



**HAL**  
open science

## Fast Mode and CU Splitting Decision for Intra Prediction in VVC SCC

Dayong Wang, Junyi Yu, Xin Lu, Frédéric Dufaux, Bo Hang, Hui Guo, Ce Zhu

► **To cite this version:**

Dayong Wang, Junyi Yu, Xin Lu, Frédéric Dufaux, Bo Hang, et al.. Fast Mode and CU Splitting Decision for Intra Prediction in VVC SCC. *IEEE Transactions on Broadcasting*, 2024, 70 (3), pp.872-883. 10.1109/TBC.2024.3394288 . hal-04572472

**HAL Id: hal-04572472**

**<https://hal.science/hal-04572472v1>**

Submitted on 10 May 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.

# Fast Mode and CU Splitting Decision for Intra Prediction in VVC SCC

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

**Abstract**—Currently, screen content video applications are increasingly widespread in our daily lives. The latest Screen Content Coding (SCC) standard, known as Versatile Video Coding (VVC) SCC, employs a quad-tree plus multi-type tree (QTMT) coding structure for Coding Unit (CU) partitioning and screen content Coding Modes (CMs) selection. While VVC SCC achieves high coding efficiency, its coding complexity poses a significant obstacle to the further widespread adoption of screen content video. Hence, it is crucial to enhance the coding speed of VVC SCC. In this paper, we propose a fast mode and splitting decision for Intra prediction in VVC SCC. Specifically, we initially exploit deep learning techniques to predict content types for all CUs. Subsequently, we examine CM distributions of different content types to predict candidate CMs for CUs. We then introduce early skip and early terminate CM decisions for different content types of CUs to further eliminate unlikely CMs. Finally, we develop Block-based Differential Pulse-Code Modulation (BDPCM) early termination and CU splitting early termination to improve coding speed. Experimental results demonstrate that the proposed algorithm improves coding speed on average by 41.14%, with the BDBR increasing by 1.17%.

**Index Terms**—VVC SCC, content type, fast coding mode decision, BDPCM, fast splitting decision.

## I. INTRODUCTION

**S**CREEN content video refers to recordings captured directly from the display screen of an electronic device. The surge in popularity of computer screen-sharing applications can be attributed to rapid advancements in networking and

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62020106011 and Grant 62162054; in part by the project of Ministry of Science and Technology, PRC under Grant G2023027007L; in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZDK202100604; in part by the Guizhou Provincial Basic Research Program (Natural Science) Qiankehejichu-ZK[2023]Yiban252; and in part by the Guangxi Key Laboratory of Machine Vision and Intelligent Control under Grant 2023B05. (*Corresponding authors: Bo Hang; Hui Guo.*)

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).

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

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 Systemes, Universite Paris-Saclay, CNRS, CentraleSupélec, 91192 Gif-sur-Yvette, France (e-mail: frederic.dufaux@l2s.centralesupelec.fr)

B. Hang is with the Computer School, Hubei University of Arts and Science, Xiangyang 441053, China (e-mail: bohng@163.com)

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).

portable devices in recent times. Such applications, including remote desktop access and video conferencing, are often complemented by document and slideshow sharing. These multimedia communication sessions typically comprise a blend of Camera-captured Content (CC) and Screen Content (SC). Additionally, numerous television programs and online videos also integrate both CC and SC elements. It is foreseeable that the future will witness the emergence of cloud services utilizing screen-sharing technologies [1]. Consequently, there is a growing demand for the efficient compression of SC. In response to this demand, a Call for Proposal (CfP) was issued in January 2014 [2] for the development of Screen Content Coding (SCC) [3] as an extension of the High Efficiency Video Coding (HEVC) standard [4]. This initiative has been undertaken by the Joint Collaborative Team on Video Coding (JCT-VC). The SCC extension introduced two major Coding Modes (CMs): Intra Block Copy (IBC) [5] and Palette (PLT) [6] modes, aimed at improving coding efficiency. In recent times, the Joint Video Experts Team (JVET) has spearheaded the development of the next-generation video coding standard, Versatile Video Coding (VVC) [7]. Early in its development, VVC incorporated IBC, and later, PLT was added in subsequent phases. It is anticipated that a wide range of SC applications utilizing VVC, with the integration of IBC and PLT capabilities, will emerge in the future. However, the coding process of VVC is extremely complex, primarily due to the introduction of novel coding tools in VVC SCC, such as the quad-tree plus multi-type tree (QTMT) coding structure for the Split Mode (SM) selection. Specifically, a Coding Tree Unit (CTU) undergoes an initial partition through a quadtree (QT) approach. Following this, the leaf nodes of the quadtree can undergo further partitioning using a multi-type tree (MT) structure. The multi-type tree structure comprises four SM types: vertical binary tree splitting (VB), horizontal binary tree splitting (HB), vertical ternary tree splitting (VT), and horizontal ternary tree splitting (HT). It's worth noting that, for the sake of simplifying the CTU partitioning process, once a Coding Unit (CU) undergoes partitioning through MT, the use of QT is no longer applicable in subsequent partitioning steps. The integration of MT significantly complicates the process of obtaining the best CTU structure. Furthermore, the introduction of IBC and PLT modes further intensifies the computational complexity. Consequently, the development of fast algorithms to mitigate the coding complexity of VVC SCC is highly desirable.

This paper introduces a fast Intra prediction algorithm aimed at enhancing the coding speed of VVC SCC. Our approach first leverages the deep learning techniques to categorize CUs

into distinct types, namely Animation (A) CUs, Camera-captured Content (CC) CUs, Text and Graphics with Motion (TGM) CUs, and Mixed content (Mixed) CUs. Based on this categorization, we predict candidate CMs and subsequently implement Block-based Differential Pulse-Code Modulation (BDPCM) early termination and CU splitting early termination to improve the coding speed. The main contributions of this paper are outlined as follows:

(1) A U-Net based network is developed for the categorization of CUs into four distinct types: A, CC, TGM, or Mixed.

(2) A strategy for CU early skip and early termination is introduced, based on content types and features. We begin with investigating CM distributions of different content types and predicting candidate CMs based on these distributions. Subsequently, we propose a strategy to early skip and terminate the CM decision process for CUs of various content types to further exclude unlikely CMs.

(3) An early termination approach for BDPCM is presented, based on content types and residual coefficients. This approach aims to exclude subsequent RDO processes for the Intra mode.

(4) An early termination approach for CU splitting is proposed, based on content types and features. This approach aims to determine whether the current CU needs further splitting, allowing for the early skipping of additional SM examinations.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents an overview of the proposed algorithm. Section IV is dedicated to the categorization methods for CU contents. Section V describes the CM prediction methods. Section VI discusses the early termination approach for BDPCM. The early termination strategy for CU splitting is presented in Section VII. Experimental results and conclusions are given in Sections VIII and IX, respectively. The source code is publicly available at <https://github.com/yujy95/FMS-SCC-TBC2024>.

## II. RELATED WORK

In order to accelerate the encoding process, several fast approaches have been proposed. Xu et al. [8], [9] reduced the encoding time by optimizing the quantization of HEVC. Wang et al. [10]–[12] employed inter-layer and spatial correlations to predict candidate modes and depths to improve coding speed of Quality Scalable High Efficiency Video Coding. Yang et al. [13] used a decision tree to predict candidate SMs, followed by the utilization of a gradient descent search method to predict likely directional modes. Li et al. [14] proposed a deep learning-based approach to predict candidate SMs, leading to a substantial reduction in encoding complexity of Intra mode in VVC. While these methods effectively reduce computational complexity of the VVC encoders, it is essential to note that they are specifically tailored for natural video contents. Given the significant distinctions between natural video content and screen video content, these methods are not suitable for VVC SCC.

Considering the scarcity of strategies available for accelerating the encoding process of VVC SCC and the existence of numerous fast SCC approaches for HEVC SCC, our primary

emphasis is on the discussion of fast algorithms for HEVC SCC. These algorithms can be broadly categorized into two groups: coding mode prediction and CU splitting prediction.

### A. Coding mode prediction

In general, CM prediction can be broadly categorized into two groups: content classification-based methods and feature classification-based methods. A detailed review of these methods is presented below.

**Content classification-based methods:** Due to the relationship between image content and CMs, candidate CMs can be derived based on their content. In [15], statistical features were selected and fed into a decision tree to effectively classify CUs into two categories: natural content CUs and screen content CUs, aiming to predict candidate CMs. In [16], factors such as color number and gradient were utilized to predict content types of CUs, thereby predicting candidate CMs. Gao et al. [17] employed a CNN to classify CUs into natural, text, image, and color content blocks. They then integrated CU content type, CU size, mode complexity, and spatial correlations between neighboring CUs and the current CU to predict candidate CMs. These methods predict candidate CMs based on content types. However, since the correlations between content types and CMs are not consistently robust, relying on content to predict candidate CMs may not always yield optimal performance.

**Feature classification-based methods:** Pan et al. [18] suggested early termination of the MERGE mode decision in HEVC to decrease computational complexity. Tsang et al. [19] predicted CU sizes by considering the RD costs and early skipped the IBC mode based on the hash value of each block. Kuang et al. [20] employed an online learning Bayesian decision rule to skip unlikely CMs early. Additionally, Kuang et al. [21] first designed various features as eigenvalues and then employed a decision tree to sequentially determine whether to examine Intra mode, IBC Merge mode, IBC search mode, and PLT mode. Tsang et al. [22] selected a more extensive set of statistical features to construct a random forest to determine whether to examine Intra mode, IBC Merge mode, IBC Search mode, and PLT mode. Kuang et al. [23] first chose statistical features and then used an online-learning-based Bayesian decision rule to determine whether to early skip unlikely CMs. Subsequently, they utilized the mode information from neighboring CUs to further eliminate unlikely CMs for the current CU. Kuang et al. [24] developed a fast prediction network based on deep learning to obtain probabilities for all CMs and to early skip CMs of low probabilities, thus improving coding speed. Tsang et al. [25] used a CNN-based classifier to predict candidate CMs for all 4×4 CUs and then determined the CMs based on the counterparts of their 4×4 sub-CUs. Wang et al. [26]–[28] developed early terminations for the Inter-layer mode to enhance coding speed. The aforementioned methods predict candidate CMs based on features. However, as the correlations between features and CMs are not consistently robust, utilizing features for predicting candidate CMs may not always yield optimal performance.

## B. CU Splitting Prediction

In general, the prediction of CU splitting is commonly used to improve coding speed. Studies in [29] and [30] emphasize the crucial importance of a fast CU partitioning decision in video encoding processes. Tsang et al. [19] utilized characteristics with pixel exactness in both horizontal and vertical directions, along with the rate-distortion costs, to facilitate early CU splitting and pruning. Kuang et al. [23] used an online-learning-based Bayesian decision rule to predict likely CU splitting and terminate unlikely CU splitting. Furthermore, Kuang et al. [31] investigated the statistical relationship between coding cost and CU partition, thus achieving an early termination of CU splitting.

As previously mentioned, both content classification-based methods and feature classification-based methods exhibit drawbacks. In addition, SMs in VVC are significantly more complex than those in HEVC. To address these issues, we employ a combined approach that utilizes both methods to predict candidate CMs. Specifically, we employ deep learning techniques to classify CUs into one of four content types. Subsequently, we analyze CM distributions for different content types to predict candidate CMs. We then introduce early skipping and early termination mechanisms for CM selection to exclude unlikely CMs. Furthermore, we develop a BDPCM early termination strategy based on content types and residual coefficients to bypass subsequent RDO processes for Intra mode. Lastly, we propose an early termination approach for CU splitting based on content types and features to exclude unlikely CU splitting, thereby improving coding speed.

## III. OVERVIEW OF THE PROPOSED FRAMEWORK

In order to enhance the coding speed of Intra prediction in VVC SCC, we propose a fast CM and CU splitting decision algorithm primarily consisting of four strategies: Deep Learning-Based Content Classification (DLBCC), Content and Feature-Based CM Prediction (CFBCMP), Content and Residual Coefficient-Based BDPCM Early Termination (CRCBBET), and Feature-Based CU Splitting Early Termination (FBCSET).

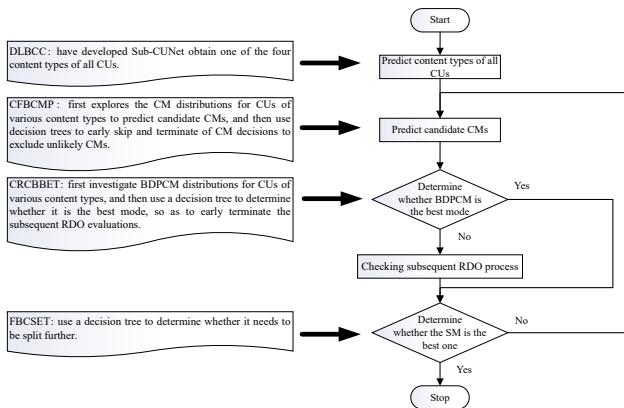


Fig. 1. Overall structure of the proposed algorithm.

In particular, DLBCC incorporates Sub-CUNet to categorize CUs into four content types. CFBCMP first analyzes CM

distributions across various content types to predict candidate CMs, and subsequently uses decision trees for early exclusion of unlikely CMs. CRCBBET investigates BDPCM distributions for CUs of various content types, and then uses a decision tree to determine whether it is the best mode, thus early terminating the subsequent RDO evaluations. After examining the current CU, FBCSET uses a decision tree to determine whether it needs to be split further.

The overall structure of the proposed algorithm is illustrated in Fig. 1, with the four strategies presented on the left-hand side.

## IV. DEEP LEARNING-BASED CONTENT CLASSIFICATION (DLBCC)

SCC sequences comprise four types of content: A, CC, TGM, and Mixed. Typically, each content type exhibits a strong correlation with specific CMs and CU splitting that are most likely to be selected for coding. Consequently, the first step of the proposed algorithm involves categorizing CU contents, followed by predicting candidate CMs and CU splitting based on their corresponding CU contents. To categorize CU contents, we designed a neural network and subsequently trained its associated model.

### A. Design of the Neural Network

In VVC, a CTU has a default size of  $128 \times 128$  and can be flexibly partitioned into blocks with a minimum size of  $4 \times 4$ . The coding structure of QTMT allows the partitioning of a CTU into CUs of up to 17 different sizes [14]. Employing deep learning techniques directly to predict candidate CMs for all CUs makes the computational process unbearably complex. Given that the minimum size of CUs is  $4 \times 4$ , we can first determine the content types of  $4 \times 4$  CUs and subsequently predict the content type of each CU based on its  $4 \times 4$  sub-CUs in the lower partition layer. If all  $4 \times 4$  sub-CUs within a CU exhibit the same type, the CU is categorized as that specific type. Conversely, if the  $4 \times 4$  sub-CUs within a CU differ in type, the CU is classified as a Mixed type. Determining the type of each  $4 \times 4$  CU is a challenging task.

Deep learning has demonstrated the capability to automatically extract extensive features, yielding exceptional performance across various applications. The Fully Convolutional Network (FCN) has proven to be highly effective, particularly for various pixel-wise segmentation tasks such as semantic segmentation, instance segmentation, and biomedical image segmentation. Among these FCNs, the U-Net stands out as a well-known choice for biomedical image segmentation [24] [25]. Inspired by U-Net, we have developed our own  $4 \times 4$  sub-CU segmentation net, denoted as Sub-CUNet. Unlike traditional pixel-wise segmentation, our Sub-CUNet produces only  $4 \times 4$  sub-CU-wise labels, as illustrated in Fig. 2 Structure of Sub-CUNet figure.2.

The luminance component of a  $128 \times 128$  CTU serves as the input to Sub-CUNet. Sub-CUNet generates 1024 labels corresponding to 1024  $4 \times 4$  sub-CUs, with each label representing the content most likely to be chosen. Sub-CUNet is composed of seven convolutional layers (CLs), five deconvolutional

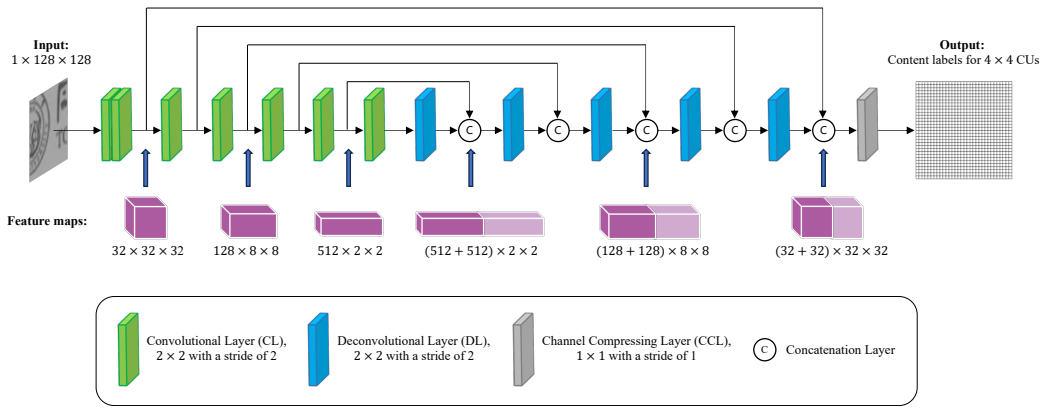


Fig. 2. Structure of Sub-CUNet.

layers (DLs), a channel compressing layer (CCL), and five concatenation layers. Following each CL or DL, a Rectified Linear Unit (ReLU) activation function is applied. However, in the case of CCL, the softmax activation function is employed to generate the output labels. The specific characteristics of these layers are detailed as follows.

**Convolutional Layers:** The luminance component of a CTU is processed by seven CLs to generate the corresponding feature maps. As shown in Fig. 2 Structure of Sub-CUNet figure.2, all CLs use a  $2 \times 2$  kernel size with a stride of 2. The strides for all CLs are consistently set to the width of the kernel sizes for non-overlapping convolutions. This design ensures that the receptive field of each node within a feature map consistently corresponds to the size of each square CU. Consequently, feature maps generated by all CLs capture the local characteristics of CUs with sizes ranging from  $4 \times 4$  to  $128 \times 128$ , respectively. With each convolutional step, the size of feature maps is halved, and the number of dimensions in the feature maps is doubled. After concatenating the feature maps from both the CLs and DLs, the CCL, which employs a kernel size of  $1 \times 1$  with a stride of 1, integrates these feature maps to generate the final set of feature maps. Ultimately, the softmax function is applied to these feature maps to generate labels for the  $4 \times 4$  CUs.

**Deconvolutional Layers:** In contrast to their convolutional counterparts, which double the dimensions and halve the sizes of feature maps, DLs are used to halve the dimensions and double the sizes of feature maps. After producing 1024 feature maps from the last CL, we employ five DLs, with a  $2 \times 2$  kernel size and a stride of 2, to enlarge the feature maps. Given that the receptive field of every node in the feature maps of the last CL is the entire CTU, the receptive field of each node in the feature maps of all DLs captures global features for CUs of sizes spanning from  $64 \times 64$  down to  $4 \times 4$ . These global features offer a substantial boost to prediction accuracy by leveraging the inherent spatial content correlation within a CTU. At each stage of enlarging the feature maps, we deliberately reduce the number of dimensions by half. Ultimately, this meticulous process ensures that both global and local feature maps maintain consistent dimensions for each CU size, thereby enhancing the overall performance of the system.

Through the Sub-CUNet, a  $128 \times 128$  CTU generates 1024 labels, each corresponding to a  $4 \times 4$  sub-block. Each  $4 \times 4$  sub-block is categorized into one of the four content types according to the probability. If all  $4 \times 4$  sub-blocks within a CU share the same content type, the CU is classified as that specific content type. Otherwise, the current CU is classified as Mixed.

### B. Training of the Neural Network

To avoid any overlap between the training set and the test set, we selected training sequences that are not included in the Common Test Conditions (CTC) [32] to generate training samples. These sequences cover A, CC, and TGM content, specifically including ClearTypeSpreadsheet, KristenAndSaraScreen, MissionControlClip1, ParkScene, PcbLayout, PeopleInVehicle, PptDocXls, RealTimeData, Seeking, VideoConferencingDocSharing, WordEditing, DOTA2, CSGO, WITCHER3, GTAV, and Hearthstone. In addition to those sequences recommended in [33] [34], we also obtained some sequences from the internet.

To fully leverage computing resources, the batch size is set to 4096. Sequences are randomly allocated at a ratio of 8:2 to the training set and the validation set. After loading the dataset, pixel values are normalized. In the training stage, PyTorch is utilized on an RTX3090 GPU. The cross-entropy loss function is selected, and the optimizer used is Adam, with default values of 0.9 for momentum 1 and 0.999 for momentum 2. The learning rate is set to 0.0001, and the epoch is set to 1000. The loss during the training process is illustrated in Fig. 3 Loss during the training process of the content classifier figure.3, and the best model is determined based on the decreasing trend of the loss value during training.

From Fig. 3 Loss during the training process of the content classifier figure.3, it is evident that as the number of epochs increases, both training and validation loss values gradually decrease and converge to similar values. When the number of epochs is greater than or equal to 800, both loss values reach a minimum and maintain stability with some slight fluctuations. This observation indicates that the networks are not experiencing overfitting problems, which typically occur when training losses are very small but validation losses are very high.

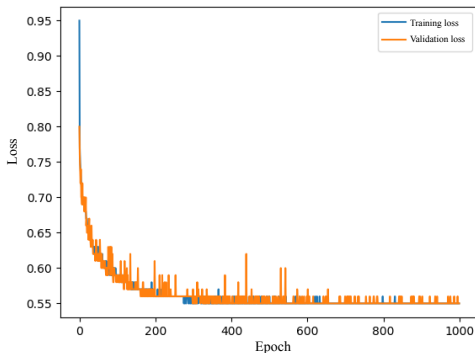


Fig. 3. Loss during the training process of the content classifier.

## V. CONTENT AND FEATURE-BASED CM PREDICTION (CFBCMP)

Through DLBCC, we can obtain information about the types of CUs. Subsequently, we examine CM distributions for CUs of various types and then identify likely CMs while excluding unlikely ones based on the obtained distributions.

To obtain the CM distribution for CUs of various types, extensive experiments were conducted. In our experiments, one sequence from each type was selected to cover different types. Specifically, the test sequences outlined in [24] were utilized. According to the CTC, all sequences were encoded using QP values of 22, 27, 32, and 37 under the All\_Intra configuration. All the training and testing sequences use a color format of YUV 444 with an 8-bits color depth. The corresponding CM distribution for CUs of various types is detailed in Table ICM distributions for CUs of various content typestable.1.

TABLE I  
CM DISTRIBUTIONS FOR CUs OF VARIOUS CONTENT TYPES

Categories	Intra (%)	PLT (%)	IBC (%)	Proportion(%)
A	93.14	1.51	5.34	22.32
CC	98.24	0.15	1.61	13.19
Mixed	59.1	12.6	28.3	1.38
TGM	33.38	26.37	40.26	63.11

In Table ICM distributions for CUs of various content typestable.1, Intra, PLT, and IBC refer to their corresponding proportions in A, CC, Mixed, and TGM CUs, Proportion refers to the proportions of A, CC, Mixed, and TGM CUs. From Table ICM distributions for CUs of various content typestable.1, it is evident that the CM distribution for CUs of various types exhibits significant differences. Accordingly, we propose separate mode selection schemes as follows:

(1) Considering that 98.24% of CC CUs employ the Intra mode, CUs in CC predominantly utilize the Intra mode. Consequently, CUs in CC exclusively examine the Intra mode.

(2) Since the proportion of Mixed CUs is only 1.38%, given such a small proportion, we directly examine all three CMs for Mixed CUs.

(3) Since 93.14% of A CUs employ the Intra mode, the proportions of the other two CMs are not considered negligible. We first examine the Intra mode and determine whether it is

the best. If affirmative, there is no need to examine the other two CMs; otherwise, further evaluation of the other two CMs is necessary.

(4) Since the proportional differences of TGM CUs among all three CMs are not substantial, it is not feasible to directly predict the candidate CMs based on their CM distributions. Instead, we use features to predict the candidate CMs. Specifically, for each CM, we first determine whether the mode can be skipped early. If affirmative, it is not needed to check the mode; otherwise, we use the mode to check and determine whether it is the best CM, thus allowing early termination of subsequent CM evaluations.

Currently, deep learning is extensively employed in video coding. Nevertheless, it often fails to leverage certain Features Obtained During Encoding (FODE), such as residual coefficients and RD costs, which are only accessible during the encoding process. These FODE may have a significant impact on CM selection. Consequently, utilizing deep learning may not always yield optimal performance. Conversely, machine learning classifiers can effectively utilize these FODE and achieve excellent performance. Therefore, we employ machine learning to predict candidate CMs.

In our experimental setup, we considered a range of machine learning classifiers, such as the Naive Bayes Classifier (NBC), Decision Tree (DT), and Support Vector Machine (SVM). NBC is a probabilistic machine learning model based on Bayes' theorem, which assumes conditional independence between features. NBC requires the statistical estimation of the probability of each feature first, which can be quite burdensome. SVM is primarily used for classification and regression analysis. In classification problems, SVM aims to find a hyperplane to maximize the spacing between different categories, which usually leads to complex training and prediction processes. DT is a widely used binary classification approach known for its comprehensible flow-chart-like structure. Each non-leaf node performs a test on a specific attribute, with branches indicating the test outcomes and leaf nodes holding class labels. Due to its simple prediction process and high prediction accuracy, we have chosen DT as the classifier in this paper. The decision tree is constructed using "Scikit-learn" in Python with offline training. The aforementioned sequences used for DLBCC are selected as training sequences.

As mentioned above, we only need to predict candidate CMs for CUs in A and TGM. As shown in Table ICM distributions for CUs of various content typestable.1, for CUs in both A and TGM, the Intra mode is frequently used. Moreover, these CMs are strongly related to textural features and can be directly predicted based on them. Therefore, it is advisable to first check the Intra mode. Since the coding process of the PLT mode is significantly simpler than that of the IBC mode, which involves IBC merge and IBC search, we next check the PLT mode and finally the IBC mode.

For the convenience of subsequent descriptions, features obtained directly before encoding are denoted as Direct Features (DFs), and features obtained during encoding are denoted as Coding Features (CFs). To enhance coding speed, before examining a CM, we can use some DFs to determine whether the CM can be skipped early. If affirmative, the CM can

be directly skipped. Otherwise, we examine it and combine DFs with CFs to determine whether the CM is the best one, thus early terminating the CM selection. We denote the aforementioned two processes as (a) Mode Early Skipping based on DFs and (b) Early Termination Based on DFs and CFs. The details are as follows:

#### A. Mode Early Skipping based on DFs

Before examining a CM, certain DFs can be acquired. We input these DFs into a decision tree to determine the confidence level of utilizing the CM. To accurately obtain the confidence level, the DFs are selected as follows.

A CTU can be partitioned into CUs of up to 17 different sizes using the QTMT structure. By applying the same experimental conditions as outlined at the beginning of Section V, we derived their CM distributions, which are presented in Table IICM distributions for CUs of various sizes.

TABLE II  
CM DISTRIBUTIONS FOR CUs OF VARIOUS SIZES

CU sizes	Intra (%)	PLT (%)	IBC (%)
64×64	37.32	61.77	0.92
32×32	49.6	37.96	12.44
32×16	61	22.3	16.7
32×8	51.53	18.07	30.4
32×4	48.07	13.19	38.74

From Table IICM distributions for CUs of various sizes, we can observe that the CM distribution for CUs of different sizes is significantly different. Notably, substantial variations in CM distributions are also observed for CU sizes not included in Table IICM distributions for CUs of various sizes. Clearly, the choice of the best CM is closely tied to the size of a CU. As a result, the size of CUs is chosen as a feature.

In general, the texture of a CU plays a significant role in the decision-making process for CM selection. Pixel variances are evidently crucial textural features in this context. Consequently, we utilize overall variances (OVP), as well as horizontal (HVP) and vertical (VVP) variances of pixels within a CU as features. These features can be calculated as follows:

$$OVP = \frac{\sum_{i=0}^W \sum_{j=0}^H (p_{ij} - \bar{p})^2}{W \times H}, \quad (1)$$

$$HVP = \frac{\sum_{i=0}^W \sum_{j=0}^H (p_{ij} - \bar{p}_i)^2}{W \times H}, \quad (2)$$

$$VVP = \frac{\sum_{i=0}^W \sum_{j=0}^H (p_{ji} - \bar{p}_j)^2}{W \times H}, \quad (3)$$

where  $W$  and  $H$  represent the width and height of a CU, respectively,  $p_{ij}$  denotes the pixel value at the position  $(i, j)$  in the CU,  $\bar{p}$  is the average pixel value of the CU,  $\bar{p}_i$  is the average pixel value of the  $i$  th row in the CU, and  $\bar{p}_j$  represents the average pixel value of the  $j$  th column of the CU.

The Number of Background Color (NBC) is a metric representing the quantity of pixels that share the same color,

with the highest frequency of occurrence within a given CU. The Number of Distinct Color (NDC) measures the pixels in a CU with unique pixel values, essentially pixels with colors distinct from others. Studies in [21], [22] and [35] demonstrate that the CM selection of PLT is strongly related to the count of colors. This is caused by the fact that the count determines the number of palette items that must be encoded by PLT following clustering. After clustering, the pixel values of the block are represented as indices in the palette, and these indices are encoded using Run-Length Coding. Only one item is required for consecutive identical colors, referred to as a color segment in this context. Consequently, the minimum Number of Color Segments (NCS) observed along both horizontal and vertical scanning paths directly influences the number of index items that PLT needs to encode and significantly impacts the resulting PLT bit rate. Among these features, CU sizes, OVP, HVP, VVP, and NCS have been proposed as novel features.

Table III Features and their respective IGAE values for mode early skipping presents the aforementioned features along with their respective Information Gain Attribute Evaluation (IGAE) values for CUs in A and TGM categories. The Information Gain is calculated by assessing the entropy difference before and after classification [36].

TABLE III  
FEATURES AND THEIR RESPECTIVE IGAE VALUES FOR MODE EARLY SKIPPING

Categories	CMs	IGAE						
		Size	OVP	HVP	VVP	NDC	NCS	
A	Intra	0.88	2.39	6.61	55.56	22.66	11.89	—
	PLT	2.40	57.44	—	—	2.45	32.54	5.18
	IBC	6.11	65.11	—	—	21.57	7.11	—
TGM	Intra	1.82	2.99	13.30	43.60	24.09	14.19	—
	PLT	4.63	35.43	—	—	1.82	43.64	14.50
	IBC	28.72	42.33	—	—	21.14	7.71	—

In Table III Features and their respective IGAE values for mode early skipping, some features have little relevance to the likelihood of choosing specific CMs. As a result, these features are not used and are labeled as "\".

Feeding the features of a CM-to-examine into a decision tree enables us to derive a confidence level for this CM. If the confidence level is less than or equal to a specified threshold, we can terminate CM selection early. To determine the optimal threshold, we used VTM-17.0 [37] for testing on a server equipped with an Intel Xeon Gold 5122 CPU and 64GB of memory. The aforementioned training sequences are encoded under the All\_Intra configuration using QP values of 22, 27, 32, and 37. Coding efficiency is measured using BDBR [38]. The confidence levels and their respective BDBRs for early skipping of all three CMs in the TGM category are illustrated in Fig. 4 Confidence level and the respective BDBR for CUs in the TGM category.

From Fig. 4 Confidence level and the respective BDBR for CUs in the TGM category, it is evident that when the confidence levels of both Intra mode and IBC mode are less than or equal to 0.2, the corresponding BDBR remains nearly the same. However, when the confidence levels of both these two CMs exceed 0.2, the corresponding BDBR dramatically

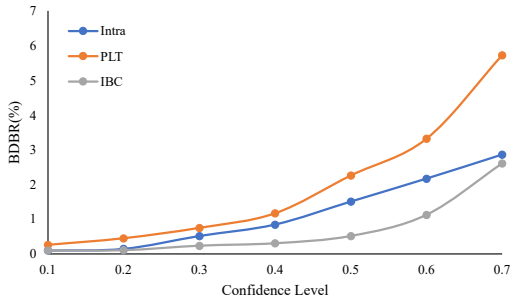


Fig. 4. Confidence level and the respective BDBR for CUs in the TGM category.

increases. Clearly, there is a turning point at a confidence level of 0.2. Therefore, we set the threshold as 0.2 for both Intra mode and IBC mode. As for the PLT mode, when its confidence level is greater than 0.1, the corresponding BDBR continuously increases. Consequently, we set the threshold as 0.1 for this CM. Employing a similar approach, we can determine thresholds for all CMs for CUs in the A category. Through the aforementioned process, we derive the thresholds for all CMs for CUs in the TGM and A categories, which are listed in Table IV. Thresholds for all CMs for CUs in the TGM and A categories are as follows:

TABLE IV  
THRESHOLDS FOR ALL CMs FOR CUs IN THE TGM AND A CATEGORIES

Categories	CMs	Thresholds
TGM	Intra	0.2
	PLT	0.1
	IBC	0.2
A	Intra	0.2
	PLT	0.5
	IBC	0.2

When the confidence level of a CM for CUs in the TGM and A categories is less than or equal to the corresponding threshold in Table IV. Thresholds for all CMs for CUs in the TGM and A categories are as follows, this CM can be skipped directly, thereby improving the coding speed.

### B. Early Termination Based on DFs and CFs

As mentioned earlier, if the confidence level of a CM exceeds the corresponding threshold in Table IV. Thresholds for all CMs for CUs in the TGM and A categories are as follows, further examination of the CM becomes necessary. After examining the CM, specific coding information, including residual coefficients and RD costs, is obtained. This coding information plays a crucial role in the CM early termination. If the residual coefficients and RD costs are effectively insignificant, it is highly likely that the corresponding CM is the best one. Consequently, CM selection can be terminated early. Therefore, considering both textural features and coding information, we can determine whether the mode-to-examine is the best one, thus early terminating the CM selection.

After examining a CM, the acquired textural features and coding information are inputted into a decision tree to de-

termine the corresponding confidence level. To achieve the optimal confidence level for early termination of the CM, we apply the aforementioned experimental conditions during testing. The experiments demonstrate that, for all three CMs, as the confidence level increases, BDBR significantly decreases. When the confidence level is greater than or equal to 0.9, BDBR remains nearly constant. Clearly, there is a turning point when the confidence level equals 0.9. Therefore, we set the threshold as 0.9 for the early termination of all three CMs.

## VI. CONTENT AND RESIDUAL COEFFICIENT-BASED BDPCM EARLY TERMINATION (CRCBBET)

In VVC, the Intra prediction process primarily involves three steps: Rough Mode Decision (RMD), BDPCM, and Rate-Distortion Optimization (RDO). RMD uses the Hadamard transform to establish a Rate-Distortion Mode List (RDL). BDPCM employs only horizontal and vertical modes for prediction, directly quantifying their residual coefficients without using a transformation. Subsequently, the RDO process uses the DCT transform to examine the RDL, Matrix Weighted Intra Prediction (MIP), and Intra Sub-Partition (ISP) to determine the best mode. Given the nature of BDPCM, as mentioned above, its coding complexity is clearly insignificant. Once BDPCM has been examined, if it is identified as the best mode, subsequent coding processes can be early terminated, thereby significantly improving coding speed. To determine whether BDPCM is the best mode, we examine both the percentage of cases in which BDPCM is selected as the best mode and the percentage of cases in which RDO determines the best mode for CUs in the TGM and A categories under the aforementioned experimental conditions. The results are shown in Table V. Comparison of BDPCM and RDO for CUs in TGM and A categories are as follows.

TABLE V  
COMPARISON OF BDPCM AND RDO FOR CUs IN TGM AND A CATEGORIES

Categories	BDPCM (%)	RDO (%)
A	2.50	97.50
TGM	38.85	61.15

Table V. Comparison of BDPCM and RDO for CUs in TGM and A categories are as follows shows the proportions of BDPCM in the A and TGM categories, which are 2.50% and 38.85%, respectively. Given the relatively small proportion of BDPCM in the A category, our focus shifts to determining whether BDPCM is the best mode for CUs in the TGM category.

The performance of BDPCM can be effectively reflected by the residual coefficients. When examining a CU using BDPCM, if the values of its residual coefficients are small and evenly distributed, this indicates that the CU is predicted very well. BDPCM is thus highly likely to be the best mode, and the subsequent RDO processes can be terminated early. As the variance of residual coefficients can accurately measure their complexity, we select the overall variance (OVRB), horizontal variance (HVRB), and vertical variance (VVRB) of residual coefficients as features. Additionally, if the proportion of zero values in the quantized residual coefficients is notably high,



it also indicates that the CU is predicted very well, making BDPCM likely to be the best mode. Therefore, the proportion of zero values in the quantized residual coefficients, denoted as PZQR, is also chosen as a feature. It is noteworthy that the RD cost of BDPCM reflects its performance well and is included as a feature. Since this is the first paper investigating BDPCM early termination, all its features are novel. Based on the aforementioned analysis, the selected features and their respective Information Gain of Attribute Evaluation (IGAE) values are listed in Table VI. Features and their respective IGAE values for BDPCM early termination are shown in Table 6.

TABLE VI  
FEATURES AND THEIR RESPECTIVE IGAE VALUES FOR BDPCM EARLY TERMINATION

Features	IGAE
OVRB	6.90
HVRB	1.31
VVRB	1.24
POZV	6.64
RD cost	84.05

Feeding these features into a decision tree allows us to determine the confidence level of utilizing BDPCM as the best mode. Clearly, confidence levels are closely linked to BDPCM selection. If the confidence level is greater than or equal to a predefined threshold, BDPCM can be chosen as the best mode, leading to the early termination of subsequent coding processes. To identify the optimal threshold, the aforementioned experimental conditions and sequences are employed in testing. Theoretically, only when thresholds are greater than or equal to 0.5 can BDPCM be selected as the best mode. Therefore, we exclusively use thresholds greater than or equal to 0.5 for testing purposes. The corresponding thresholds and their associated BDBRs are illustrated in Fig. 5. Thresholds and their respective BDBR for CUs in the TGM category are shown in Fig. 5.

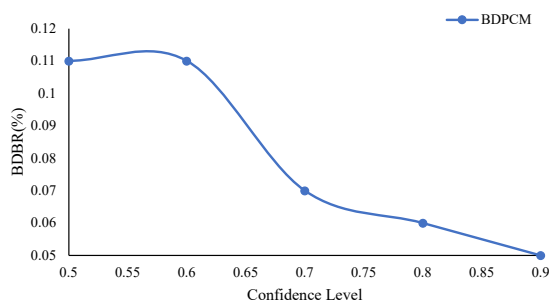


Fig. 5. Thresholds and their respective BDBR for CUs in the TGM category.

From Fig. 5, it is evident that as the thresholds increase, the corresponding BDBR gradually decreases. This phenomenon can be explained by the fact that, as confidence levels increase, the number of mispredicted CUs also decreases gradually. However, with smaller confidence levels, more CUs meet the conditions, leading to a greater improvement in coding speed. A tradeoff exists between coding speed and coding efficiency. Considering both factors, we choose a threshold of 0.5. When a CU is examined by BDPCM, if its

confidence level is greater than or equal to 0.5, BDPCM is selected as the best mode, and subsequent RDO processes can be terminated early.

## VII. FEATURE-BASED CU SPLITTING EARLY TERMINATION (FBCSET)

As previously mentioned, VVC SCC employs the QTMT coding structure for CU partitioning. A CTU can be partitioned into CUs of up to 17 different sizes, and each CU can support a maximum of six SMs: non-splitting (NS), HB, HT, VB, VT, and QT. This results in a substantial number of SMs and a highly complex coding process. Although each CU size can have up to six SMs, their proportions of use vary significantly. Notably, NS constitutes a significant portion. Furthermore, the smaller the size of the CU, the higher the proportions of NS. After examining the NS mode for a CU, if we can determine that it does not need to be split, subsequent coding processes can be terminated early, and coding speed can be significantly improved.

To accurately determine whether a CU needs to be split further, the key lies in selecting the best features. In general, the higher the QP, the less likely CUs are to be split, namely, they are more likely to select the NS mode, and vice versa. Additionally, both quad-tree depth (QD) and multi-type tree depth (MD) play important roles in the decision of splitting CUs [39], [40]. Therefore, QP, QD, and MD are chosen as features.

After examining the NS mode for a CU, we can acquire valuable CF that directly reflects its coding performance and is closely related to CU splitting. Therefore, the following coding information has been chosen as features:

(1) ISP mode: This mode is introduced to subdivide an intra-predicted block into multiple subpartitions. Therefore, in areas where pixels within the same subpartition are very similar but are significantly different in different subpartitions, ISP mode is often chosen as the best coding mode. It is observed that CUs using ISP mode are more likely to be further split.

(2) Coding information of the Best CM (BCM): After examining all three CMs for a CU, the CM with the smallest RD cost is chosen as the BCM. The coding information of the BCM is closely related to CU splitting. It is evident that the smaller the variance of residual coefficients and the RD cost of the BCM, the higher the probability that a CU will not be split, and vice versa. Therefore, we can leverage the coding information of BCM, including its residual coefficients and RD cost, to determine whether the CU needs to be split.

Based on the above analysis, the Flag of ISP (FISP), BCM, and the RD cost, as well as the overall, horizontal, and vertical variances of residual coefficients (denoted as OVRB, HVRB, and VVRB, respectively), are all selected as features of a CU. Since coding information from the BCM is never utilized in prediction, all these features are novel. Specifically, BCM, RCBM, OVRB, HVRB, and VVRB are all novel features. These features, along with their IGAE values for CU splitting early termination, are listed in Table VII. Features and their respective IGAE values for CU splitting early termination are shown in Table 7.

Feeding these features into a decision tree enables us to determine the corresponding confidence level. If the con-

TABLE VII  
FEATURES AND THEIR RESPECTIVE IGAE VALUES FOR CU SPLITTING  
EARLY TERMINATION

Categories	Features	IGAE(A)	IGAE(CC)
DF	QP	0.14	0.28
	QD	0.44	0.55
	MD	23.60	30.31
CF	FISP	1.12	1.55
	BCM	0.54	0.55
	OVBC	1.50	4.50
	HVBC	0.27	0.27
	VVBC	0.39	0.46
	RCBC	72.37	61.56

confidence level is greater than or equal to a threshold, the corresponding CU does not need to be split. To obtain the optimal threshold, we utilize the aforementioned experimental conditions and sequences in testing. The thresholds and their respective BDBRs for CUs in the A category are illustrated in Fig. 6. Thresholds and their respective BDBR for CUs in the A category figure.6.

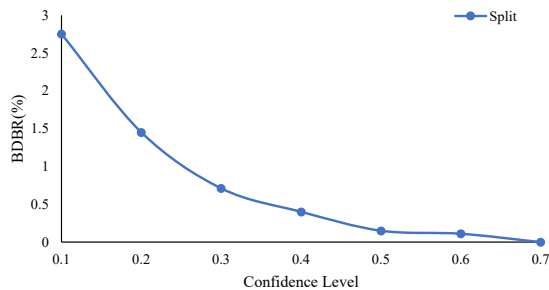


Fig. 6. Thresholds and their respective BDBR for CUs in the A category.

From Fig. 6. Thresholds and their respective BDBR for CUs in the A category figure.6, it is evident that as confidence levels increase, the corresponding BDBR gradually decreases. Additionally, when confidence levels are smaller than or equal to 0.5, BDBR decreases significantly. Otherwise, BDBR decreases steadily and slowly. Clearly, there is a turning point at 0.5. Therefore, we choose 0.5 as the threshold for CUs in the A category. After examining the NS mode for a CU, if its confidence level is greater than or equal to 0.5, the CU does not need to be split. Similarly, we establish the threshold for CUs in the CC category, which is also set to 0.5.

The texture of TGM CUs is relatively simple, mostly encoded at small depths, and has similar features. For example, a large number of color blocks in TGM CUs do not require further splitting, and the existing optimization algorithms in VTM are able to accurately terminate the coding process for these types of CUs. Therefore, FBCSET is not suitable for TGM CUs.

### VIII. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, we employed the VTM-17.0 reference software for testing on a server equipped with an Intel Xeon Gold 5122 CPU and 64GB of memory. The experimental parameters were configured in

accordance with the CTC. Specifically, both GOPSize and IntraPeriod were set to 1, and QPs were set as (22, 27, 32, 37). Due to the unavailability of certain video sequences, only some sequences of the SC included in the CTC are considered. The coding efficiency was evaluated using BDBR, representing the percentage of bitrate increase for the same quality compared to the reference software. Positive and negative values indicate a loss and an improvement in coding efficiency, respectively. Computational complexity was measured by the percentage of the coding time savings, denoted as  $\Delta T$ .

The proposed algorithm integrates four strategies: DLBCC, CFBCMP, CRCBBET, and FBCSET. DLBCC is specifically designed for classifying content types of CUs, and although it does not directly improve coding speed, it aids the other three strategies in doing so. Therefore, it is necessary to present only the performance of the remaining three strategies in Table VIII. Performance of the three strategies table.8.

From Table VIII. Performance of the three strategies table.8, it is evident that CFBCMP, CRCBBET, and FBCSET exhibit average coding speed improvements of 29.65%, 7.38%, and 11.02%, respectively. The corresponding average BDBRs increase by 0.77%, 0.10%, and 0.15%, respectively. Additionally, it can be observed that CFBCMP enhances coding speed for all sequences. Since the proportions of BDPCM for sequences in the A and CC categories are very small, CRCBBET cannot improve their coding speed. Furthermore, FBCSET significantly improves coding speed for CUs in both A and CC categories, but it achieves only insignificant or minimal improvement in coding speed for some Mixed and TGM sequences. The reason is that, as aforementioned, FBCSET is not applied to TGM CUs but only applied for A and CC CUs. Since the proportions of A and CC CUs in many Mixed and TGM sequences are not very high, their coding speed improvements for FBCSET are also not very significant accordingly. However, sequence SlideShow may include a high proportion of A and CC CUs, FBCSET achieves remarkable coding speed improvement for this sequence. Different CU content types exhibit significantly different characteristics, which seriously influence coding speed improvements. Therefore, we need to classify CU content types and then develop their corresponding methods to improve coding speed.

Considering the limited research on VVC SCC and the abundance of works on HEVC SCC, we compare our overall performance with MLMD [21], and OLBD [23], FSC-NET [25]. It is important to note that we did not compare with the latest two papers, namely Gao et al.'s algorithm [17] and Zuo et al.'s algorithm [41]. The rationale behind this decision is that Gao et al.'s algorithm assigns different CMs to CUs of various sizes in HEVC, which encompasses only four CUs of different sizes. In contrast, VVC supports CUs of 17 different sizes. Consequently, Gao's method is not suitable for a direct comparison with our proposed algorithm. Incorporating Gao et al.'s algorithm into VVC would result in a very limited time reduction, rendering it an unfair basis for comparison. The coding speed of Zuo et al.'s algorithm has increased by 90.51%, but this improvement comes at the cost of a 9.47% increase in BDBR. Despite the notable improvement in coding speed, the coding efficiency of Zuo et

TABLE VIII  
PERFORMANCE OF THE THREE STRATEGIES

Categories	Sequences	CFBCMP		CRCBBET		FBCSET	
		BDBR (%)	$\Delta T$ (%)	BDBR (%)	$\Delta T$ (%)	BDBR (%)	$\Delta T$ (%)
A	Robot	0.92	24.17	-0.06	-5.63	0.44	19.54
CC	EBURainFruits	0.00	33.07	-0.11	-6.4	0.1	28.62
	Kimono1	1.07	31.12	0.18	-4.96	0.53	42.16
Mixed	BasketballScreen	0.84	29.49	0.16	9.03	0.19	6.47
	MissionControlClip2	0.49	24.64	0.18	12.72	0.07	9.98
	MissionControlClip3	0.52	29.69	0.05	10.68	0.08	5.6
TGM	Console	0.32	35.91	0.11	15.23	-0.03	2.03
	Desktop	0.08	46.09	0.02	19.56	-0.14	4.4
	FlyingGraphics	2.88	30.90	0.23	12.59	0.18	3.89
	Map	1.03	18.86	0.15	1.74	0.07	-0.76
	Programming	0.87	25.15	0.3	12.54	-0.01	2.5
	SlideShow	0.67	16.94	0.04	0.61	0.7	13.88
	WebBrowsing	0.32	39.48	0.07	18.18	-0.18	4.97
All	Average	<b>0.77</b>	<b>29.65</b>	<b>0.10</b>	<b>7.38</b>	<b>0.15</b>	<b>11.02</b>

TABLE IX  
OVERALL PERFORMANCE COMPARISON AMONG DIFFERENT ALGORITHMS

Categories	Sequences	Proposed		MLMD [21]		OLBD [23]		FSCNET [25]	
		BDBR (%)	$\Delta T$ (%)	BDBR (%)	$\Delta T$ (%)	BDBR (%)	$\Delta T$ (%)	BDBR (%)	$\Delta T$ (%)
A	Robot	1.46	38.78	1.57	17.49	0.69	18.50	1.98	37.33
CC	EBURainFruits	0.23	53.40	1.32	16.87	0.27	28.01	0.07	36.44
	Kimono1	1.51	56.99	2.13	16.77	1.79	38.37	-0.07	24.66
Mixed	BasketballScreen	1.39	39.28	1.44	21.90	0.72	10.01	4.38	25.16
	MissionControlClip2	1.08	41.99	1.22	22.19	0.67	10.98	2.17	31.23
	MissionControlClip3	0.89	39.48	1.10	24.59	0.47	9.17	5.71	29.32
TGM	Console	0.43	41.39	2.33	27.37	0.49	16.74	1.98	23.69
	Desktop	0.30	51.03	1.56	30.89	0.5	20.72	2.24	32.29
	FlyingGraphics	3.25	38.28	1.71	24.28	0.57	13.19	6.66	18.51
	Map	1.54	26.66	1.65	20.56	1.03	10.84	2.28	26.87
	Programming	1.30	33.04	1.29	23.09	0.79	9.73	3.64	24.55
	SlideShow	1.37	28.53	1.74	17.86	0.85	11.01	3.77	23.19
	WebBrowsing	0.44	45.93	1.87	26.35	0.46	17.35	7.01	17.62
All	Average	<b>1.17</b>	<b>41.14</b>	<b>1.61</b>	<b>22.32</b>	<b>0.72</b>	<b>16.51</b>	<b>3.22</b>	<b>26.98</b>

al.'s algorithm has significantly deteriorated. This degradation in coding efficiency cannot be deemed an acceptable trade-off for improved coding performance. Consequently, Zuo et al.'s algorithm is not an ideal benchmark for comparison. To the best of our knowledge, these three algorithms are the state-of-the-art fast Intra coding algorithms for VVC SCC. To ensure a fair comparison, we implement these algorithms on the same computing platform and use identical experimental parameters. The comparison of coding performance and coding speed is presented in Table IX Overall performance comparison among different algorithmstable.9. In consideration of the practicality of our algorithm, the inference time of the models employed in our algorithm has been included in the experimental results.

From Table IX Overall performance comparison among different algorithmstable.9, it can be observed that coding speed improvements achieved by our proposed algorithms, MLMD, OLBD, and FSCNET, are 41.14%, 22.32%, 16.51% and 26.98%, respectively. The corresponding average BDBR increases by 1.17%, 1.61%, 0.72%, and 3.22%, respectively. When compared with FSCNET and MLMD, our proposed algorithm performs significantly better in terms of both coding

speed and coding efficiency. In comparison to OLBD, the proposed algorithm is faster by 24.63% with a 0.45% BDBR increase. Although the proposed algorithm exhibits slight coding efficiency losses, they can be considered negligible when considering the substantial coding speed improvements. Table VIII Performance of the three strategietable.8 presents the respective performance metrics in BDBR of the three proposed techniques, while Table IX Overall performance comparison among different algorithmstable.9 is their collective performance. Each proposed technique can improve coding speed but also results to misprediction. When the three proposed techniques are used simultaneously, they inevitably influence each other, leading to a further increase in misprediction. Therefore, the sum of the BDBR performances of the proposed method presented in Table IX Overall performance comparison among different algorithmstable.9 is greater than the sum of the BDBR performances of the three proposed techniques in Table VIII Performance of the three strategietable.8.

The VVC SCC introduces BDPCM and QTMT, but algorithms developed for HEVC SCC, such as MLMD and OLBD, have not investigated BDPCM and QTMT. In both VVC and

HEVC, CM selection and CU splitting exhibit notable similarities. To ensure a fair comparison, we primarily investigate CM selection and CU splitting in this paper. FSCNET is developed for VVC SCC, it is solely focused on CM selection, leading to only marginal improvements in coding speed. Furthermore, the trained model's accuracy is not sufficiently high, causing a significant loss of coding efficiency.

The key factors that contribute to the effective enhancement of coding speed by the proposed algorithm are as follows: Due to the substantial differences in the distribution of CMs, BDPCM, and SMs across various content types of CUs, deep learning is employed to classify CUs into distinct content categories. Building upon this classification, we independently predict candidate CMs, resulting in significantly improved prediction accuracy. Additionally, beyond CM prediction, we have advanced the early termination mechanisms for both BDPCM and CU splitting. Consequently, our algorithm significantly improves coding speed.

## IX. CONCLUSIONS

In this paper, we propose a fast mode and CU splitting decision for Intra prediction in VVC SCC. We initially employ deep learning techniques to classify CUs. Subsequently, we develop CM prediction, BDPCM early termination, and CU splitting early termination based on CU content types and features. Experimental results demonstrate that the proposed algorithm, on average, improves coding speed by 41.14%, with BDBR increasing by 1.17% on average. In our future work, we plan to develop efficient deep learning-based algorithms to further improve the coding speed of VVC SCC.

## REFERENCES

- [1] Y. Lu, S. Li, and H. Shen, "Virtualized screen: A third element for cloudmobile convergence," *IEEE MultiMedia*, vol. 18, no. 2, pp. 4–11, 2011.
- [2] "Joint call for proposals for coding of screen content," *ISO/IEC JTC1/SC29/WG11*, N14175, San Jose, CA, USA, Jan. 2014.
- [3] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging HEVC screen content coding extension," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 50–62, 2016.
- [4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [5] X. Xu, S. Liu, T.-D. Chuang, Y.-W. Huang, S.-M. Lei, K. Rapaka, C. Pang, V. Seregin, Y.-K. Wang, and M. Karczewicz, "Intra block copy in HEVC screen content coding extensions," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 409–419, 2016.
- [6] Z. Ma, W. Wang, M. Xu, and H. Yu, "Advanced screen content coding using color table and index map," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4399–4412, 2014.
- [7] Y.-W. Huang, C.-W. Hsu, C.-Y. Chen, T.-D. Chuang, S.-T. Hsiang, C.-C. Chen, M.-S. Chiang, C.-Y. Lai, C.-M. Tsai, Y.-C. Su, Z.-Y. Lin, Y.-L. Hsiao, O. Chubach, Y.-C. Lin, and S.-M. Lei, "A VVC proposal with quaternary tree plus binary-ternary tree coding block structure and advanced coding techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1311–1325, 2020.
- [8] M. Xu and B. Jeon, "Learning-based efficient quantizer selection for fast HEVC encoder," *IEEE Transactions on Broadcasting*, pp. 1–13, 2023.
- [9] M. Xu, T. Nguyen Canh, and B. Jeon, "Simplified level estimation for rate-distortion optimized quantization of HEVC," *IEEE Transactions on Broadcasting*, vol. 66, no. 1, pp. 88–99, 2020.
- [10] D. Wang, Y. Sun, C. Zhu, W. Li, F. Dufaux, and J. Luo, "Fast depth and mode decision in intra prediction for quality shvc," *IEEE Transactions on Image Processing*, vol. 29, pp. 6136–6150, 2020.
- [11] D. Wang, C. Zhu, Y. Sun, F. Dufaux, and Y. Huang, "Efficient multi-strategy intra prediction for quality scalable high efficiency video coding," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 2063–2074, 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," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 833–845, 2020.
- [13] 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," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1668–1682, 2020.
- [14] 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," *IEEE Transactions on Image Processing*, vol. 30, pp. 5377–5390, 2021.
- [15] H. Yang, L. Shen, and P. An, "An efficient intra coding algorithm based on statistical learning for screen content coding," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2468–2472.
- [16] J. Lei, D. Li, Z. Pan, Z. Sun, S. Kwong, and C. Hou, "Fast intra prediction based on content property analysis for low complexity HEVC-based screen content coding," *IEEE Transactions on Broadcasting*, vol. 63, no. 1, pp. 48–58, 2017.
- [17] C. Gao, L. Li, D. Liu, Z. Chen, W. Li, and F. Wu, "Two-step fast mode decision for intra coding of screen content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5608–5622, 2022.
- [18] Z. Pan, S. Kwong, M.-T. Sun, and J. Lei, "Early MERGE mode decision based on motion estimation and hierarchical depth correlation for HEVC," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 405–412, 2014.
- [19] S.-H. Tsang, Y.-L. Chan, W. Kuang, and W.-C. Siu, "Reduced-complexity intra block copy (IntraBC) mode with early cu splitting and pruning for HEVC screen content coding," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 269–283, 2019.
- [20] W. Kuang, S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Fast mode decision algorithm for HEVC screen content intra coding," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2473–2477.
- [21] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Machine learning-based fast intra mode decision for HEVC screen content coding via decision trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1481–1496, 2020.
- [22] S.-H. Tsang, Y.-L. Chan, and W. Kuang, "Mode skipping for HEVC screen content coding via random forest," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2433–2446, 2019.
- [23] 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," *IEEE Transactions on Image Processing*, vol. 29, pp. 170–185, 2020.
- [24] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "DeepSCC: Deep learning-based fast prediction network for screen content coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 1917–1932, 2020.
- [25] S.-H. Tsang, N.-W. Kwong, and Y.-L. Chan, "FastSCCNet: Fast mode decision in VVC screen content coding via fully convolutional network," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2020, pp. 177–180.
- [26] D. Wang, Y. Sun, J. Liu, F. Dufaux, X. Lu, and B. Hang, "Probability-based fast intra prediction algorithm for spatial SHVC," *IEEE Transactions on Broadcasting*, vol. 68, no. 1, pp. 83–96, 2022.
- [27] D. Wang, Y. Sun, W. Li, L. Xie, X. Lu, F. Dufaux, and C. Zhu, "Hybrid strategies for efficient intra prediction in spatial SHVC," *IEEE Transactions on Broadcasting*, vol. 69, no. 2, pp. 455–468, 2023.
- [28] 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," *IEEE Transactions on Broadcasting*, vol. 69, no. 2, pp. 469–481, 2023.
- [29] L. Zhu, Y. Zhang, Z. Pan, R. Wang, S. Kwong, and Z. Peng, "Binary and multi-class learning based low complexity optimization for HEVC encoding," *IEEE Transactions on Broadcasting*, vol. 63, no. 3, pp. 547–561, 2017.
- [30] H. L. Tan, C. C. Ko, and S. Rahardja, "Fast coding quad-tree decisions using prediction residuals statistics for high efficiency video coding (HEVC)," *IEEE Transactions on Broadcasting*, vol. 62, no. 1, pp. 128–133, 2016.

- [31] Z. Liu, W. Zhang, and M. Zhang, "A fast cu partition algorithm based on the coding cost in screen content coding," in *International Conference on Digital Image Processing*, 2018.
- [32] C. Yung-Hsuan, S. Yu-Chen, X. Jizheng, and X. Xiaozhong, "JVET common test conditions and software reference configurations for non-4:2:0 colour formats," document JVET, JVET-R2013, Teleconference, Apr. 2020.
- [33] T. L. Jing Guo, Liping Zhao, "Response to b1002 call for test materials: Five test sequences for screen content video coding," document JVETC0044, 3th JVET Meeting, Geneva, Switzerland, May 2016.
- [34] K. S. Karl Sharman, "Common test conditions," document JCTVC-X1100, 26th Meeting: Geneva, CH, January 2017.
- [35] S.-H. Tsang, N.-W. Kwong, and Y.-L. Chan, "Ultra fast screen content coding via random forest," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020, pp. 1112–1117.
- [36] X. Dong, L. Shen, M. Yu, and H. Yang, "Fast intra mode decision algorithm for versatile video coding," *IEEE Transactions on Multimedia*, vol. 24, pp. 400–414, 2022.
- [37] "VVC test model version 17.0 VTM-17.0," Available: [https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware\\_VTM](https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM).
- [38] G. Bjontegaard, "Calculation of average PSNR differences between RD-Curves," *ITU T Q/sg16 Vceg Vceg M*, 2001.
- [39] T. Amestoy, A. Mercat, W. Hamidouche, D. Menard, and C. Bergeron, "Tunable VVC frame partitioning based on lightweight machine learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 1313–1328, 2020.
- [40] J. Chen, H. Sun, J. Katto, X. Zeng, and Y. Fan, "Fast QTMT partition decision algorithm in VVC intra coding based on variance and gradient," in *2019 IEEE Visual Communications and Image Processing (VCIP)*, 2019, pp. 1–4.
- [41] Y. Zuo, C. Gao, D. Liu, L. Li, Y. Zhang, and X. Sun, "Learned rate-distortion cost prediction for ultrafast screen content intra coding," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023.



**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.



**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.



**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.



**Bo Hang** received his M.S. degree in computer science from Harbin Engineering University, Harbin, Heilongjiang province, P.R.China, in 2003, and received his Ph.D. degree in Communication and Information System from Wuhan University, Wuhan, Hubei province, P.R.China in 2012. Now he is a professor of Computer School, Hubei University of Arts and Science, Xiayang, Hubei province, P.R.China. His current research interests include multimedia information processing and computer networks.



**Hui Guo** received the M.Sc. degree from Guangxi Normal University, Guilin, China, in 2010. She is currently pursuing the Ph.D. degree with the Macao University of Science and Technology, Macau, China. Meanwhile, 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, 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.