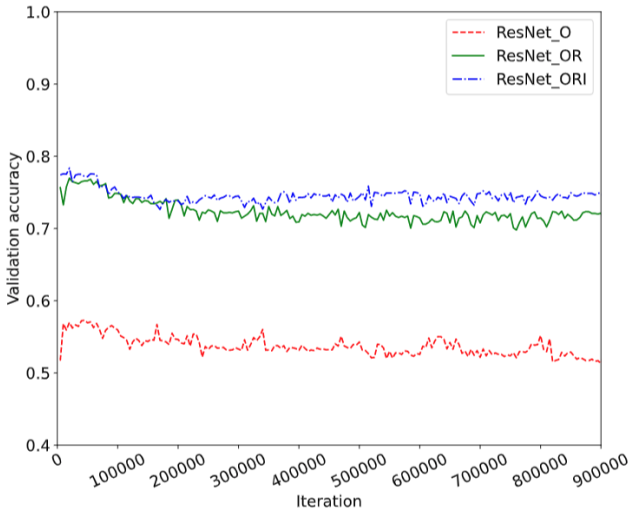

Fig. 14. Validation accuracy of 128×128 ResNet models in every 500 iterations.

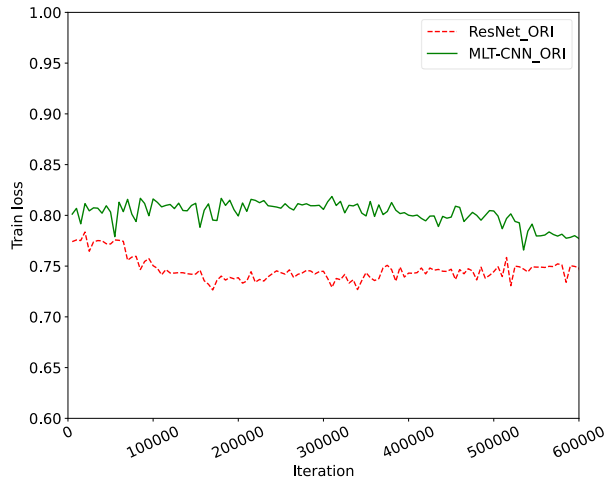

Fig. 15. Validation accuracy of 128×128 ResNet_ORI and the proposed MLT-CNN model in every 500 iterations.

For training the ResNet, the categorical cross-entropy loss was used as the loss function. The weighted categorical cross-entropy loss which was mentioned in Section 3.2 was used for training the MLT-CNN.

Figure 15 shows that the proposed MLT-CNN outperforms the ResNet model in terms of validation accuracy. As the MLT-CNN can learn from multiple levels, it can be trained the complex structure of split mode tree.

4.3. Performance Analysis

In our experiments, all complexity reduction approaches were implemented in the VVC reference software VTM 11.0 [28]. The experiments were conducted on JVET common test sequences [29]. The sequences were encoded as many frames per second (fps) at the random access configuration (using the file *encoder_randomaccess_vtm.cfg*) at five QP values {22, 27, 32, 37, 42}. After encoding, ΔT, which denotes the time-saving rate of encoding compared to the original VTM, was recorded to measure the complexity reduction. In addition, the Bjøntegaard delta bit rate (BDBR) was used to assess the RD performance. All experiments were conducted on a system with the same condition in Table 2.

$$\Delta T = \frac{(T_{VTM} - T_{Proposed})}{T_{VTM}} \times 100\ \%. \qquad (4)$$

Table 3 shows the BDBR increment and time reduction rate per sequence compared between the proposed method and VTM 11.0 anchor under RA test configuration. For all the sequences, the proposed method achieves 11.53% encoding time saving with 1.01% BDBR increase. The results indicate that the time reduction by the proposed framework happens usually on the high-resolution sequences. This is because there are more CTUs in high resolution sequences. In other words, there are more opportunities for time reduction.

By comparing the 3-rd column and the 4-th column in Table 3, less time reduction occurs on QP 22. Since CUs tend to be split on smaller QPs, it spends more time splitting compared to when encoded with higher QPs, meaning non-split predicted CUs cause more time reduction.

Figure 16 and Figure 17 represent the RD curves of *Tango2*, *BasketballDrive*, *RaceHorses*, and *BlowingBubbles* sequences. It shows VTM-11.0 and the proposed method have similar RD curves while the proposed method takes less time for encoding. As the MLT-CNN was trained with 128×128 CUs, the same size CUs are aimed in the encoding process.

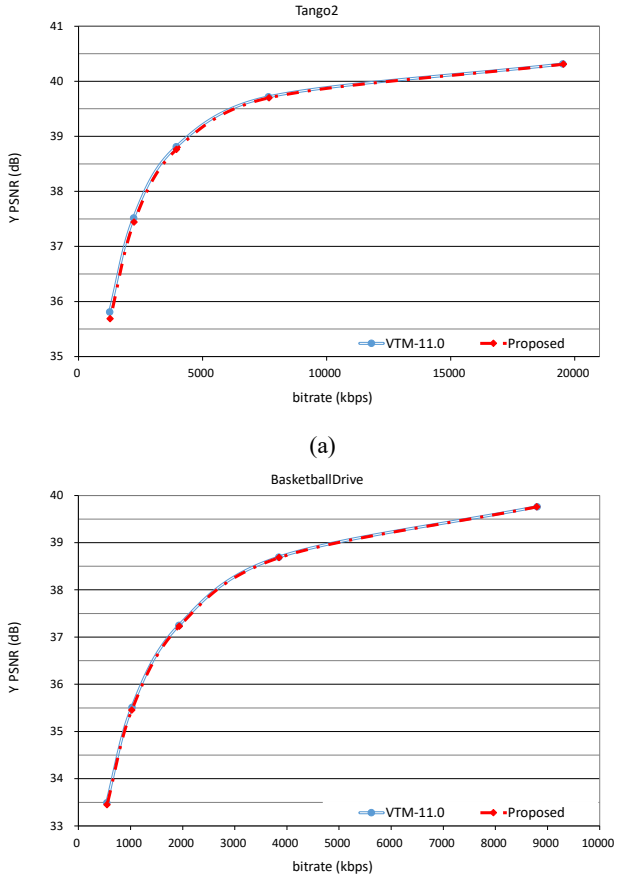

Fig. 16. RD curves of the test sequences under the RA configuration: (a) Tango2, (b) BasketballDrive.

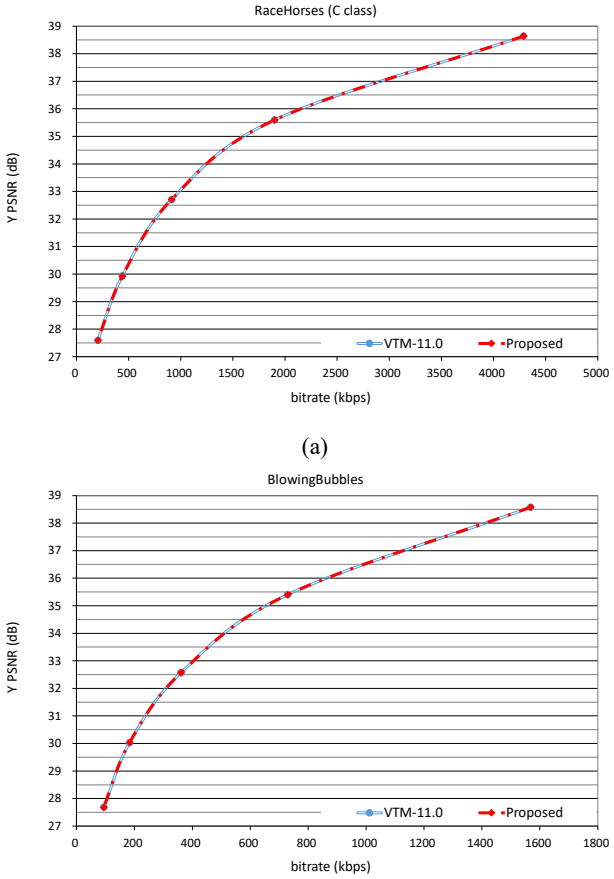

Fig. 17. RD curves of the test sequences under the RA configuration: (a) RaceHorses[C], (b) BlowingBubbles.

Table 3. BD Bit-Rate increase (BDBR: %) and time saving (ΔT: %) performance of the Proposed algorithm (128×128) compared to VTM-11.0 Baseline encoded under the RA for 1 second of each sequence.

Class	Sequence	QP={22, 27, 32, 37, 42}				QP={22, 27, 32, 37}				QP={27, 32, 37, 42}			
		ΔT	BDBR			ΔT	BDBR			ΔT	BDBR		
			Y	U	V		Y	U	V		Y	U	V
A1 (4K)	Tango2	26.14%	3.28%	2.40%	3.42%	25.95%	2.70%	1.72%	1.16%	26.18%	3.32%	0.89%	2.21%
	Campfire	9.81%	0.95%	0.31%	2.05%	7.36%	0.58%	0.00%	1.30%	11.59%	1.20%	0.70%	2.76%
A2 (4K)	CatRobot	25.89%	2.82%	2.41%	2.70%	25.68%	2.58%	1.89%	2.11%	28.66%	2.92%	2.69%	3.00%
A Avg.		**20.61%**	**2.35%**	**1.71%**	**2.72%**	**19.66%**	**1.95%**	**1.20%**	**1.92%**	**23.13%**	**2.55%**	**2.04%**	**3.22%**
B (1080p)	MarketPlace	24.78%	1.81%	0.98%	0.62%	23.25%	1.33%	0.69%	0.75%	28.26%	2.05%	1.04%	0.62%
	RitualDance	17.62%	1.21%	0.65%	1.18%	16.35%	0.92%	0.21%	0.66%	20.26%	1.49%	0.86%	1.54%
	Cactus	16.23%	0.98%	0.51%	0.30%	13.80%	0.82%	0.47%	0.07%	19.45%	1.06%	0.49%	0.29%
	BasketballDrive	17.50%	1.28%	0.93%	1.18%	15.65%	0.99%	0.86%	0.98%	20.62%	1.43%	1.18%	1.56%
	BQTerrace	14.95%	0.62%	-0.08%	0.07%	13.54%	0.67%	0.19%	0.19%	18.38%	0.69%	-0.19%	0.01%
B Avg.		**18.22%**	**1.18%**	**0.60%**	**0.67%**	**16.52%**	**0.95%**	**0.48%**	**0.53%**	**21.39%**	**1.34%**	**0.68%**	**0.80%**
C (WVGA)	BasketballDrive	9.73%	0.22%	0.43%	-0.36%	7.43%	0.19%	0.20%	-0.90%	11.45%	0.20%	0.57%	-0.13%
	BQMall	4.90%	0.16%	-0.42%	0.00%	3.99%	0.12%	-0.50%	-0.14%	5.71%	0.20%	-1.07%	-0.08%
	PartyScene	4.59%	0.04%	-0.01%	-0.36%	2.88%	-0.10%	-0.09%	-0.32%	5.75%	0.13%	-0.15%	-0.64%
	RaceHorses	5.06%	0.20%	-0.35%	0.12%	3.16%	0.15%	-0.68%	-0.49%	5.66%	0.22%	-0.88%	0.11%
C Avg.		**6.07%**	**0.16%**	**-0.09%**	**-0.15%**	**4.36%**	**0.09%**	**-0.27%**	**-0.46%**	**7.14%**	**0.19%**	**-0.38%**	**-0.19%**
D (WQVGA)	BasketballPass	4.28%	0.15%	-0.16%	-0.18%	3.37%	0.06%	-0.08%	-0.10%	4.05%	0.22%	-0.20%	-0.18%
	BQSquare	5.42%	0.12%	0.03%	-0.03%	3.54%	0.09%	0.06%	0.01%	6.30%	0.14%	0.00%	-0.08%
	BlowingBubbles	4.67%	0.09%	-0.03%	0.62%	3.59%	0.12%	-0.03%	0.94%	5.50%	0.15%	-0.47%	0.82%
	RaceHorses	0.25%	0.09%	-0.80%	-0.39%	0.70%	0.16%	-0.95%	-0.66%	0.35%	0.13%	-1.03%	-0.59%
D Avg.		**3.66%**	**0.11%**	**-0.24%**	**0.01%**	**2.80%**	**0.11%**	**-0.25%**	**0.05%**	**4.05%**	**0.16%**	**-0.42%**	**-0.01%**
F	BasketballDrillText	8.90%	0.30%	-0.08%	0.11%	6.87%	0.32%	0.09%	0.22%	10.21%	0.38%	-0.27%	0.51%
	ArenaOfValor	11.81%	0.71%	0.24%	0.50%	10.03%	0.50%	0.19%	0.24%	14.23%	0.87%	0.39%	0.67%
	SlideEditing	9.97%	0.15%	0.20%	0.14%	11.98%	-0.01%	0.16%	0.30%	9.09%	0.15%	-0.05%	0.40%
	SlideShow	8.03%	4.93%	10.53%	15.18%	10.17%	6.20%	6.07%	6.60%	7.26%	4.88%	10.35%	25.50%
F Avg.		**9.68%**	**1.52%**	**2.72%**	**3.98%**	**9.76%**	**1.75%**	**1.63%**	**1.84%**	**10.19%**	**1.57%**	**2.60%**	**6.77%**
Total Avg.		**11.53%**	**1.01%**	**0.88%**	**1.34%**	**10.46%**	**0.92%**	**0.52%**	**0.71%**	**13.10%**	**1.10%**	**0.83%**	**2.00%**

V. CONCLUSION

In this paper, we have aimed to reduce the time complexity of inter-picture prediction mode as the inter prediction accounts for a large portion of the total encoding time. The problem was defined as classifying the split mode of each CU using the proposed multi-level tree (MLT) CNN. The MLT-CNN reflects the tree structure of six split modes: non-split, quad-tree split, binary-tree horizontal split, binary-tree vertical split, ternary-tree horizontal split, and ternary-tree vertical split. By predicting and computing the loss at each level as in the split mode tree, the network learned complex structure effectively. To improve training performance, we observed the split tendency of CU in different conditions to figure out which information affects split mode. As a result, the original and residual image of a CU, the picture order count (POC), and the CU-level quantization parameter (QP) value were considered. For training MLT-CNN, a dataset that contains the original and residual image, POC, CU-level QP, and the ground-truth of one CU was constructed. In this study, we have targeted 128×128 CTUs. 92 video sequences from the BVI-DVC database were used to build the training and validation set.

The overall algorithm including the MLT-CNN inference process was implemented on VVC Test Model (VTM) 11.0. The sequences were encoded at the random access (RA) configuration with five QPs {22, 27, 32, 37, 42}. The time-saving rate of encoding compared over the original VTM was recorded to measure the complexity reduction. Also, the Bjøntegaard delta bit rate (BDBR) was measured to assess the rate-distortion performance. The experimental results showed that the proposed algorithm can reduce the computational complexity by 11.53% on average, and 26.14% for the maximum with an average 1.01% increase in BDBR. Especially, the proposed method showed higher performance on the sequences of the A and B class, reducing 9.81%~26.14% encoding time with 0.95%~3.28% BDBR increase.

The future work is to improve the performance of robustness on the sequences with different resolutions. MLT-CNN models on 64×64, 32×32, and 16×16 CUs should be combined on the proposed algorithm. Also, the performance analysis on the low delay (LD) configuration is needed to validate the overall performance.

REFERENCES

[1] B. Bross, J. Chen, S. Liu, and Y.-K. Wang, “Versatile video coding editorial refinements on draft 10,” JVET-T2001, October 2020.

[2] F. Pakdaman, M. A. Adelimanesh, M. Gabbouj, and M. R. Hashemi, “Complexity analysis of next-generation VVC encoding and decoding,” in *Proceeding of 2020 IEEE International Conference on Image Processing (ICIP)*, pp. 3134-3138, IEEE, 2020.

[3] F. Pakdaman, M. A. Adelimanesh, M. Gabbouj, and M. R. Hashemi, “Dataset for complexity analysis of VVC encoding and decoding,” *IEEE Dataport*, doi: https://dx.doi.org/10.21227/p0rm-4b03, 2020.

[4] J. Chen, Y. Ye, and S. Kim, “Algorithm description for Versatile Video Coding and Test Model 11 (VTM 11),” JVET-T2002, October 2020.

[5] Byung-Gyu Kim, “Novel Inter-Mode Decision Algorithm Based on Macroblock (MB) Tracking for the P-Slice in H.264/AVC Video Coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 2, pp. 273-279, Feb. 2008.

[6] Y. Zhang, S. Kwong, G. Zhang, Z. Pan, H. Yuan, and G. Jiang, "Low complexity HEVC INTRA coding for high-quality mobile video communication,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1492-1504, 2015.

[7] K. Lim, J. Lee, S. Kim, and S. Lee, “Fast PU skip and split termination algorithm for HEVC intra prediction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 8, pp. 1335-1346, 2014.

[8] Kalyan Goswami, Jong-Hyeok Lee, Byung-Gyu Kim, “Fast algorithm for the High Efficiency Video Coding (HEVC) encoder using texture analysis,” *Information Sciences*, vol. 364-365, pp. 72-90, 2016.

[9] L. Shen, Z. Zhang, and Z. Liu, “Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1709-1722, 2014.

[10] J. Zhang, B. Li, and H. Li, “An efficient fast mode decision method for inter prediction in HEVC,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 8, pp. 1502-1515, 2015.

[11] J. Xiong, H. Li, F. Meng, Q.Wu, and K. N. Ngan, “Fast HEVC inter CU decision based on latent SAD estimation,” *IEEE Transactions on Multimedia*, vol. 17, no. 12, pp. 2147-2159, 2015.

[12] Jong-Hyeok Lee, C.-S. Park, B.-G. Kim, Dong-San Jun, Soon-Heung Jung, Jin-Soo Choi, “Novel Fast PU Decision Algorithm for The HEVC Video,” in *Proceeding of IEEE International Conference on Image Processing (ICIP) (IEEE)*, pp. 1982-1985, doi: 10.1109/ICIP.2013.6738408, Melbourne, VIC, Australia, 15-18 Sept. 2013.

[13] Kalyan Goswami, Byung-Gyu Kim, “A Design of Fast High-Efficiency Video Coding Scheme Based on Markov Chain Monte Carlo Model and Bayesian Classifier,” *IEEE Transactions Industrial Electronics*, vol. 65, no. 11, pp. 8861-8871, 2018.

[14] Z. Jin, P. An, L. Shen, and C. Yang, “CNN oriented fast QTBT partition algorithm for JVET intra coding,” in *Proceeding of IEEE Visual Communications and Image*

Processing (VCIP), pp. 1-4, 2017.

[15] Z. Wang, S. Wang, X. Zhang, S. Wang, and S. Ma, "Fast QTBT partitioning decision for interframe coding with convolution neural network," in *Proceeding of the 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2550-2554, IEEE, 2018.

[16] F. Galpin, F. Racapé, S. Jaiswal, P. Bordes, F. Le Léannec, and E. François, "CNN-based driving of block partitioning for intra slices encoding," in *Proceeding of 2019 Data Compression Conference (DCC)*, pp. 162-171, 2019.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.

[18] K. Kim and W. W. Ro, "Fast CU depth decision for HEVC using neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1462-1473, 2018.

[19] Young-Woon Lee, Ji-Hae Kim, Young-Ju Choi, Byung-Gyu Kim, "CNN-based Approach for Visual Quality Improvement on HEVC," in *Proceeding of IEEE International Conference on Consumer Electronics (ICCE)*, pp. 498-500, Lasvegas USA, Jan. 11-14, 2018.

[20] Y. Li, Z. Liu, X. Ji, and D. Wang, "CNN based CU partition mode decision algorithm for HEVC inter coding," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 993-997, IEEE, 2018.

[21] T. Li, M. Xu, R. Tang, Y. Chen, and Q. Xing, "DeepQTMT: A Deep Learning Approach for Fast QTMT-based CU Partition of Intra-mode VVC," *arXiv preprint*, arXiv:2006.13125, 2020.

[22] X. Zhu and M. Bain, "B-CNN: branch convolutional neural network for hierarchical classification," *arXiv preprint*, arXiv:1709.09890, 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.

[24] D. Ma, F. Zhang, and D. R. Bull, "BVI-DVC: a training database for deep video compression," *arXiv preprint*, arXiv:2003.13552, 2020.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proceeding of Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024-8035, Curran Associates, Inc., 2019.

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

[27] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint*, arXiv:1608.03983, 2016.

[28] J. Chen, Y. Ye, and S. Kim, "Algorithm description for Versatile Video Coding and Test Model 11 (VTM 11)," JVET-T2002, October 2020.

[29] S. Liu, A. Segall, E. Alshina, and R.-L. Liao, "JVET common test conditions and evaluation procedures for neural network-based video coding technology," Doc. JVET-T2006, Joint Video Exploration Team (JVET), 2020.

Authors

Woon-Ha Yeo received her BS degree in the Department of IT Engineering from Sookmyung Womens University, Korea, in 2019. In 2019, she joined the Department of Computer Engineering for pursuing her MS degree at Sookmyung Women's University.

Her research interests include video coding standard, deep learning, and image classification.

BYUNG-GYU KIM (Senior Member, IEEE) received the B.S. degree from Pusan National University, South Korea, in 1996, the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), in 1998, and the Ph. D. degree from the Department of Electrical Engineering and Computer Science, KAIST, in 2004. In March 2004, he joined the Real-Time Multimedia Research Team, Electronics and Telecommunications Research Institute (ETRI), South Korea, where he was a Senior Researcher. In ETRI, he developed so many real-time video signal processing algorithms and patents and received the Best Paper Award, in 2007. From February 2009 to February 2016, he was an Associate Professor with the Division of Computer Science and Engineering, Sun Moon University, South Korea. In March 2016, he joined the Department of Information Technology (IT) Engineering, Sookmyung Women's University, South Korea, where he is currently a Full Professor. He has published over 250 international journal articles and conference papers, patents in his field. His research interests include image and video signal processing for the content-based image coding, video coding techniques, 3D video signal processing, deep/reinforcement learning algorithm, embedded multimedia systems, and intelligent information system for image signal processing.

Dr. Kim is a Professional Member of ACM and IEICE. He also served or serves on Organizing Committee of CSIP 2011, a Co-Organizer of CICCAT2016/2017, Mining Intelligence and Knowledge Exploration (MIKE 2017), The Seventh International Conference on Advanced Computing, Networking, and Informatics (ICACNI2019), the EAI 13-th International Conference on Wireless Internet Communications Conference (WiCON 2020), and the Program Committee Members of many international conferences. He has received the Special Merit Award for Outstanding Paper from the IEEE Consumer Electronics Society, at IEEE ICCE 2012, the Certification

Appreciation Award from the SPIE Optical Engineering, in 2013, and the Best Academic Award from the CIS, in 2014. He has been honored as an IEEE Senior Member, in 2015. He has also received the Excellent Paper Award from the IEEE Consumer Electronics Society, at IEEE ICCE 2021. He has been serving as a Professional Reviewer in many academic journals, including IEEE, ACM, Elsevier, Springer, Oxford, SPIE, IET, MDPI, IS&T, and so on.

In 2007, he has served as an Editorial Board Member for the International Journal of Soft Computing, Journal of Signal and Information Processing, and the Journal of Engineering and Applied Sciences. He has been serving as an Associate Editor for Circuits, Systems and Signal Processing (Springer), The Journal of Supercomputing (Springer), The Journal of Real-Time Image Processing (Springer), Heliyon-Computer Science (Cell press), CAAI Transactions on Intelligence Technology (IET), Electronics (MDPI), Sensor (MDPI), and Applied Sciences (MDPI). Since March 2018, he has been serving as the Editor-in-Chief for Journal of Multimedia Information System (KMMS) and an Associate Editor for IEEE Access (IEEE), SN-Computer Science (Springer Nature). He has been serving as a Topic Editor for Sensors and Electronics (MDPI).