Learning-Based Fast Splitting and Directional Mode Decision for VVC Intra Prediction

Yuanyuan Huang, Junyi Yu, Dayong Wang, Xin Lu, Frédéric Dufaux, Hui Guo, Ce Zhu

▶ **To cite this version:**

Yuanyuan Huang, Junyi Yu, Dayong Wang, Xin Lu, Frédéric Dufaux, et al.. Learning-Based Fast Splitting and Directional Mode Decision for VVC Intra Prediction. IEEE Transactions on Broadcasting, In press, 10.1109/TBC.2024.3360729 . hal-04467050

HAL Id: hal-04467050
https://hal.science/hal-04467050

Submitted on 20 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning-Based Fast Splitting and Directional Mode Decision for VVC Intra Prediction

Yuanyuan Huang, Junyi Yu, Dayong Wang, Xin Lu, Frederic Dufaux, *Fellow, IEEE*, Hui Guo, Ce Zhu, *Fellow, IEEE*

***Abstract*—As the latest video coding standard, Versatile Video Coding (VVC) is highly efficient at the cost of very high coding complexity, which seriously hinders its practical application. Therefore, it is very crucial to improve its coding speed. In this paper, we propose a learning-based fast split mode (SM) and directional mode (DM) decision algorithm for VVC intra prediction using a deep learning approach. Specifically, given the observation that the SM distributions of coding units (CUs) of different sizes are significantly distinct, we first design the neural networks separately and train the SM models for all CUs of different sizes to obtain the probability of SMs and skip the unlikely ones. Second, given a similar observation that the DM distributions of CUs of different sizes are distinct, we design neural networks to train the DM models for all CUs of different sizes separately to obtain the probabilities of DMs, and then adaptively select candidate DMs based on probabilities of their located SMs. Third, after an SM is checked, we select its probability, residual coefficients, rate-distortion (RD) cost, etc. as features, and design a lightweight neural network (LNN) model to early terminate SM selection. Experimental results demonstrate that the proposed algorithm can reduce the encoding time of VVC by 70.73% with 2.44% increase in Bjøntegaard delta bit-rate (BDBR) on average.**

***Index Terms*—Versatile video coding, split mode, directional mode, early termination, deep learning.**

I. Introduction

This research is supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62162054 and Grant 62020106011; in part by the National Key Research and Development Program of China under Grant 2023YFC3806003; in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZDK202100604; and in part by Guangxi Key Laboratory of Machine Vision and Intelligent Control under Grant 2023B05.*(Corresponding author: Junyi Yu and Dayong Wang)*

Y. Huang is with the Department of Network Engineering, Chengdu University of Information Technology, Chengdu 610225, China (e-mail: iyyhuang@hotmail.com)

J. Yu is with School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China(e-mail: yujy95@qq.com).

D. Wang is with Chongqing Key Laboratory on Big Data for Bio Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China(e-mail: wangdayong@cqupt.edu.cn).

X. Lu is with the School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, United Kingdom (e-mail: xin.lu@dmu.ac.uk).

F. Dufaux is with the Laboratoire des Signaux et Systèmes,Université Paris-Saclay, CNRS, CentraleSupélec, 91192 Gif-sur-Yvette, France (e-mail: frederic.dufaux@l2s.centralesupelec.fr).

H. Guo is with the Guangxi Key Laboratory of Machine Vision and Intelligent Control, Wuzhou University, Wuzhou 543002, China (e-mail: guohui928@qq.com).

C. Zhu is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: eczhu@uestc.edu.cn).

WITH the rapid development of information technology, various video applications have been widely used in our daily lives, such as network broadcasting, video conferencing and smartphone communications. Meanwhile, users are demanding higher video quality and various ultra-high definition (UHD) video applications are becoming popular, such as 4K/8K video and virtual reality (VR) video, causing the explosive growth of visual data. The previous generation of coding standard, namely High Efficiency Video Coding (HEVC) [1], has gradually failed to meet the market demand for more efficient coding efficiency targets for high-resolution video. As a result, the Joint Video Exploration Team (JVET) has developed the latest generation of video coding standard, i.e., Versatile Video Coding (VVC) [2]. Thanks to a range of new tools, such as the quad-tree plus multi-type tree (QTMT) structure of coding unit (CU) partitioning and the additional intra prediction mode (IPM), VVC achieves significantly higher coding efficiency compared with HEVC [3]. However, its computational complexity is also greatly increased, making VVC unsuitable for real-time applications [4].

In VVC, a CU can be recursively divided into four sub-CUs by means of the quad-tree (QT) structure. A multi-type tree (MTT)-based coding structure is then used to further divide the quad-tree leaf nodes. The MTT structure consists of a binary-tree-horizontal (BTH) splitting, a binary-tree-vertical (BTV) splitting, a ternary-tree-horizontal (TTH) splitting, and a ternary-tree-vertical (TTV) splitting. To avoid the redundant CU splitting process, once MTT is selected by a CU, QT will not be used further. Fig. 1 shows an example of the QTMT structure. The QTMT partitioning structure of VVC accounts for most of the increased complexity. Studies have shown that the QTMT-based CU partitioning process accounts for over 97% of the coding time [5]. In addition, there are 67 intra modes in VVC, including two non-directional modes (DC mode and Planar mode) and 65 directional modes (DMs). By examining all the split modes (SMs) and intra modes, VVC selects the SMs and intra modes with the lowest rate-distortion (RD) cost as the best ones. Through the above process, VVC achieves very high coding efficiency, but the coding process is also very complex. The high coding complexity of VVC seriously hinders its practical application, especially in wireless and real-time environments. Therefore, it is very important to improve the coding speed of VVC.

Intra prediction is an important and widely used prediction mode with irreplaceable advantages on compression quality and broadcasting latency. Therefore, improving the coding speed of intra prediction is very desirable. To this end, we

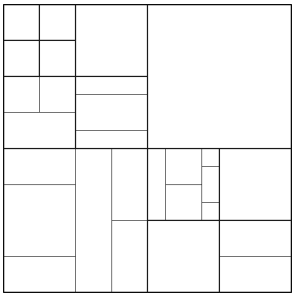


Fig. 1. An example of QTMT structure.

propose a fast learning-based SM and DM decision algorithm for VVC intra prediction in this paper. We first construct a large-scale database containing a sufficiently large number of SMs with various video resolutions and contents. Since the distribution of SMs varies significantly for different CU sizes, we design neural networks separately and train SM models for CUs with the full range of CU sizes to predict candidate SMs. Subsequently, we construct a large-scale database containing a sufficiently large number of DMs with various video resolutions and contents. Since the distribution of DMs also varies for different CU sizes, we develop neural networks and trained the DM models for CU with all different CU sizes to obtain the probability of each DM. Obviously, the probability of an SM is also related to the CU's DM choice. For a high-probability SM, we should examine more DMs and vice versa. Therefore, we combine the probability of each DM with an SM probability to adaptively select candidate DMs. Finally, after the evaluation of an SM, its related RD cost and residual coefficients, etc. are then obtained, and this gives a good indication of the coding performance of this SM. We choose the SM probability, residual coefficient and RD cost as features and train a lightweight neural network (LNN) model to terminate the SM selection process early.

The contributions of this paper are as follows:

(1) We establish a large-scale database to learn the SM selection of intra mode VVC, and propose deep SM models for all different-sized CUs.

(2) We establish a large-scale database to learn the DM selection of intra mode VVC, and propose deep DM models for all different-sized CUs. We then develop a method to adaptively select candidate DMs.

(3) We propose an LNN model to terminate the SM selection early.

The remainder of this paper is organised as follows. Section II discusses related works. Section III provides an overview of the proposed method. Section IV presents the probability-based SM prediction. Section V describes the probability-based DM prediction. Section VI explains the probability-based SM early termination. Section VII discusses and analyses the experimental results. Finally, Section VIII concludes this study and outlines the plans for future work.

II. Related Works

Many approaches have been proposed to improve the speed of video coding. These methods can be roughly classified into three categories, namely statistical analysis-based methods, machine learning-based methods and deep learning-based methods. The representative methods in each category are reviewed as follows.

A. Statistical Analysis-Based Methods

These approches first use statistical methods to build the relationship between some features, such as textures, residual coefficients, RD cost and the CU partitions and intra mode selections, and then use the relationship to skip unlikely CU partitions and intra modes.

Huo et al. [6] propose a fast RDO method for depth maps during its calculation process. Huang et al. [7] control the maximum depth of QuadTree with nested Multi-type Tree to precisely control the encoding complexity. Liu et al. [8] propose a flexible complexity optimization method for 360-degree video coding based on different complexity constraints in diversified broadcasting scenarios. Wang et al. [9] discovered that the RD costs for both inter layer reference (ILR) and intra modes follow the gaussian distribution, but their average values are significantly different. A bayes decision rule is then used for the early termination of mode selection. The work in [10] obtains the probability of coding mode and coding depth being selected as the best ones based on spatial and inter-layer correlations, and then combine with both All-Zero Blocks and Partial-Zero Blocks to derive their early termination condition to improve coding speed. Wang et al. [11] proposed a multi-strategy to predict possible depth levels, ILR modes and DMs respectively to improve the coding speed of intra prediction in Scalable High Efficiency Video Coding (SHVC). The work in [12] and [13] first uses inter layer and spatial correlations to skip unlikely depths, and then uses residual coefficients to terminate the mode and depth selection process early. The work in [14] first derives the probabilities of depths and modes based on inter layer and spatial correlations and then combines them with textural and residual coefficients to remove unlikely depths and modes. Kuo et al. [15] predict candidate sized CUs and skip unlikely sized CUs based on temporal and spatial correlations. Li et al. [16] first use the difference between original luminance pixels and predicted ones to develop two decision models, and then use these two models to early skip the vertical binary-tree and horizontal and vertical ternary-tree partition for each CU.

While the above algorithms can improve coding speed to some extent, these statistical analysis-based methods are highly dependent on prior statistical information, resulting in low prediction accuracy for complex situations.

B. Machine Learning-Based Methods

In recent years, many methods using machine learning have been introduced to improve coding speed.

Zhu et al. [17] formulated the CU decision process as a cascaded multilevel classification task and then proposed a fuzzy SVM to predict the likely CUs. Wang et al. [18] use hybrid strategies to predict likely ILR mode, directional mode and coding depth early termination, excluding unlikely ILR mode, directional mode, and coding depth to improve coding

speed. Wang et al. [19] used a naive bayes classifier, which uses temporal and spatial correlations as features to predict the probabilities of depth and ILR mode, thus skipping unlikely depths and ILR modes. Kuang et al. [20] used online learning of bayesian decision rules to predict likely modes and CU sizes in screen content coding (SCC).

Based on the textural features of the current CU and the contextual information of neighbouring CUs, Yang et al. [21] used a decision tree to predict candidate CU partition structures. Based on the information on horizontal binary partitioning, bayesian decision rules were used to predict likely CU partition structures [22]. Wang et al. [23] first proposed the dynamic partition parameter derivation and then used a four-output decision tree to predict likely partition structures. Dong et al. [24] proposed a learning-based classifier to eliminate intra sub-partitions (ISP) and intra block copy (IBC) mode. Zhang et al. [25] developed an improved directed acyclic graph support vector machine (DAG-SVM) model to skip unlikely CU partition structures to improve coding speed.

While the above algorithms can improve coding speed to some extent, these machine learning-based methods are highly dependent on hand-crafted features, which are difficult to obtain and can involve a great deal of work. Deep learning can effectively solve these problems by automatically extracting features from sufficient data.

C. Deep Learning-Based Methods

Recently, deep learning has become a hot research topic in the field of computer vision and video coding. Liu et al. [26] use convolutional neural network to develop a fast depth intra coding approach to improve coding speed of 3D-HEVC. Laude et al. [27] used a CNN classifier to select likely prediction modes in HEVC. Kim et al. [28] selected CUs image values and vector data from the encoding information of CU, and then used CNN to predict likely coding depths. Li et al. [29] modelled CTU partitioning as a three-level classification problem, and then used CNN to predict CTU partitioning. Xu et al. [30] used CNN and long short-term memory (LSTM) networks, respectively, to predict HEVCs intra coding CU partitioning and inter coding CU partitioning. The above algorithms were developed for HEVC. Since the structure of HEVC is very different from that of VVC, these HEVC-oriented methods cannot be directly applied to VVC.

Given the complex QTMT partition structure of VVC, Li et al. [5] used a multi-stage with an early exit mechanism to predict candidate SMs for VVC. However, errors in feature extraction can spread across multiple stages, thus affecting prediction accuracy, especially for small CU sizes. In addition, the convolution operation of residual units is complex, hindering improvements in coding speed. Wang et al. [31] use deep learning approach to predict canidate split types (STs), exclude unlikely STs to improve coding speed. Park et al. [32] selected two useful types of features - explicit VVC features and derived VVC features, and then used an LNN model to terminate the nested ternary tree (TT) block structure based on these features. As only the TT partitions were terminated early, the coding speed can hardly be improved significantly. Furthermore, this work does not take into account the SM distribution of different CU sizes, which may also degrade the coding performance. Furthermore, these VVC-oriented methods have not been investigated for DMs. Therefore, there is still room for further coding speed improvements.

Based on the analysis above, we develop a learning-based fast SM and DM decision algorithm for VVC intra prediction. We first construct a large-scale SM database and train the SM models for all different sizes of CU to predict candidate SMs, then we construct a large-scale DM database and train the DM models for all different sizes of CU to adaptively select the candidate DMs. After the evaluation of an SM, we further use its residual coefficients, RD costs, etc. as inputs and then use LNN to terminate the SM selection early.

III. Overview of the Proposed Algorithm

To improve the coding speed of VVC intra prediction and maintain coding efficiency, we propose three strategies: Learning-Based SM Prediction (LB-SMP), Learning-Based DM Prediction (LB-DMP), and Learning-Based SM Early Termination (LB-SMET). An overview of the algorithm is shown in Fig. 2. We first obtain the candidate SMs via LB-SMP. For the selected SM candidates, we use LB-DMP to predict likely DMs. After the SM is checked, LB-SMET is used to determine whether it is the best SM for early termination.

As shown in Fig. 2, the three strategies are shown on the left, and the flow of the proposed algorithm is shown on the right.

In LB-SMP, we use the image data and QPs to train CNN models for CUs with all different sizes, and then use these models to obtain the SMs probabilities. Afterwards, we sort the SMs in descending order of probability, skipping the unlikely SMs.

In LB-DMP, we select image data, reference pixels and QPs as inputs, then use CNN to train the DM models for CUs with all different sizes and finally combine the probability of the current SM to adaptively select candidate DMs.

In LB-SMET, we select the probability of the current SM, residual coefficients, RD cost, etc. as inputs, and then use LNN to determine whether the current SM is the best one, thus terminating the SM selection early.

With the above three strategies, many unlikely SMs and DMs are skipped, so the coding speed can be significantly improved.

IV. Learning-Based SM Prediction (LB-SMP)

In VVC, a CTU has a default size of 128×128 pixels and it can be divided into 4×4 blocks at minimum for a more flexible partitioning. In addition, a CU can be recursively divided into multiple sub-CUs with a QT structure and an MTT-based coding structure. Thus, there are at most six split modes: no split, QT, BTH, BTV, TTH and TTV, as shown in Figure. 3. The number of SMs for different CUs may vary from 2 to 6 in VVC's intra coding and is listed in the second column of Table I.

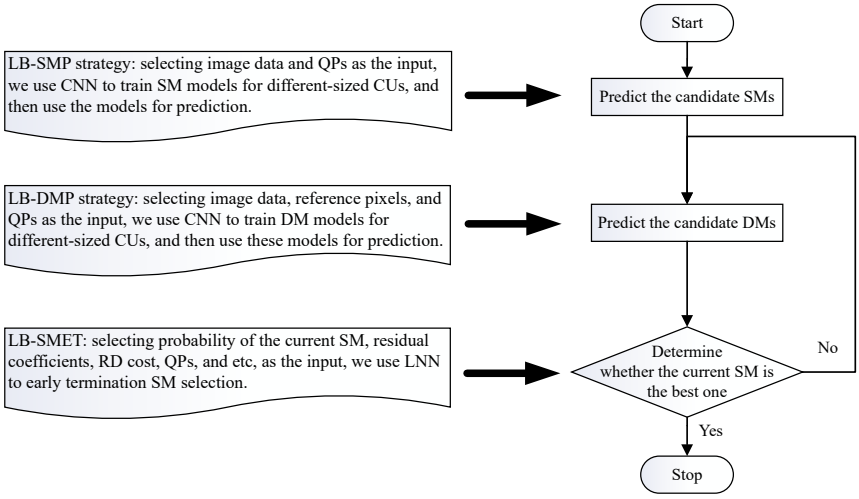


Fig. 2. Overview of the proposed algorithm.

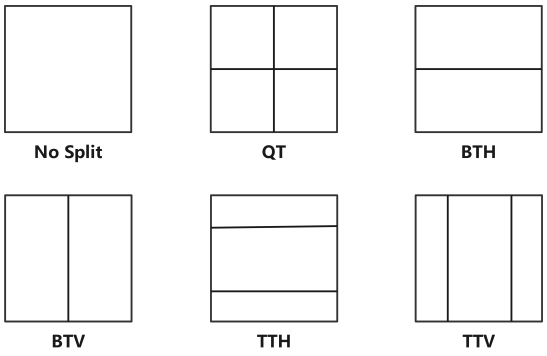


Fig. 3. VVC split modes.

TABLE I
SMs AND THEIR CORRESPONDING MERGED CLASSES OF DIFFERENT CUs

CU Size	Split Modes	Merged classes
64×64	No split, QT	NSC, QC
32×32	No split, QT, BTH, BTV, TTH, TTV	NSC, QC, HSC, VSC
32×16/16×32	No split, BTH, BTV, TTH, TTV	NSC, HSC, VSC
32×8/8×32	No split, BTH, BTV, TTV/TTH	NSC, HSC, VSC
32×4/4×32	No split, BTV/BTH, TTV/TTH	NSC, VSC/HSC
16×16	No split, QT, BTH, BTV, TTH, TTV	NSC, QC, HSC, VSC
16×8/8×16	No split, BTH, BTV, TTV/TTH	NSC, HSC, VSC
16×4/4×16	No split, BTV/BTH, TTV/TTH	NSC, VSC/HSC
8×8	No split, BTH, BTV	NSC, HSC, VSC
8×4/4×8	No split, BTV/BTH	NSC, VSC/HSC

CNNs are widely used for image and video processing and can achieve excellent performance. To efficiently predict candidate SMs, we use CNNs to extract features. More specifically, we first build a database to train and test the CNN models to predict VVC SMs. Then, we design the structure of the CNN models. Finally, we use these models to obtain the probabilities of SMs for the current CU and select likely SMs based on the probabilities. The proposed approach is described in detail below.

A. Database Construction

To train and test the neural network models, we need to build a large-scale database for the SM prediction of VVC intra coding (called the SPVIC database). As both image data and QPs are closely related to the SM selection, we choose both of them as components of the database. The image data are derived from 204 original video sequences [5], which come with different resolutions and content. These video sequences were divided into three non-overlapping sets, including a training set (160 sequences), a validation set (22 sequences) and a test set (22 sequences).

According to the Common Test Conditions (CTC) [33], the QPs are set to 22, 27, 32 and 37 with All-Intra (AI) configuration for testing. All video sequences and images are encoded by the VVC reference software VTM-10.2. The ground truth SMs of the CUs can be obtained accordingly. The SPVIC database is constructed through the process described above. Each sample in the database includes the image data, CUs QP value, as well as its corresponding ground truth SM.

B. Design of the Neural Network Architecture

Before designing the neural network, we first investigate the SM distribution of CUs. For this purpose, we conduct extensive experiments. In order to cover a wide range of resolutions, we choose one sequence each from classes A1, A2, B, C, D and E in our experiments. More specifically, the test sequences are listed as follows: Tango2 in class A1; CatRobot in class A2; MarketPlace in class B; BQMall in class C; Basketballpass in class D; and FourPeople in class E. As these sequences cover different sorts of movements and textures, from simple to complex, they are representative. The corresponding SM distributions for different CUs are listed in Table II.

As there are too many different CUs, the SMs of 64×64, 32×32 and 32×16 CUs are listed in Table II, where NS, QT,

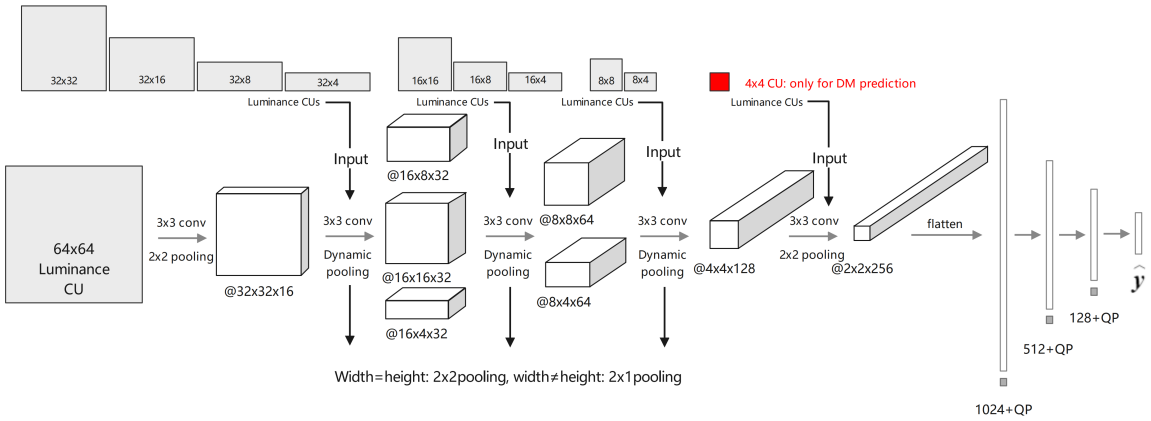


Fig. 4. Proposed neural network architecture for SM/DM prediction(conv and pooling refer to the convolutional layer and pooling layers, respectively).

TABLE II
SM DISTRIBUTIONS (%) OF DIFFERENT CUS

Class	Sequence	QP	64×64		32×32						32×16				
			NS	QT	NS	QT	BH	BV	TH	TV	NS	BH	BV	TH	TV
A1	Tango2	22	9	91	20	9	28	28	7	8	40	15	33	2	10
		27	31	69	48	3	21	21	3	4	69	8	18	1	4
		32	43	57	55	3	19	18	3	2	76	6	13	1	4
		37	54	46	64	1	15	15	2	3	83	5	10	0	2
A2	CatRobot	22	12	88	18	27	16	24	5	10	35	12	33	4	16
		27	23	77	26	16	15	29	5	9	44	12	27	4	13
		32	29	71	31	10	15	30	5	9	49	11	25	4	11
		37	36	64	40	4	15	29	4	8	56	12	22	4	6
B	MarketPlace	22	3	97	11	19	34	21	10	5	42	15	30	3	10
		27	6	94	15	12	38	19	12	4	49	19	23	3	6
		32	11	89	23	7	37	19	11	3	55	17	20	3	5
		37	21	79	34	4	32	18	8	4	65	13	16	3	3
C	BQMall	22	0	100	5	62	8	12	6	7	23	20	27	16	14
		27	0	100	6	52	10	15	7	10	28	17	26	15	14
		32	1	99	8	41	13	20	7	11	29	19	28	11	13
		37	4	96	10	27	16	27	8	12	31	18	28	11	12
D	BasketballPass	22	0	100	4	46	21	8	13	8	21	34	13	26	6
		27	0	100	6	38	24	12	14	6	30	35	16	15	4
		32	3	97	10	29	24	16	14	7	36	29	16	13	6
		37	9	91	19	22	23	18	12	6	45	23	15	11	6
E	FourPeople	22	7	93	10	37	16	20	6	11	36	21	22	8	13
		27	9	91	14	33	16	20	7	10	39	18	23	8	12
		32	12	88	16	23	20	23	7	11	40	18	23	8	11
		37	16	84	21	13	23	26	8	9	44	18	22	8	8

BH, BV, TH and TV refer to No split, QT, BTH, BTV, TTH and TTV, respectively. We can see from Table II that the SM distributions of these three CUs are significantly different. In addition, even within the same sequences and CUs, their SM distributions are significantly different when different QPs are used. It is clear that QPs are closely related to the selection of SM. Extensive experimental results show that the SM distributions of all different CUs are also different. Clearly, if all CUs use the same model to predict candidate SMs, the corresponding predictions cannot always obtain the best performance. Therefore, we should train separate models for different CUs.

In VVC, the default size of CTU is 128×128, and there can be up to six partition types. However, only 64×64, 32×32, 32×16, 16×32, 32×8, 8×32, 32×4, 4×32, 16×16, 16×8, 8×16, 16×4, 4×16, 8×8, 8×4 and 4×8 CUs are partitioned in VVC's intra coding, and their corresponding SMs are in the second column of Table I. Many CUs have their corresponding transposed CUs, such as 32×16 CU and its transposed 16×32 CU. To avoid training redundant models, we only train the CUs whose widths are larger than their heights. Therefore, we only train the models for 64×64, 32×32, 32×16, 32×8, 32×4, 16×16, 16×8, 16×4, 8×8 and 8×4 CUs, respectively.

Visual Geometry Group (VGG) [34] can obtain excellent performance in CNNs with a 3×3 convolutional kernel and a 2×2 pooling kernel for training and testing. Since both the convolutional kernel and pooling kernel in VGG are very small, it is very fast to train and test them. Meanwhile, the prediction accuracy of VGG is very high. Therefore, we adopt it as our backbone to design the network to predict likely SMs and skip the unlikely ones for all CUs.

Fig. 4 shows the proposed neural network architecture for training and inference of the SM prediction. The structure first uses convolutional layers and pooling layers to process image data and finally uses fully connected layers to process vector data. The convolutional layer can extract textural features efficiently. Only 3×3 convolutional kernels with stride 1 and pad 1 are used in the convolutional layers. The pooling layer is implemented to reduce the size of its representation. As there are not only square CUs but also rectangular CUs, a dynamic max-pooling layer is implemented to accommodate both cases. specifically, the 2×2 pooling kernels are used for square CUs and the 2×1 pooling kernels are for rectangular CUs. Going through a 2×1 pooling kernel, the ratio between the width and height of a rectangular CU is reduced by 2. A smaller square CU can be achieved from a large rectangular CU by going through a series of 2x1 pooling kernels. For example, a 32x8 CU is reduced into a 16x8 CU by going through a 2x1 pooling kernel. The 16x8 CU is further reduced into an 8x8 CU by going through this process again.

After being processed in a series of convolutional layers and pooling layers, the vector data is concatenated to feed into the fully connected layer. Furthermore, QPs are closely related to the partitioning of CU. For small QPs, CUs are more likely to be partitioned, and vice versa. Therefore, QPs need to be considered as a neuron. In addition, all convolutional layers and fully connected layers are activated with rectified linear units (ReLU). Finally, we obtain the predicted value $\widehat{y}$ by the softmax output function. The number of $\widehat{y}$ depends on the size of the CU itself and ranges between 2 and 6.

TABLE III
TEST ACCURACY (%) OF SMS OF 32×32 CUS

Predicted SM \ Actual SM	No split	QT	BTH	BTV	TTH	TTV
No split	74	1	9	8	6	8
QT	1	75	8	11	14	15
BTH	12	13	52	14	26	12
BTV	6	9	10	47	7	24
TTH	5	1	20	1	46	1
TTV	2	1	1	19	1	40

C. Training and Testing of Neural Network Architecture

We construct the database and designed the neural network architecture through the process discussed above. Models can be learned by repeating the training and testing processes. In the training phase, the weights of the network are updated by backpropagation. In the testing phase, the predicted results obtained from the model are evaluated in terms of test accuracy and loss value. As there is no overlap between the data sets used in the training and testing phases, the models we trained are generic and effective for different SMs of video sequences.

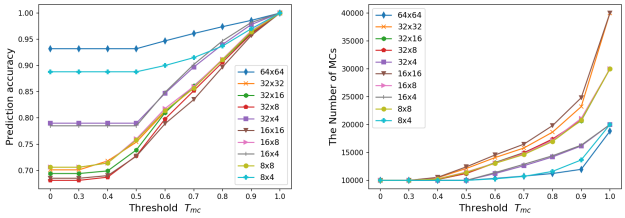
In order to obtain the best-trained model for each CU, we need to get training and testing losses, which are measured by the distance between the predicted SM and the ground truth SM at each phase. It is apparent that the smaller the loss value, the better the trained network. The best model is obtained by continuously reducing the loss values until the smallest loss value is obtained. We use the following cross-entropy loss function to represent the loss value.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j\in M} y_{ij}\log\left(\widehat{y_{ij}}\right), \tag{1}$$

where N is the size of the mini-batch, M is the SMs of the current CU, y_{ij} is the ground-truth value of the ith CU for SM j, and $\widehat{y_{ij}}$ is the corresponding predicted value.

The test accuracy is used to evaluate the trained model, which refers to the ratio of a correctly predicted SM to all predicted the SM in the testing phase. Using the trained model of 32×32 CUs, six SMs are predicted, and their corresponding test accuracies are shown in Table III.

In Table III, the rows and columns represent predicted SMs and actual SMs, respectively. From Table III, we can find that it is difficult to identify BT and TT in the horizontal and vertical direction. For example, 26% of CUs use TTH when the predicted SM is BTH; 24% of CUs use TTV when the predicted SM is BTV. To solve this problem, we merged the ambiguous SMs into one class. Specifically, we merged BTH and TTH into the horizontal split class (HSC), and BTV and TTV into the vertical split class (VSC). Accordingly, we grouped the six SMs into four classes, namely, no split class (NSC), QT class (QC), HSC and VSC. We merged the ambiguous SMs for CUs of sizes 64×64 to 8×4, and their corresponding merged classes (MCs) are listed in the third column of Table I.



(a) Prediction accuracy (b) The Number of MCs

Fig. 5. Prediction accuracy and the number of MCs under different T_{MC} on the validation data.

D. Probability-Based MC selection

For MCs, we retrain MC-based models. Using these models, we can obtain the probabilities of the MCs. The MCs with high probabilities are more likely to be selected, and vice versa. Therefore, we rank the MCs according to the probability from highest to lowest. a_i denotes the probability of the ith MC of the current CU. The sum s_{MC} of the probabilities of the first n MCs is:

$$s_{MC} = \sum_{i=1}^{n} a_i. \tag{2}$$

If s_{MC} is greater than or equal to a threshold value, denoted T_{MC}, we can obtain the first n MCs as candidate SMs. If T_{MC} is too high, the improvement in coding speed is limited. Otherwise, the coding efficiency will deteriorate significantly if T_{MC} is too low. How to choose the best T_{MC} is the key. Since the sum of the probabilities of all MCs is equal to 1, the range of T_{MC} is in [0, 1]. Therefore, we choose some sampling values as T_{MC}, such as 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0, to test their performance. On the validation data, the prediction accuracy and the number of MCs at different T_{MC} are shown in Fig. 5.

From Fig. 5, we can see that both the prediction accuracy and the number of MCs increase as T_{MC} increases. While high prediction accuracy leads to high coding efficiency, the high number of MCs also results in high coding complexity. Therefore, there is a trade-off between coding efficiency and coding complexity. Considering this trade-off, we set 0.7 as the threshold. The MC selection can be written as follows:

$$\sum_{i=1}^{n} a_i \geq 0.7. \tag{3}$$

If the sum of the probability values of the first n MCs is greater than or equal to 0.7, only these modes need to be examined, and the subsequent ones can be skipped directly to enhance coding speed and maintain coding efficiency.

V. LEARNING-BASED DM PREDICTION (LB-DMP)

In the VVC intra prediction, there are 67 intra modes, including two non-DMs (DC mode and Planar mode) and 65 DMs, as shown in Fig. 6.

Since there are 65 DMs, the accuracy of the prediction would be low if we directly grouped them into 65 categories. To solve this problem, we merged some of the adjacent DMs into one category [35], a DM category is denoted as DMC.

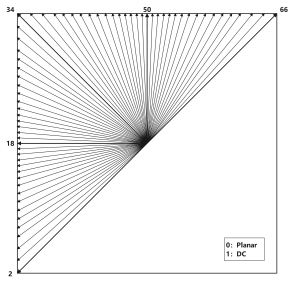


Fig. 6. 67 intra prediction modes of VVC.

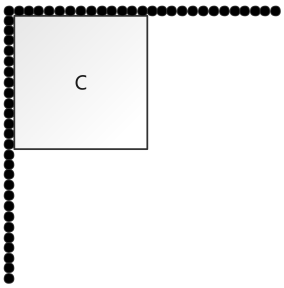
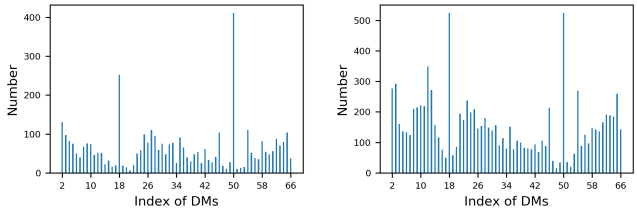


Fig. 7. Reference pixels of a CU.

TABLE IV
DM AND THEIR CORRESPONDING DMC

DM	2-7	8-13	14-22	23-29	30-38
DMC	0	1	2	3	4
DM	39-45	46-54	55-60	61-66	
DMC	5	6	7	8	



(a) Tango2 for 8×4 CU (b) Tango2 for 32×16 CU

Fig. 8. DM distribution of CUs with different sizes.

Therefore, we grouped the 65 DMs into 9 categories, as shown in Table IV.

Similar to the SM prediction, we also use CNN to predict candidate DMCs. Specially, we first establish a database to train and test the DM models. Then, we design the structure of the DM models. Finally, we used these models to obtain the probabilities of DMCs and to select likely DMCs based on the probabilities. The details are as follows.

A. Database Construction

To train and test the neural network models, a large-scale database of DM predictions was built for VVC intra coding (called the DPVIC database). In VVC, the image data of a CU is predicted from its left and top reference pixels along all DMs so that the best DM can be obtained. As shown in Fig. 7, the black dots at the top and to the left of the CU are its reference pixels. The best DM of a CU is closely related to its image data and reference pixels.

Therefore, we should combine the image data of a CU with its reference pixels. Moreover, we also select QPs to predict the candidate DMCs. Therefore, we select all these data as the components of the database. Using the same video sequences and test condition as for testing, the ground-truth DMCs of CUs can be obtained accordingly. Through the above process, the DPVIC database is constructed. Each sample in the database consists of the image data, the reference pixels and the QP value of a CU and its corresponding ground truth DMC.

B. Design, Training and Testing of the Neural Network

Using the aforementioned condition in testing, the corresponding DM distributions of the 8×4 and 32×16 CUs for sequence Tango2 are shown in Fig. 8.

In Fig. 8, the x-axis represents the index of DMs and the y-axis represents the histogram, namely, the corresponding number of DMs in each bin. Fig. 8 (a) shows the DM distribution for the 8×4 CUs and Fig. 8 (b) shows the DM distribution for the 32×16 CUs. From Fig. 8, we can see that the DM distributions of these two categories of CUs are significantly different. Extensive experimental results show that the DM distributions are distinct for CUs of different sizes. Therefore, we should train different models for CUs of different sizes separately.

Therefore, we combined the image data of a CU with its reference pixels and the QP values to train DM models for CUs of different sizes, respectively. As shown in Fig. 4, the CNN structure for DM prediction is the same as that for the SM prediction, except the DM model for the 4×4 CUs. Since 4×4 CUs cannot be further split, we do not need to train a model for 4×4 CUs in SM prediction. However, 4×4 CUs need to perform intra prediction, so the model for 4×4 CUs need to be trained in DM prediction.

As mentioned in the SM prediction, we only train the CUs whose width is greater than their height to avoid training redundant models. For a similar reason, we train the DM prediction models for these CUs only. For those CUs whose width is smaller than their height, we can transpose them in advance. We use the same loss function as for the SM prediction, as shown in Eq. (1).

Through the above process, we obtained the constructed database, the neural network, the loss function and DMCs to train and obtain the best DM models for CUs of different sizes.

C. Probability-Based DMC Selection

After obtaining the best DM models, we can use them to predict the probabilities for all DMCs. The DMCs with high probabilities are more likely to be selected, and vice versa. Therefore, we rank the DMCs according to the probability from highest to lowest. Suppose b_i refers to the probability

of the i-th DMC of the current CU. The sum s_{DMC} of the probabilities of the first m DMCs is:

$$s_{DMC} = \sum_{i=1}^{m} b_i. \tag{4}$$

A great s_{DMC} value leads to high coding efficiency, but it will also results in high coding complexity, and vice versa. How to choose the best s_{DMC} is the key. In addition, DMCs belong to an SM, so probability of current SM should have a strong relationship with DMC selection. If an SM has a high probability to be chosen, more DMCs should be selected to maintain coding efficiency. Otherwise, fewer DMCs should be selected to increase the coding speed. Therefore, we should select the candidate DMCs adaptively according to the probability of SM. Suppose the probability of the SM is p , the DMC selection can be written as follows:

$$\sum_{i=1}^{m} b_i \geq p. \tag{5}$$

If the above conditions are met, only the first m DMCs are selected and the later ones skipped to increase coding speed and maintain coding efficiency. In addition, DC mode and Planar mode are also selected.

VI. Learning-Based SM Early Termination (LB-SMET)

Through the above two processes, the candidate SMs and DMs can be obtained. The residual coefficients and RD costs of the corresponding CUs can be obtained by testing these SMs and DMs. We know that the residual coefficient and RD cost of a CU can be used as a good indicator of prediction performance. If the residual coefficient and RD cost of an SM is very small, this indicates that the corresponding prediction performance is good. Therefore, the subsequent SMs of a CU may not need to be checked, thus the SM selection can be terminated early. Therefore, the residual coefficients and RD costs should be used to determine whether the SM selection can be terminated early. Since this is a relatively simple binary classification problem, we use an LNN model rather than a CNN model to determine whether the SM selection can be terminated early.

In order to use the LNN model to terminate SM selection early, we first select features and then use the LNN model to terminate the SM selection early. The details are as follows.

A. Features Selection in LNN Model

Feature selection is very important in LNN model prediction, and the features are selected as follows:

1) Probability of SM: If the probability of an SM is high, this SM is likely to be selected as the best one and the selection of SM can be terminated early, and vice versa. Therefore, the probability of an SM is highly related to the selection of SM.

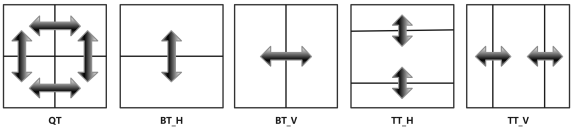


Fig. 9. Two direct neighbor sub-blocks in five SMs.

2) Difference in pixel values of neighboring sub-blocks (DPVNS): As shown in Fig. 9, five SMs need to be further split, including QT, BTH, BTV, TTH, and TTV. If the pixel values of neighbouring sub-blocks in an SM are more different, this SM is more likely to be selected and the SM selection is more likely to be terminated early, and vice versa. Therefore, DPVNS is also closely related to the SM selection. DPVNS is calculated as follows:

$$DPVNS = \frac{1}{n} \sum_{k=1}^{n} \left| \overline{x}_k - \overline{y}_k \right|, \tag{6}$$

where n is the number of all two directly neighboring sub-block sets in an SM, $\overline{x}_k$ and $\overline{y}_k$ are the average pixel values of k-th set of the two directly neighboring sub-blocks, respectively. As shown in Fig. 9, n in QT, BTH, BTV, TTH, and TTV are 4, 1, 1, 2, 2, respectively.

3) Average variance of sub-block pixel values (AVSPV): In general, if the variance of the sub-blocks' pixel values in an SM is small, the SM is more likely to be selected and the SM selection is more likely to be terminated early, and vice versa. Therefore, AVSPV is also strongly related to the SM selection. Since the size of different sub-blocks in SM may be different, the AVSPV is calculated as follows in combination with the size of the sub-blocks.

$$AVSPV = \sum_{k=1}^{n} \frac{w_k \times h_k}{w_0 \times h_0} \sigma_k, \tag{7}$$

where n is the number of sub-block sets in an SM, w_0 and h_0 are width and height of the current CU, w_k and h_k are the width and height of k-th sub-block, respectively. σ_k is the variance of the pixel values of the k-th sub-block.

4) Average variance of the residual coefficients of the Sub-blocks (AVRCS): After examining an SM, the corresponding residual coefficients can be obtained. In general, the prediction of that sub-block is accurate if the variance of the residual coefficients of a sub-block is small. Therefore, this SM is likely to be the best one, and the subsequent SMs do not need to be checked further. Thus, the variance of the residual coefficients is strongly related to the SM selection. By using a method similar to AVSPV, we can calculate and obtain the AVRCS.

For different QP values, different SMs can be selected for the same CU. The width and height of the CU also have an impact on the SM selection. In addition, the RD cost is also closely related to the SM selection. Therefore, the QP values, the width and height of a CU, the RD cost and SMs are also selected as characteristics. Based on the above analysis, these features are listed in Table V.

TABLE V
FEATURES SELECTED FOR LNN MODEL

The number	Features
1	SM
2	Probability of SM
3	DPVNS
4	AVSPV
5	AVRCS
6	QP values
7	width of a CU
8	height of a CU
9	RD cost

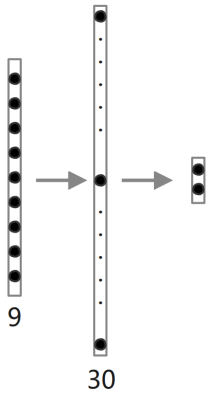


Fig. 10. The LNN structure(The output of two nodes represents the probability that the current SM is or is not the best SM).

B. SM Selection Early Termination in LNN Model

Through the aforementioned process, we obtained the nine features mentioned above. The structure of the LNN model is illustrated in Figure 10. As increasing the number of hidden layers and nodes does not significantly enhance accuracy [31], we opted for a single hidden layer with 30 nodes to greatly reduce calculation complexity. Additionally, we utilized the same dataset and cross-entropy loss function as the SM models. Following this process, we acquired the corresponding dataset, neural network structuFre, and loss function to train and obtain the LNN model.

The LNN uses these features to obtain the probability that the current SM is selected as the best one. The probability of the i-th SM being selected as the best SM is denoted as p_i. Assuming that the first k SMs have already been checked, the total probability of early termination, p_T, is calculated as:

$$p_T = 1 - \prod_{i=1}^{k} (1 - p_i). \tag{8}$$

If p_T is greater than or equal to a threshold value, denoted as T_{sm}, we can terminate SM seletion early. If T_{sm} is too high, the improvement in coding speed is limited. Otherwise, the coding efficiency will deteriorate significantly if T_{sm} is too low. How to choose the best T_{sm} is the key. In order to obtain the best T_{sm}, we choose some sampling values, such as 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, to test their performance. According to CTC [33], all 22 video sequences are tested under the AI configuration with QP values of (22, 27, 32, 37) and the corresponding coding performance is shown in Fig. 11.

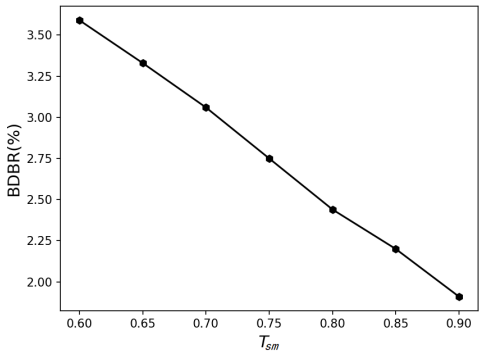


Fig. 11. T_{sm} and the corresponding coding efficiency loss.

In Fig. 11, the x-axis represents the value of T_{sm} and the y-axis represents the average coding efficiency loss, denoted by Bjøntegaard delta bit-rate (BDBR) [36]. From Fig. 11, we can observe that BDBR decreases as T_{sm} increases. In general, if T_{sm} is high, the corresponding coding efficiency is also high. However, a high T_{sm} usually requires checking more SMs, which will lead to high coding complexity. Therefore, there is a trade-off between coding efficiency and coding complexity. We choose 0.8 as the threshold value. If p_T is greater than or equal to 0.8, the subsequent SMs can be skipped directly and the SM selection can be terminated early accordingly.

VII. EXPERIMENTAL RESULTS

To verify the performance of the proposed VVC fast coding algorithm, we performed the evaluations on a server with an Intel(R) 2.0 GHz CPU and 30 GB of RAM using the reference software VTM 10.2. For a fair comparison, we use the GeForce RTX 3090 GPU to speed up the training only but disable it when testing the coding performance. When training the CNN model, we set all weights and bias parameters randomly with Xavier initialization [37]. For each model trained from scratch, we perform 200,000 iterations with a batch size of 128. In every 2000 iterations, we initially set the learning rate to 10^{-4} and then reduced it exponentially by 1%. During the training process, we use the Adam algorithm to optimize the parameters of the trainable components [38], while keeping other parameters unchanged. We use the deep learning framework PyTorch to train the CNN models, and then embed them into the VTM encoder during the test.

According to CTC [33], 22 video sequences in classes A to E are selected in our experiments. All these sequences are tested under the AI configuration with QP values (22, 27, 32, 37). The performance of our proposed algorithm is evaluated by coding efficiency and computational complexity. The coding efficiency refers to the visual quality together with its corresponding bit rate, which is measured by BDBR. BDBR refers to the difference in bit rate at the same PSNR. A positive BDBR represents the loss in encoding efficiency compared to the reference software, or more specifically, the percentage

TABLE VI
ABLATION RESULTS

Class	Sequence	LB-SMP		LB-SMP & LB-DMP		Overall	
		BDBR(%)	TS(%)	BDBR(%)	TS(%)	BDBR(%)	TS(%)
A1	Tango2	1.66	66.68	2.30	68.59	2.68	69.87
	Foodmarket4	1.49	54.48	2.00	55.97	2.39	58.02
	Campfire	1.65	65.00	1.95	66.68	2.08	67.34
A2	CatRobot	1.99	64.94	2.45	66.64	2.61	68.12
	DaylightRoad2	1.97	72.63	2.52	74.58	2.77	76.74
	ParkRunning3	1.45	58.05	1.68	59.32	1.83	61.23
B	MarketPlace	1.03	74.66	1.38	76.41	1.53	78.39
	RitualDance	1.63	64.79	2.07	66.52	2.17	68.35
	Cactus	1.71	71.37	2.16	73.01	2.25	74.50
	BasketballDrive	2.20	74.75	2.74	76.02	2.98	77.89
	BQTerrace	2.58	70.45	3.06	72.24	3.20	74.18
C	BasketballDrill	3.76	63.84	4.24	65.74	4.30	66.81
	BQMall	1.88	69.84	2.41	71.01	2.47	72.79
	PartyScene	1.30	65.81	1.56	67.53	1.60	68.94
	RaceHorses	1.41	67.47	1.76	69.27	1.78	70.35
D	BasketballPass	2.13	67.14	2.56	68.26	2.71	70.07
	BQSquare	1.76	66.77	2.12	68.24	2.10	68.56
	BlowingBubbles	1.40	65.02	1.70	66.61	1.77	68.85
	RaceHorses	1.75	65.35	1.89	66.92	1.91	68.01
E	FourPeople	2.18	72.71	2.61	74.22	2.71	76.43
	Johnny	2.58	72.01	2.98	73.38	3.14	75.47
	KristenAndSara	2.14	71.94	2.52	73.21	2.60	75.09
Average		1.89	67.53	2.30	69.11	2.44	70.73

TABLE VII
OVERALL PERFORMANCE COMPARISONS OF THREE ALGORITHMS

Class	Sequence	Proposed		DQTMT [5]:"faster"		LCCPI [21]	
		BDBR(%)	TS(%)	BDBR(%)	TS(%)	BDBR(%)	TS(%)
A1	Tango2	2.68	69.87	3.49	54.35	1.63	45.28
	Foodmarket4	2.39	58.02	2.78	57.91	5.15	60.49
	Campfire	2.08	67.34	4.17	66.52	2.64	50.12
A2	CatRobot	2.61	68.12	4.88	62.87	1.13	47.17
	DaylightRoad2	2.77	76.74	2.78	65.63	2.18	52.45
	ParkRunning3	1.83	61.23	2.68	63.01	1.25	45.58
B	MarketPlace	1.53	78.39	1.89	65.15	4.20	57.97
	RitualDance	2.17	68.35	2.69	62.98	3.73	61.93
	Cactus	2.25	74.5	2.85	67.70	1.32	60.68
	BasketballDrive	2.98	77.89	3.87	68.5	2.19	55.07
	BQTerrace	3.20	74.18	2.57	64.39	5.25	58.96
C	BasketballDrill	4.30	66.81	4.72	60.98	4.29	60.09
	BQMall	2.47	72.79	3.10	67.45	2.84	55.04
	PartyScene	1.60	68.94	1.86	64.64	2.79	57.2
	RaceHorses	1.78	70.35	2.50	65.68	2.39	54.91
D	BasketballPass	2.71	70.07	3.66	62.62	1.88	54.69
	BQSquare	2.10	68.56	2.04	62.52	1.37	51.36
	BlowingBubbles	1.77	68.85	2.38	61.85	3.17	52.42
	RaceHorses	1.91	68.01	2.92	61.29	1.19	45.47
E	FourPeople	2.71	76.43	3.30	66.91	1.66	51.51
	Johnny	3.14	75.47	5.08	64.35	2.43	57.49
	KristenAndSara	2.60	75.09	3.93	66.11	3.78	58.85
Average		2.44	70.73	3.19	63.79	2.66	54.31

increase in bit rate for a given quality. The computational complexity is measured by the percentage of coding time saved, denoted as TS, which is calculated as follows:

$$TS = \frac{T_s - T_p}{T_s} \times 100\%, \tag{9}$$

where T_s and T_p refer to the standard encoding time of reference software and the encoding time of the proposed algorithm, respectively.

Since the proposed algorithm includes three strategies, we first conduct an ablation study to separately investigate their individual performance and then further compare them with the two state-of-the-art methods to evaluate the performance of the proposed algorithm. The details are as follows.

A. Ablation Study

As mentioned above, our proposed algorithm includes three strategies: LB-SMP, LB-DMP and LB-SMET. The ablation experiments are carried out and the corresponding ablation results are listed in Table VI.

In Table VI, LB-SMP improves the average TS by 67.53% and increases the average BDBR by 1.89%. By introducing LB-DMP, the average TS is improved to 69.11% and the BDBR is increased by 2.30% because some unnecessary DMs are skipped. Finally, by further incorporating LB-SMET, the average improvement in coding speed can reach 70.73%, while BDBR increases to 2.44%. Compared with LB-SMP, both LB-DMP and LB-SMET slightly improve the coding speed due to the fact that many SMs are skipped in LB-SMP and only a few CUs can use LB-DMP and LB-SMET. Therefore, the coding speed is slightly improved while the loss of coding efficiency remains small.

B. Performance Evaluation

In order to evaluate the performance of the proposed algorithm, We compare the proposed approach with the DQTMT algorithm [5] and the LCCPI algorithm [21], in terms of complexity reduction and coding efficiency. To the best of our knowledge, these two algorithms are the state-of-the-art for fast intra coding algorithms in VVC. The corresponding overall performance comparisons are listed in Table VII.

From Table VII, we find that the proposed algorithm achieves the BDBR and TS of 2.44% and 70.73% on average, the DQTMT algorithm achieves 3.19% and 63.79%, and the LCCPI algorithm achieves 2.66% and 54.31%, respectively. The proposed algorithm achieved a smaller average BDBR and a greater average TS compared to the two reference algorithms. Therefore, we can conclude that the proposed algorithm performs better in terms of coding speed and coding efficiency compared to the DQTMT algorithm and the LCCPI algorithm.

The main reasons why the proposed algorithm can effectively improve the coding speed of intra coding for VVC are: (1) Since the SM distributions of CUs of different sizes are distinct, we designed the neural networks separately and trained the SM models for all CUs of different sizes to predict the candidate MCs. (2) Since the DM distributions of CUs of different sizes are also distinct, we designed the neural networks separately and trained the DM models for all CUs of different sizes to obtain the probabilities of DMs, and then adaptively selected the candidate DMs based on the probabilities of their located SMs. (3) After the evaluation of an SM, we selected its probability obtained from the PB-SMP, residual coefficients, RD cost, etc. as features, and used LNN to early terminate SM selection.

VIII. CONCLUSION

In this paper, we proposed a novel learning-based fast SM and DM prediction algorithm for VVC. In order to improve coding speed, we developed three fast strategies, including LB-SMP, LB-DMP and LB-SMET. Since SM distributions

and DM distributions CUs of different sizes are significantly distinct, we designed Neural Networks and trained models for CUs of different sizes to predict the candidate SMs and DMs, and then used LNN to early terminate SM selection. The experimental results verify our approach can outperform other state-of-the-art approaches. The proposed method focuses on luma only. However, given that in VVC intra prediction, both luma and chroma are closely linked to textural features, the proposed method is also applicable to chroma. The proposed algorithm allows for cost effective encoding and transmission of all types videos, especially high-definition videos. Therefore, it is very useful for broadcasters.

For future works, we can further improve the prediction accuracy and reduce the coding time by improving the structure of neural networks and optimizing the loss function. In addition, we also plan to use deep neural networks to accelerate other components of VVC, such as zero block prediction, inter mode prediction, which are important research directions to accelerate VVC coding.

REFERENCES

[1] G. J. Sullivan, J. -R. Ohm, W. -J. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[2] B. Bross, Y. K. Wang, Y. Ye, S. Liu, J. L. Chen, G. J. Sullivan, and J. R. Ohm, "Overview of the Versatile Video Coding (VVC) Standard and its Applications," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736-3764, Oct. 2021.

[3] F. Bossen, X. Li, A. Norkin, K. Sühring, JVET AHG report: Test model software development (AHG3), *Joint Video Experts Team (JVET) of ITU-T and ISO/IEC*, Document JVET-O0003, Jul. 2019.

[4] A. Tissier, A. Mercat, T. Amestoy, W. Hamidouche, J. Vanne and D. Menard, "Complexity Reduction Opportunities in the Future VVC Intra Encoder," *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, 2019, pp. 1-6.

[5] 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," in *IEEE Transactions on Image Processing*, vol. 30, pp. 5377-5390, 2021.

[6] J. Huo, X. Zhou, H. Yuan, S. Wan and F. Yang, "Fast Rate-Distortion Optimization for Depth Maps in 3-D Video Coding," in *IEEE Transactions on Broadcasting*, vol. 69, no. 1, pp. 21-32, March 2023.

[7] Y. Huang, J. Xu, C. Zhu, L. Song and W. Zhang, "Precise Encoding Complexity Control for Versatile Video Coding," in *IEEE Transactions on Broadcasting*, vol. 69, no. 1, pp. 33-48, March 2023.

[8] J. Lin, L. Lin, W. Li, Y. Xu and T. Zhao, "Latitude-Based Flexible Complexity Allocation for 360-Degree Video Coding," in *IEEE Transactions on Broadcasting*, vol. 68, no. 3, pp. 572-581, Sept. 2022.

[9] D. Wang, X. Wang, Y. Sun, W. Li, X. Lu and F. Dufaux, "Gaussian Distribution-based Mode Selection for Intra Prediction of Spatial SHVC," *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 2711-2715.

[10] D. Wang, X. Lu, Y. Sun, Q. Wang, W. Li, F. Dufaux and C. Zhu, "A Probability-Based Zero-Block Early Termination Algorithm for QSHVC," in *IEEE Transactions on Broadcasting*, vol. 69, no. 2, pp. 469-481, June 2023.

[11] D. Wang, C. Zhu, Y. Sun, F. Dufaux and Y. Huang, "Efficient Multi-Strategy Intra Prediction for Quality Scalable High Efficiency Video Coding," in *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 2063-2074, April 2019.

[12] D. Wang, Y. Sun, C. Zhu, W. Li and F. Dufaux, "Fast Depth and Inter Mode Prediction for Quality Scalable High Efficiency Video Coding," in *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 833-845, April 2020.

[13] D. Wang, Y. Sun, W. Li, C. Zhu and F. Dufaux, "Fast Inter Mode Predictions for SHVC," *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019, pp. 1696-1701.

[14] D. Wang, Y. Sun, J. Liu, F. Dufaux, X. Lu and B. Hang, "Probability-Based Fast Intra Prediction Algorithm for Spatial SHVC," in *IEEE Transactions on Broadcasting*, vol. 68, no. 1, pp. 83-96, March 2022.

[15] Y. T. Kuo, P. Y. Chen and H. C. Lin, "A Spatiotemporal Content-Based CU Size Decision Algorithm for HEVC," in *IEEE Transactions on Broadcasting*, vol. 66, no. 1, pp. 100-112, March 2020.

[16] Y. Li, G. Yang, Y. Song, H. Zhang, X. Ding and D. Zhang, "Early Intra CU Size Decision for Versatile Video Coding Based on a Tunable Decision Model," in *IEEE Transactions on Broadcasting*, vol. 67, no. 3, pp. 710-720, Sept. 2021.

[17] L. Zhu, Y. Zhang, S. Kwong, X. Wang and T. Zhao, "Fuzzy SVM-Based Coding Unit Decision in HEVC," in *IEEE Transactions on Broadcasting*, vol. 64, no. 3, pp. 681-694, Sept. 2018.

[18] D. Wang, Y. Sun, W. Li, L. Xie, X. Lu, F. Dufaux and C. Zhu, "Hybrid Strategies for Efficient Intra Prediction in Spatial SHVC," in *IEEE Transactions on Broadcasting*, vol. 69, no. 2, pp. 455-468, June 2023.

[19] D. Wang, Y. Sun, C. Zhu, W. Li, F. Dufaux and J. Luo, "Fast Depth and Mode Decision in Intra Prediction for Quality SHVC," in *IEEE Transactions on Image Processing*, vol. 29, pp. 6136-6150, 2020.

[20] W. Kuang, Y. -L. Chan, S. -H. Tsang and W. -C. Siu, "Online-Learning-Based Bayesian Decision Rule for Fast Intra Mode and CU Partitioning Algorithm in HEVC Screen Content Coding," in *IEEE Transactions on Image Processing*, vol. 29, pp. 170-185, 2020.

[21] H. Yang, L. Shen, X. Dong, Q. Ding, P. An and G. Jiang, "Low-Complexity CTU Partition Structure Decision and Fast Intra Mode Decision for Versatile Video Coding," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1668-1682, June 2020.

[22] T. Fu, H. Zhang, F. Mu and H. Chen, "Fast CU Partitioning Algorithm for H.266/VVC Intra-Frame Coding," *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019, pp. 55-60.

[23] Z. Wang, S. Wang, J. Zhang, S. Wang and S. Ma, "Effective Quadtree Plus Binary Tree Block Partition Decision for Future Video Coding," *2017 Data Compression Conference (DCC)*, 2017, pp. 23-32.

[24] X. Dong, L. Shen, M. Yu and H. Yang, "Fast Intra Mode Decision Algorithm for Versatile Video Coding," in *IEEE Transactions on Multimedia*, vol. 24, pp. 400-414, 2022.

[25] Q. Zhang, Y. Wang, L. Huang, B. Jiang and X. Wang, "Fast CU partition decision for H. 266/VVC based on the improved DAG-SVM classifier model," *Multimedia Systems*, vol. 27, no. 1, pp. 1-14, 2021.

[26] C. Liu, K. Jia and P. Liu, "Fast Depth Intra Coding Based on Depth Edge Classification Network in 3D-HEVC," in *IEEE Transactions on Broadcasting*, vol. 68, no. 1, pp. 97-109, March 2022.

[27] T. Laude and J. Ostermann, "Deep learning-based intra prediction mode decision for HEVC," *2016 Picture Coding Symposium (PCS)*, 2016, pp. 1-5.

[28] K. Kim and W. W. Ro, "Fast CU Depth Decision for HEVC Using Neural Networks," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1462-1473, May 2019.

[29] T. Li, M. Xu and X. Deng, "A deep convolutional neural network approach for complexity reduction on intra-mode HEVC," *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 2017, pp. 1255-1260.

[30] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang and Z. Guan, "Reducing Complexity of HEVC: A Deep Learning Approach," in *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5044-5059, Oct. 2018.

[31] D. Wang, L. Chen, X. Lu, F. Dufaux, W. Li and C. Zhu, "Fast Learning-Based Split Type Prediction Algorithm for VVC," in *2023 IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 2315-2319.

[32] S. -h. Park and J. -W. Kang, "Fast Multi-Type Tree Partitioning for Versatile Video Coding Using a Lightweight Neural Network," in *IEEE Transactions on Multimedia*, vol. 23, pp. 4388-4399, 2021.

[33] J. Boyce, K. Suehring, X. Li, and V. Seregin, *JVET Common Test Conditions and Software Reference Configurations*, document JVETJ1010, Apr. 2018.

[34] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, in *International Conference on Learning Representations*, 2015.

[35] S. Ryu and J. Kang, "Machine Learning-Based Fast Angular Prediction Mode Decision Technique in Video Coding," in *IEEE Transactions on Image Processing*, vol. 27, no. 11, pp. 5525-5538, Nov. 2018.

[36] G. Bjontegaard, Calculation of average psnr difference between rd-curves, in *Proc. 13th VCEG-M33 Meet*. IEEE, 2001, pp. 14.

[37] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proc. AISTATS*, vol. 9, 2010, pp. 249256.

[38] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *International Conference on Learning Representations(ICLR)*, May. 2015, pp. 115.

Yuanyuan Huang received the B.Sc., M.Sc. and Ph.D. degrees from the University of Electronic Science and Technology of China in 2004, 2007 and 2013 respectively, all in computer science. He was a Visiting Scholar with the University of Washington, Seattle, USA, from 2009 to 2011. He had been a Postdoctoral Researcher with the University of Electronic Science and Technology of China. He is currently an Associate Professor with the Chengdu University of Information Technology. His main research interests include image/video processing, big data and artificial intelligence.

Junyi Yu is currently pursuing the M.Sc. degree with the Chongqing University of Posts and Telecommunications, Chongqing, China. His research interest is video coding.

Dayong Wang received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China in 2010. He was a Lecturer with the Hubei University of Arts and Science from 2010 to 2012, and a Postdoctoral Researcher with the Graduate School, Tsinghua University, Shenzhen, from 2012 to 2015. He is currently an Associate Professor with the Chongqing University of Posts and Telecommunications. His main research interest is video coding.

Xin Lu received the B.Sc. and M.Sc. degrees from the Harbin Institute of Technology (HIT), Harbin, China, in 2008 and 2010, respectively, and the Ph.D. degree in computer science from the University of Warwick, Coventry, U.K., in 2013. He is currently a Senior Lecturer with the School of Computer Science and Informatics, De Montfort University (DMU), Leicester, U.K. Before joining DMU, he was a Lecturer (Assistant Professor) with the School of Electronics and Information Engineering, HIT, China. His current research interests include video coding standards, data compression, deep learning, convolutional neural network, multimedia coding and transmission, and pattern recognition.

Frederic Dufaux is a CNRS Research Director at Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes (L2S, UMR 8506), where he is head of the Telecom and Networking research hub. He received his M.Sc. in physics and Ph.D. in electrical engineering from EPFL in 1990 and 1994 respectively. Frederic is a Fellow of IEEE. He was Vice General Chair of ICIP 2014, General Chair of MMSP 2018, and Technical Program co-Chair of ICIP 2019 and ICIP 2021. He served as Chair of the IEEE SPS Multimedia Signal Processing (MMSP) Technical Committee in 2018 and 2019. He was a member of the IEEE SPS Technical Directions Board from 2018 to 2021. He was Chair of the Steering Committee of ICME in 2022 and 2023. He is Chair-Elect of the IEEE CAS Multimedia Systems and Applications (MSA) Technical Committee (09/2023 08/2025), and then will serve as Chair (09/2025 08/2027). He was also a founding member and the Chair of the EURASIP Technical Area Committee on Visual Information Processing from 2015 to 2021. He was Editor-in-Chief of Signal Processing: *Image Communication* from 2010 until 2019. Since 2021, he is Specialty Chief Editor of the section on Image Processing in the journal *Frontiers in Signal Processing*. Frederic is also on the Executive Board of Systematic Paris-Region, a European competitiveness cluster which brings together and drives an ecosystem of excellence in digital technologies and DeepTech. He has been involved in the standardization of digital video and imaging technologies for more than 15 years, participating both in the MPEG and JPEG committees. He was co-chairman of JPEG 2000 over wireless (JPWL) and co-chairman of JPSearch. He is the recipient of two ISO awards for these contributions. His research interests include image and video coding, 3D video, high dynamic range imaging, visual quality assessment, video surveillance, privacy protection, image and video analysis, multimedia content search and retrieval, and video transmission over wireless network. He is author or co-author of 3 books, more than 200 research publications (h-index=50, 10000+ citations) and 20 patents issued or pending.

Hui Guo received the M.Sc. degree from the Guangxi Normal University, Guilin, China, in 2010. She is currently working toward the Ph.D. degree with the Macao University of Science and Technology, Macao, China. She is a Professor with Wuzhou University, Wuzhou, China. Her research interests include image/video processing and data visualization.

Ce Zhu (M'03-SM'04-F'17) received the B.S. degree from Sichuan University, Chengdu, China, in 1989, and the M.Eng and Ph.D. degrees from Southeast University, Nanjing, China, in 1992 and 1994, respectively, all in electronic and information engineering. He held a post-doctoral research position with the Chinese University of Hong Kong in 1995, the City University of Hong Kong, and the University of Melbourne, Australia, from 1996 to 1998. He was with Nanyang Technological University, Singapore, for 14 years from 1998 to 2012, where he was a Research Fellow, a Program Manager, an Assistant Professor, and then promoted to an Associate Professor in 2005. He has been with University of Electronic Science and Technology of China (UESTC), Chengdu, China, as a Professor since 2012, and serves as the Dean of Glasgow College, a joint school between the University of Glasgow, UK and UESTC, China. His research interests include video coding and communications, video analysis and processing, 3D video, visual perception and applications. He has served on the editorial boards of a few journals, including as an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON BROADCASTING , and IEEE SIGNAL PROCESSING LETTERS, an Editor of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and an Area Editor of *Signal Processing: Image Communication*. He has also served as a Guest Editor of a few special issues in international journals, including as a Guest Editor in the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING. He was an APSIPA Distinguished Lecturer (2021-2022), and also an IEEE Distinguished Lecturer of Circuits and Systems Society (2019-2020). He is serving as the Chair of IEEE ICME Steering Committee (2024-2025). He is a co-recipient of multiple paper awards at international conferences, including the most recent Best Demo Award in IEEE MMSP 2022, and the Best Paper Runner Up Award in IEEE ICME 2020.