

Article

A Fast VVC Intra Prediction Based on Gradient Analysis and Multi-Feature Fusion CNN

Zhiyong Jing, Wendi Zhu and Qiuwen Zhang *

College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China; jingzhiyong@zzuli.edu.cn (Z.J.); 332107040626@email.zzuli.edu.cn (W.Z.)
* Correspondence: 2012032@zzuli.edu.cn; Tel.: +86-371-8660-9559

Abstract: The Joint Video Exploration Team (JVET) has created the Versatile Video Coding Standard (VVC/H.266), the most up-to-date video coding standard, offering a broad selection of coding tools. The maturity of commercial VVC codecs can significantly reduce costs and improve coding efficiency. However, the latest video coding standards have introduced binomial and trinomial tree partitioning methods, which cause the coding units (CUs) to have various shapes, increasing the complexity of coding. This article proposes a technique to simplify VVC intra prediction through the use of gradient analysis and a multi-feature fusion CNN. The gradient of CUs is computed by employing the Sobel operator, the calculation results are used for predecision-making. Further decisions can be made by CNN for coding units that cannot be judged whether they should be segmented or not. We calculate the standard deviation (SD) and the initial depth as the input features of the CNN. To implement this method, the initial depth can be determined by constructing a segmented depth prediction dictionary. For the initial segmentation depth of the coding unit, regardless of its shape, it can also be determined by consulting the dictionary. The algorithm can determine whether to split CUs of varying sizes, decreasing the complexity of the CU division process and making VVC more practical. Experimental results demonstrate that the proposed algorithm can reduce encoding time by 36.56% with a minimal increase of 1.06% Bjøntegaard delta bit rate (BD-BR) compared to the original algorithm.

Keywords: H.266/VVC; intra prediction; CNN; multi-feature fusion



Citation: Jing, Z.; Zhu, W.; Zhang, Q. A Fast VVC Intra Prediction Based on Gradient Analysis and Multi-Feature Fusion CNN. *Electronics* **2023**, *12*, 1963. <https://doi.org/10.3390/electronics12091963>

Academic Editors: Ionut Schiopu and Radu Ciprian Bilcu

Received: 15 March 2023

Revised: 19 April 2023

Accepted: 20 April 2023

Published: 23 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today, digital video has been widely adopted for a variety of purposes, including multimedia messaging, video telephony, video conferencing, high-resolution format display, VR panoramic technology, mobile internet live streaming and digital film [1]. The emergence of various application scenarios will further require the effectiveness and functionality of video compression, and the past coding standards cannot meet the needs of real-life applications. Advanced video codecs have been developed, but they require more complex computations to achieve optimal results. Block segmentation is a key technique in video coding and it takes up most of the time in the coding process; based on this conclusion, finding a way to speed up CU segmentation is important to save coding time.

In H.264 [2], frames are divided into fixed-size macro blocks (MBs) that are 16×16 blocks in size, while H.265/HEVC [3] introduces quadtree partitioning (QT) as a new technology. QT divides each frame into 64×64 , followed by the further division of the CTU into smaller CUs of varying sizes (square-shaped). The minimum size of the split is 8×8 .

The Joint Video Exploration Team (JVET) has proposed a new method for VVC block division. On the basis of the original quadtree division, a method of quadtree-nested multi-type tree (QTMT) is proposed. The maximum size of CTU is also changed from 64×64 to 128×128 . The specific division form is shown in Figure 1. VVC has added the segmentation method of binary tree (BT) and ternary tree (TT). The binary tree partition

is divided in equal proportions, while the ratio of ternary tree partition is 1:2:1. These two segmentation methods can be performed in both horizontal and vertical directions. Since the QTMT division structure supports multiple division types, different division types may be divided into the same coding block structure, resulting in redundant division. Therefore, some redundant division methods are restricted in H.266/VVC. If the CU is divided into VTT, the sub-CU in the middle is prohibited from using VBT division to prevent overlapping with two VBT divisions. If the CU is divided into VBT and one of its sub-CUs is divided into HBT, the other sub-CU is prohibited from being used, and the HBT division prevents overlapping with a QT division. If a CU is divided into VBT and one of its sub-CUs is further divided into HTT, the other sub-CU is prohibited from using HTT division to prevent overlapping with a VBT division and an HTT division. For the same reason, the above prohibition of partitioning rules also applies to the horizontal direction. Figure 1a takes CTU as an example to illustrate and Figure 1b shows all possible division modes. Rate distortion optimization (RDO) is the basis for determining CU partitions. For each CU, all possible partition modes are iterated, the rate-distortion (RD) cost is computed, and then the least costly division method is chosen [4]. RD cost J is determined by the following Formula (1):

$$J_{mode} = D + (\lambda \times R_{mode}), \tag{1}$$

$$D = SSE_{luma} + (W_{chroma} \times SSE_{chroma}), \tag{2}$$

where R_{mode} represents the encoding bits of the homologous *mode*; SSE_{chroma} is the aggregate of chromaticity squares for the original and reconstructed image. Similarly, SSE_{luma} represents the sum of squares of brightness; λ and W_{chroma} are the Lagrange multiplier and *chroma* distortion weights, respectively.

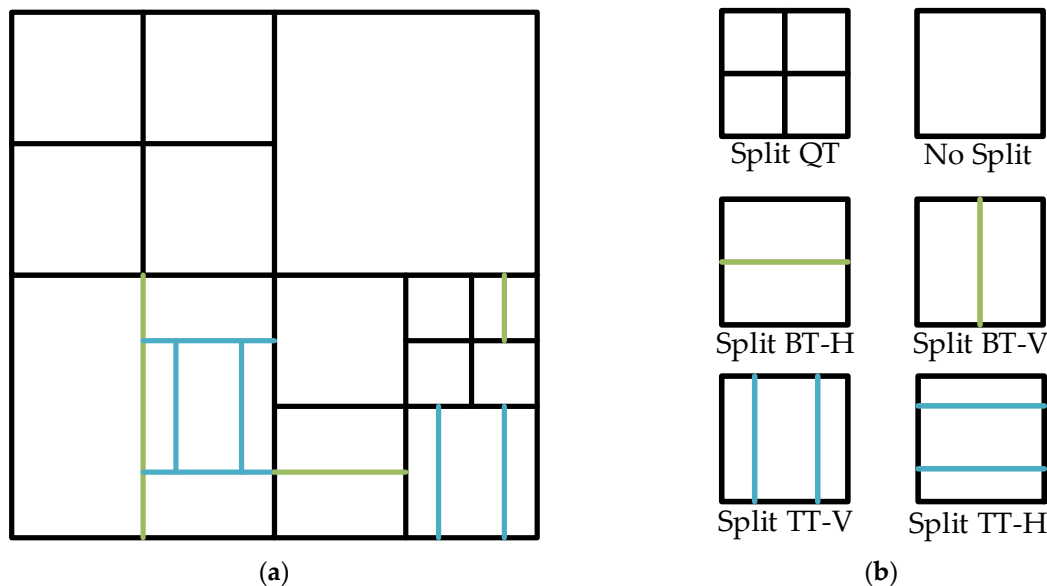


Figure 1. CTU partitioning in VVC. (a) An example of CUT division; (b) specific division method.

In order to determine the final division pattern of a CU in VVC, the RD cost of quadtree, binomial tree and trinomial tree division patterns of all depths need to be calculated iteratively, and the partition with the lowest RD cost is the final selection. Since all CUs have to be traversed, the newly proposed QTMT structure greatly increases the computational cost of CU partitioning [5]. The VVC coding standard introduces immense computational complexity while achieving significant bit rate compression gain, which greatly hinders the development of video coding technology. Although there are rich machine learning-based research results on reducing the computational burden of CU partition decision while maintaining coding quality, it also has some limitations, such as an inability to adapt to

different sizes of CUs and the need to call models repeatedly. In order to further improve the intra-frame coding efficiency of H.266/VVC, it is urgent and necessary to solve the above problems [1]. Motivated by this, we introduce a multi-feature fusion CNN to simplify VVC intra prediction and a preprocessing algorithm based on gradient computation.

The subsequent sections are structured as follows. Section 2 outlines the related work in this paper. Section 3 outlines the overall algorithm process through a flowchart, introduce related methods and explain the dataset building and training scheme. Section 4 shows the experimental results of the algorithm and contrast it with other algorithms. Finally, Section 5 concludes the paper.

2. Related Work

Over the past decade, accelerating the CU division has been the focal point of research in video coding, with numerous methods proposed for implementation in HEVC and the most recent coding standards. In addition to traditional methods, related artificial intelligence algorithms are becoming more and more popular.

2.1. Related Method of HEVC

Based on the texture complexity of CUs, Shen et al. [6] introduced a method for early decision-making in CU partition that utilizes an adaptive threshold. Lee et al. [7] proposed three kinds of jumping decisions, which considerably diminished the intricacy of attainable coding. Soulef et al. [8] combined two machine learning methods. These two methods can reduce the coding complexity and make decisions on the division method. Kuanar et al. [9] proposed a neural network-based method to classify based on image features and reduce the complexity of predictive models. Kuo et al. [10] suggested to check the coding unit boundary with a smaller size through the deblocking filter (DBF), which reduced the cost of encoding and reduced the encoding time by 59.73%. The method suggested by Kuang et al. [11] is applied to HEVC-based screen content coding (SCC), mainly using Bayesian decision rules to make fast decisions on CUs. Zhang et al. [12] integrated conventional techniques with CNN approaches by designing a CNN architecture to forecast the CU partition mode for various depths. Siham et al. [13] initially employed the conventional approach to assess the homogeneity of coding units (CUs) and subsequently constructed a classification model of CUs by extracting features. This methodology has been employed in the extension of 3D high-efficiency video coding (3D-HEVC) based on the HEVC framework. Fu et al. [14] introduced an innovative early skip approach for coding units that relies primarily on a series of decision trees to enable accurate judgment, resulting in time savings of up to 71.63%.

2.2. Approaches for VVC

Zhang et al. [15] established a correlation between the characteristics of the coding unit and its partition mode and preemptively discarded redundant partition modes to mitigate encoding complexity. Tang et al. [16] employed an edge detector to extract the salient edge features of the CU. Such an approach enabled the model to bypass superfluous vertical or horizontal split modes and thereby minimize encoding time. Fan et al. [17] mainly used the traditional method, using the Sobel operator to extract gradient features to make decisions on the division mode. Experiments showed that this method is more effective than deep learning. Wang et al. [18] proposed a CNN model with multi-level fetch termination that can predict all CU partition patterns of the size 32×32 . Ni et al. [19] devised a partition strategy for binary and ternary trees by calculating the gradient and applying regression analysis. To accommodate CUs of varying sizes, Pan et al. [20] proposed a CNN model that fuses multiple sources of information to enable the early termination of the QTMT partitioning process, thus reducing the encoding complexity. Li et al. [21] developed two decision-making models for binary and ternary tree partitioning. The key feature of this method is the adjustability of the decision-making model, which can strike a balance between encoding loss and encoding time. Finally, Tang et al. [22] proposed a Laplacian-

based segmentation algorithm that enables the early termination of judge. The algorithm calculates the texture degree of each pixel and assigns three different thresholds to the three sizes of CUs. Consequently, a CU will not split further when its texture degree falls below the corresponding threshold. This approach optimizes the encoding process by saving time.

3. The Proposed Algorithm

CNN has proven to be an effective tool in various applications, including video coding. To accommodate different sizes of CUs, we extracted their features as the input to the CNN. In order to curtail the coding time of VVC, we presented a novel approach for making fast decisions regarding CU division, which leveraged gradient analysis and multi-feature fusion CNN. The flowchart of this method is depicted in Figure 2.

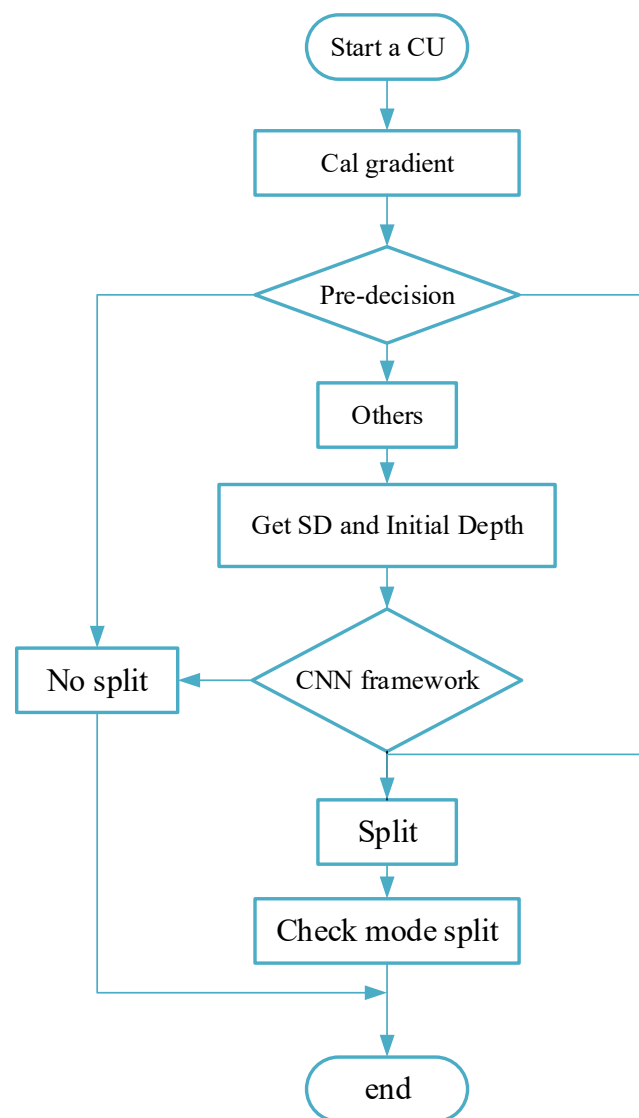


Figure 2. The proposed method is illustrated through a flowchart.

The gradient of the residual block for a single CU, regardless of its shape, will be calculated and used to make a predecision. Finally, the CU that cannot be judged by the gradient processing will be handed over to the CNN for decision-making, while those judged to be nonsegmented will directly end the algorithm and those resulting in segmentation will be further judged by CheckModeSplit for the division method. This

section will outline the algorithm of preprocessing, multi-feature fusion CNN architecture and training scheme.

3.1. Gradient-Based Early Decision Methods

Based on the optical flow theory, the gradient direction of a pixel represents its maximum change direction and the direction perpendicular to the pixel represents the minimum change direction. For example, it can be easily understood that the minimum and maximum change directions on the edge of the object are tangent and vertical, respectively, and this is further expressed in Figure 3. Therefore, the gradient direction can be leveraged to determine the optimal intra mode.

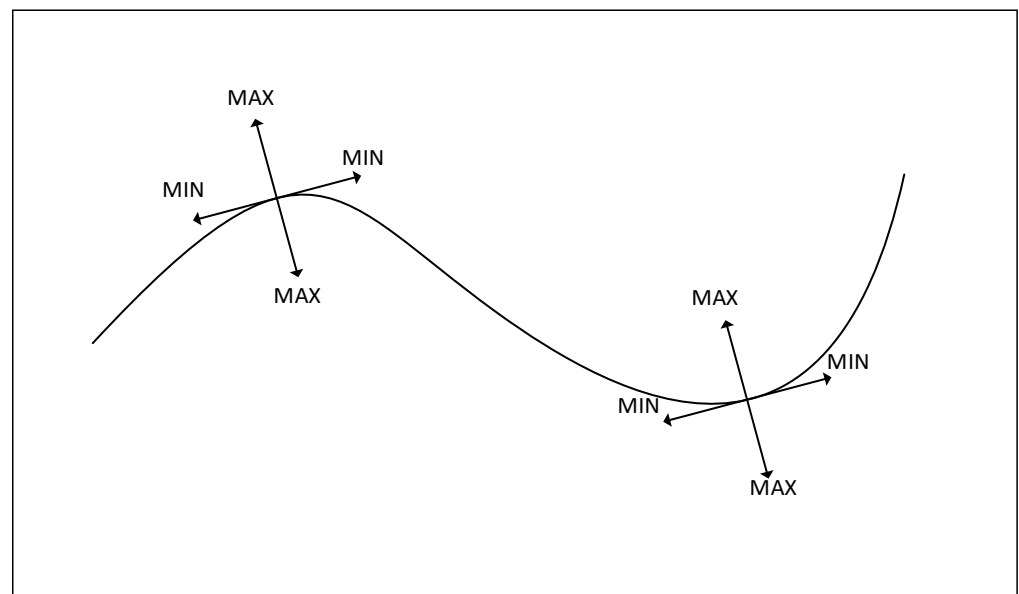


Figure 3. MAX and MIN variation directions of an object edge.

This chapter also uses Sobel operator to express direction complexity (DC) [23], which can obtain effective computable gradient estimation. Figure 4a,c depict the Sobel operator for measuring the horizontal and vertical direction characteristics, respectively. However, for an image content with high computational complexity, only horizontal and vertical directions are not sufficient [24]. Henceforth, the estimation of DC is performed by incorporating diagonal Sobel operator components of 45° and 135°, as exemplified in Figure 4b,d.

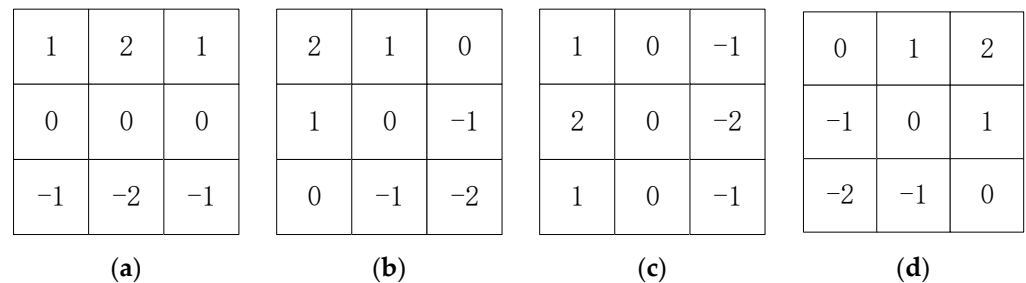


Figure 4. Sobel masks for gradients calculation. (a) S_{0° . (b) S_{45° . (c) S_{90° . (d) S_{135° .

Each pixel of the coding unit (CU) undergoes gradient calculation by employing the four Sobel operators, wherein the gradient is computed in every direction, as follows:

$$G_d(a, b) = S_d * A, (d = 0^\circ, 45^\circ, 90^\circ, 135^\circ), \tag{3}$$

$$A = \begin{pmatrix} f(a-1, b-1) & f(a-1, b) & f(a-1, b+1) \\ f(a, b-1) & f(a, b) & f(a, b+1) \\ f(a+1, b-1) & f(a+1, b) & f(a+1, b+1) \end{pmatrix}, \quad (4)$$

where $f(a, b)$ represents the brightness value of the pixel at position (a, b) . $f(a-1, b-1)$, $f(a-1, b)$, $f(a-1, b+1)$, $f(a, b-1)$, $f(a, b)$, $f(a, b+1)$, $f(a+1, b-1)$, $f(a+1, b)$, $f(a+1, b+1)$, respectively, at position (a, b) is the brightness value of surrounding pixels; and (a, b) represents the position of a pixel. DC is defined as follows:

$$DC = \frac{1}{W \times H} \sum_{a=0}^{W-1} \sum_{b=0}^{H-1} (|G_{0^\circ}(a, b)| + |G_{45^\circ}(a, b)| + |G_{90^\circ}(a, b)| + |G_{135^\circ}(a, b)|) \quad (5)$$

where W and H indicate the width and height of the CU, respectively. $G_{0^\circ}(a, b)$, $G_{45^\circ}(a, b)$, $G_{90^\circ}(a, b)$ and $G_{135^\circ}(a, b)$ represent the gradient values in four directions, respectively. Based on the gradient values in these four directions. DC can well-reflect the direction complexity of the CU.

The variables W and H represent the width and height of the CU, respectively. $G_{0^\circ}(a, b)$, $G_{45^\circ}(a, b)$, $G_{90^\circ}(a, b)$ and $G_{135^\circ}(a, b)$ represent the gradient values in the four directions, respectively. Utilizing the gradients of the aforementioned cardinal directions, it can be inferred that DC is capable of effectively elucidating the directional intricacies of CUs.

As a consequence of the intricate patterns and textures that characterize CUs, certain ones may present more challenges when it comes to partitioning. To save time, we can utilize algorithms to detect them. We employ a gradient-based approach to make preliminary decisions. The gradient in the x and y directions, denoted as g_x and g_y , is determined by the Sobel operator. The equation is as follows:

$$grad = \frac{\sum_{i=1}^w \sum_{j=1}^h (g_x^2 + g_y^2)}{w \cdot h} \quad (6)$$

Upon the calculation of the gradient, a forecast can be deduced by means of Equation (6). Grad represents the gradient within the horizontal and vertical orientations, whereupon a comparison is made between the gradient of the calculated CU and parameter Q . Q is the maximum value of QP^2 and QS^2 , where QP stands for the quantization parameter and QS for the quantization step size. The lower limit of these parameters is specified in [25], while the upper limit is determined through extensive statistical analysis. If the result of the formula is no split, the CU will bypass the checking of the division pattern and terminate. If a split is obtained, the CU will not be passed to CNN for processing and begin to evaluate the division pattern. The residual block of the CU is an input to the CNN only after the result sent to the CNN has been obtained.

$$result = \begin{cases} no\ split & (grad < Q \cdot 0.15) \\ sent\ to\ CNN & (uncertain), \\ split & (grad > Q \cdot 8) \end{cases}, \quad (7)$$

Our comprehensive test revealed that the predecision can accurately ascertain whether around 5.4% of the CUs are split or not. It is more time-efficient than CNN. Therefore, pre-decision can filter some CUs without relying on CNN, thus saving external time for CNN and streamlining its training.

3.2. Calculate Standard Deviation

During the encoding process, a larger coding unit (CU) is employed to depict a single region of the image. Conversely, smaller CUs are employed for regions with intricate details. Thus, we can take the texture complexity of CUs as an important feature of CUs. Inspired by the heuristic method [26,27], the standard deviation of CUs can be used to measure

the energy difference between two pixels, and it is the most widely employed metric for measuring texture complexity. Henceforth, we shall evaluate the standard deviation of the CU residual block and employ *SD* as an input parameter for the multi-feature fusion CNN. The calculation process of *SD* is delineated below [28]:

$$SD = \sqrt{\frac{1}{W \times H} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} p(a,b)^2 - \left(\frac{1}{W \times h} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} p(a,b) \right)^2}, \tag{8}$$

where *W* and *H* represent the width and height of the CU, respectively, and *p(a, b)* denotes the value of the pixel located at (*a, b*).

3.3. Determine the Initial Segmentation Depth of CUs by Prediction Dictionary

The depth of a CU’s division is intrinsically tied to the content and texture. CUs with complex textures are more susceptible to be divided, whereas those with simpler textures are less likely to be fragmented. The setting of QP also affects the division pattern of a CU when coding. In this section, a prediction dictionary is built to explore the effect of texture and QP on CU depth, the initial depth of the CU division is determined and the initial depth of the CU division is identified. The joint solution of the decision dictionary is created based on texture and QP. VVC standard test sequences with various themes, varied content and varied resolutions are selected (Campfire, Basketballdrill, Kimono, CatRobot1). The specific results of this scheme are shown in Figure 5. Firstly, we have to count the results of the division of the various shapes of the CU and encode the different QPs using the encoder. The QP is taken as the horizontal axis (0–51) and the unit of each cell is 1; the entropy value is taken as the vertical axis (0.5–7.5) and the unit of each cell is 0.1.

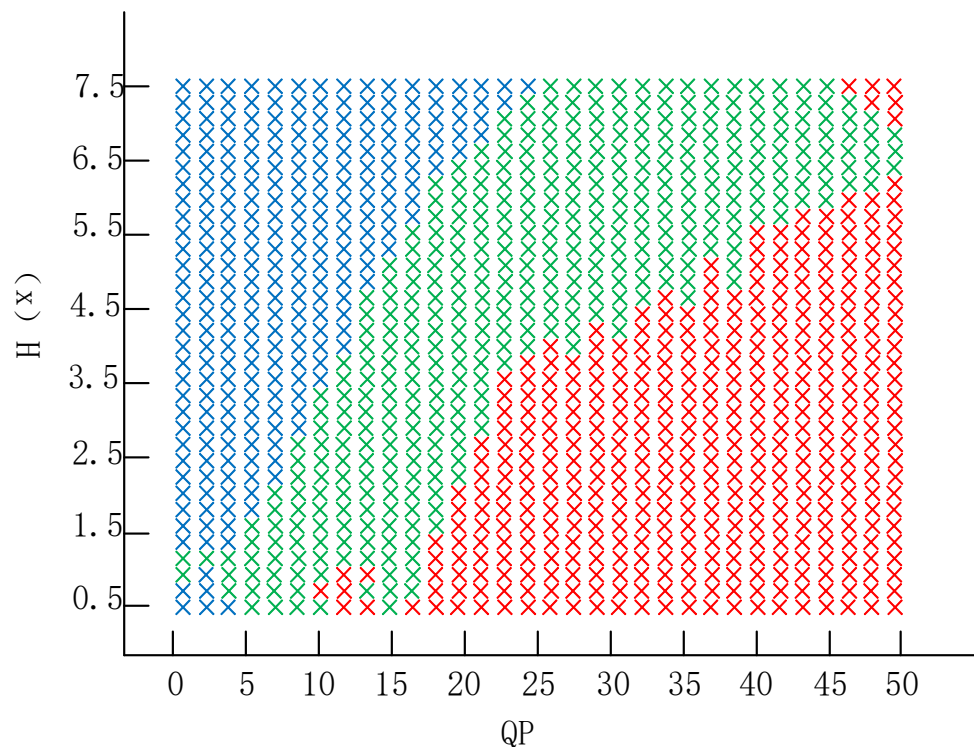


Figure 5. Determine the initial depth of CUs.

Figure 5 indicates a prediction dictionary of CUs’ initial depth, with the red cross (×) showing that the test results have more simple coding units than complex ones, and CUs with the corresponding QP and entropy values are simple coding units. The blue cross (×) shows that the test results have more complex coding units than simple ones, and CUs

with the corresponding QP and entropy values are complex coding units. The green cross (×) indicates that the simple coding unit is ultimately equal to the complex coding unit in the test result, and the CU corresponding to QP and entropy value is a general coding unit. The dictionary lookup determines the initial depth of the CU: 0 for simple CUs, 1 for general CUs, and 2 for complex CUs.

3.4. Multi-Feature Fusion CNN

In order to judge CUs of different sizes and speed up the division, we propose a multi-feature fusion CNN. We set different parameters for Conv Core and FCL and obtain the prediction accuracy of CNN under different parameters, as shown in Table 1. Finally, we choose the parameters with the best accuracy to set CNN.

Table 1. Prediction accuracy of CNN with different parameters.

(%)	Comparison of Different Parameters			Accuracy
	Conv	Kernel	FCL	
Default parameter	4	6	2	72.30
Kernel comparison	4	8	2	81.25
	4	10	2	87.65
Conv comparison	2	6	2	73.45
	4	6	2	80.26
FCL comparison	4	6	2	71.26
	4	6	3	76.51

Figure 6 displays the flow chart of a CNN architecture. A CNN architecture combining texture and depth features has been constructed due to the conventional CNN structure. This convolutional neural network architecture comprises two channels, each of which encompasses convolutional, pooling and fully connected layers. We only need to input the SD and the initial segmentation depth to obtain the corresponding results. Because the pool layer is an effective component of the CNN architecture, we utilize the largest pooling type. The rectified linear unit (ReLU) function has become a ubiquitous activation function for various neural network architectures. The use of ReLU as the activation function often leads to better performance and more straightforward training. We apply the ReLU function to all convolutional and fully connected layers of the proposed CNN.

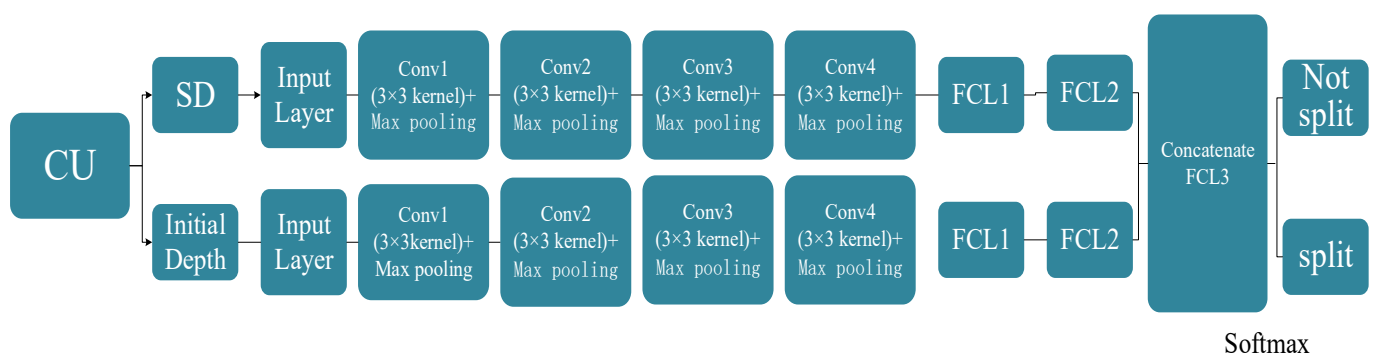


Figure 6. Specific details of CNN architecture.

Moreover, loss function (LF) is employed to boost the classification accuracy by categorizing the ultimate output, thus ensuring a precise classification of the CU with these parameters.

3.5. CNN Training

Since HEVC only performs quadtree division, when CNN is applied to past coding standards, the input sample size is square. Due to the varying shapes of coding units in VVC, traditional CNN training methods are not applicable. Our proposed CNN structure divides the training data into batches based on size; the samples are assigned to batches according to their size with those of equal size grouped together. This allows for batch training of the CNN architecture, with different shapes of CUs trained separately.

Multi-feature fusion CNNs extract features from training samples, which are then used as inputs for a two-way training CNN structure, as shown in Table 2. We select videos with different content, different topics and resolutions from the VVC standard test sequence for training, such as “Kristen AndSara”, “Kimono”, “CatRobot1” and “PartyScene”. The selected video sequences are encoded under various QPs (22, 27, 32 and 37). Initially, 35 frames are randomly chosen, of which the initial 30 frames are utilized to train the CNN. The remaining five frames are held out as a test set to evaluate the performance of the model.

Table 2. Table of video sequence selection.

Sequence	Class	Resolution
Kristen AndSara	E	1280 × 720
Kimono	B	1920 × 1080
CatRobot1	A	3840 × 2160
PartyScene	C	832 × 480

Following the encoding process, we assign labels to the CU residual blocks of various dimensions based on the CU division. Specifically, a divided CU is marked with “1.0” while an undivided CU is designated as “0.1”. Given the adaptability of the multi-feature fusion CNN to varying CU sizes, it is unsuitable to rely on traditional training methods for dataset construction. Therefore, the training and testing datasets must be fragmented into multiple sets, each contingent upon the size of the CU. Subsequently, a gradient analysis is conducted to sift through the extracted datasets and eliminate CUs that do not necessitate CNN, resulting in the final dataset. The CNN model is fine-tuned through optimization via stochastic gradient descent (SGD) techniques, with the cross-entropy function serving as the *loss* function. The formulation of the *loss* function is as follows:

$$loss = \sum_{i=1}^n [T_i \log(\hat{P}) + (1 - T_i) \log(1 - \hat{P})], \quad (9)$$

$$\hat{P} = \frac{e^{x_i}}{\sum_{i=1}^n e^{x_j}}, \quad (10)$$

T_i denotes the veritable value, whereas \hat{P} signifies the predicted value that has undergone processing through the softmax function.

4. Experimental Results

In this chapter, we first introduce the experimental environment and related configurations and the selection of experimental samples, and then explain the evaluation criteria of algorithm performance. These evaluation criteria are compared with previous algorithms to demonstrate the good performance of the proposed algorithm.

4.1. Experimental Setup

To evaluate the performance of the algorithm, JEVF formulated 26 test video sequences. According to different resolutions and video content characteristics, it is divided into seven categories: A1, A2 and B–F, including different resolutions, frame numbers, frame rates and bit depths. At the same time, JVET promulgated the standard test software VTM of

H.266/VVC, and all the tests of the improved algorithms need to be integrated on the VTM. Our empirical investigations were executed on a Windows 10 machine, outfitted with an AMD Ryzen 5 3600 processor, operating at a frequency of 3.60 GHz; random access memory was 8 GB. For the experimentation, we employed version 10.0 of the official VVC test software. The video sequences, selected from the catalog of sequences specified by the Joint Video Exploration Team, amounted to 14 sequences, each of which was classified according to different resolutions: A (3840×2160), B (1920×1080), C (832×480), D (416×240) and E (1280×720). In order to guarantee the credibility of our experimental findings, we encoded under varied quantization parameters (22, 27, 32 and 37), taking the average value as the experimental outcome.

The performance of the fast algorithm depends on whether the encoding time can be saved while maintaining the video quality as much as possible. Therefore, the evaluation of fast algorithms needs to comprehensively consider factors such as encoding time saving, bit rate, etc. We have selectively focused on three parameters from the experimental results. We have used BD-BR as the metric to evaluate its performance. The reduction in coding complexity is expressed by calculating the time saving (TS) rate of the proposed algorithm compared to the VTM anchoring algorithm to encode videos.

$$TS(\%) = \frac{T_{VTM10.0} - T_{proposed}}{T_{VTM10.0}} \quad (11)$$

$T_{VTM10.0}$ indicates the encoding time of VTM10.0; $T_{proposed}$ indicates the encoding time of the proposed method. In addition, to draw a comparison with the previous algorithm, we have computed the TS/BD-BR ratio to effectively evaluate the trade-off between encoding complexity and quality.

4.2. Results Presentation and Comparative Analysis

We subjected our algorithm to a comparative analysis against Li et al.'s [29] and Tang et al.'s [16] works. These two algorithms are evaluated using VTM7.0 and VTM4.0, respectively. The result selection in [29] was derived from their algorithm's "fast" mode. The comprehensive experimental findings are detailed in Table 3.

Table 3. Comparison of the proposed algorithm with previous algorithms.

Class	Sequence	Ref. [29], VTM7.0			Ref. [16], VTM4.0			Proposed Algorithm		
		BD-BR(%)	TS(%)	TS/BD-BR	BD-BR(%)	TS(%)	TS/BD-BR(%)	BD-BR(%)	TS(%)	TS/BD-BR(%)
A	Campfire	2.91	59.87	20.57	/	/	/	1.02	34.80	34.11
	CatRobot1	3.28	55.99	17.07	/	/	/	1.06	38.71	36.52
B	Kimono	/	/	/	1.98	41.82	21.12	1.13	38.56	34.12
	MarketPlace	1.28	58.22	45.48	/	/	/	0.87	34.12	39.22
	BQTerrace	1.79	56.94	31.81	1.19	29.47	24.76	0.97	33.89	47.73
	Cactus	1.86	60.56	32.56	/	/	/	1.05	35.86	34.15
C	BasketballDrill	2.98	52.62	17.66	1.36	28.73	21.13	1.25	38.40	51.2
	RaceHorsesC	1.61	57.89	35.96	2.96	33.89	11.45	0.89	37.69	56.25
	PartyScene	1.16	58.94	50.81	1.05	35.23	33.55	1.16	34.83	30.03
D	BQSquare	1.33	55.16	41.47	1.19	29.47	24.76	0.94	38.93	52.61
	BlowingBubbles	1.57	53.40	34.01	0.73	21.87	29.96	1.08	34.75	32.18
	RaceHorses	1.88	53.34	28.37	2.96	33.89	11.45	1.34	36.03	26.89
E	FourPeople	2.20	59.74	27.15	1.37	26.65	19.45	1.05	37.66	44.31
	Kristen	2.75	60.01	21.82	1.53	25.32	16.55	0.97	37.62	38.78
	AndSara									
Average		2.05	57.13	27.87	1.63	30.63	18.79	1.06	36.56	34.49

Based on the experimental data presented in the above table, it is evident that the proposed method saves 36.56% of the encoding time when compared to the conventional approach, with only a minimal increase of 1.06% in BD-BR. In Figure 7, we selected several sequences and utilized TS as the comparison index to contrast with [16]. It is clearly discernible that the proposed algorithm has a more favorable impact. Figure 8 depicts the RD performance of the proposed method in comparison to VTM10.0 for test videos. It is evident from the figure that the proposed scheme achieves comparable RD performance to VTM.

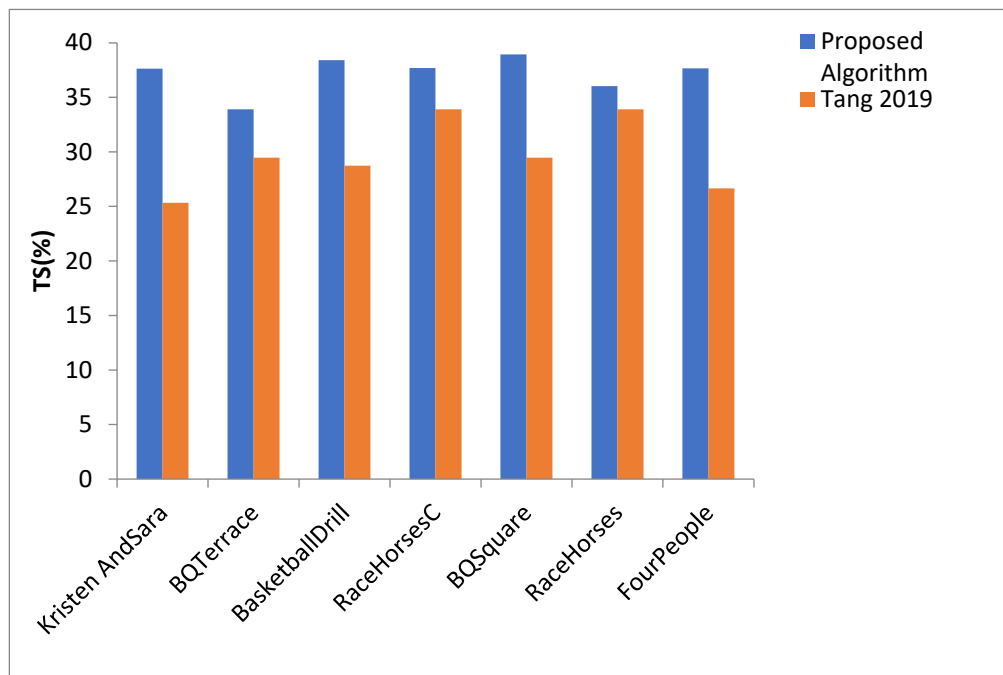


Figure 7. Histogram comparing the experimental results of the proposed algorithm with those of ref. [16].

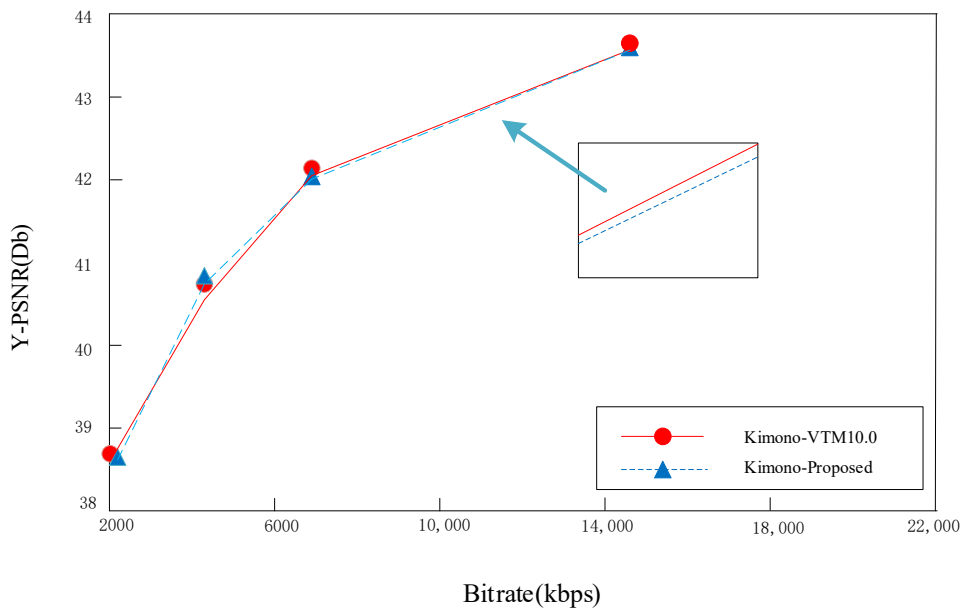


Figure 8. RD capability of the proposed method.

The algorithms proposed in [29] are effective in reducing encoding time; however, they did not account for the increase in BD-BR. When BD-BR is positive, this leads to an

increase in code rate, resulting in a decrease in encoding performance. Therefore, we will compare the TS/BD-BR of the two algorithms; the average TS/BD-BR of the proposed algorithm is 34.49%, which is significantly higher than the algorithm in [29].

In conclusion, our algorithm offers significant advantages compared to existing ones, especially in achieving a balance between the diminution of intricacy and the augmentation of bit rate.

5. Conclusions

The present study puts forth a novel VVC intraframe prediction technique that leverages gradient analysis and multifeature fusion to expedite encoding by curtailing the complex RDO calculation. Notably, it successfully merges the traditional and deep learning algorithms. The algorithmic flow of our technique is designed in such a way that both algorithms function independently yet complementarily, substantially simplifying VVC intra prediction, hastening CU division and trimming encoding time. Comparing our proposed method with the original algorithm, we find that our approach reduces encoding complexity by 36.56%, with only a marginal 1.06% increase in BD-BR. Moreover, a comparison with the previously proposed algorithms reveals that our technique strikes an optimal balance between saving encoding time and boosting BD-BR, thus reinforcing its superiority in this domain. Although the scheme proposed in this paper can decide whether to divide CUs of different sizes, it does not fully consider the decision of its division type. The follow-up work will conduct in-depth research on CU partition type decisions of different sizes and design an algorithm for partition type decision-making on CUs of different sizes, which can completely skip the RDO process and further speed up the CU partition process.

Author Contributions: Conceptualization, Z.J. and W.Z.; methodology, Z.J.; software, W.Z.; validation, Z.J., Q.Z. and W.Z.; formal analysis, W.Z.; investigation, W.Z.; resources, Q.Z.; data curation, W.Z.; writing—original draft, W.Z.; writing—review and editing, Z.J.; visualization, Z.J.; supervision, Q.Z.; project administration, Q.Z.; funding acquisition, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China No. 61771432, 61302118, and 61702464, the Basic Research Projects of Education Department of Henan No. 21zx003, and No. 20A880004, the Key Research and Development Program of Henan No. 222102210156, and the Postgraduate Education Reform and Quality Improvement Project of Henan Province YJS2021KC12 and YJS2022AL034.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the editors and the anonymous reviewers for their helpful comments and suggestions which have improved the presentation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qian, X.; Zeng, Y.; Wang, W.; Zhang, Q. Co-saliency Detection Guided by Group Weakly Supervised Learning. *IEEE Trans. Multimed.* **2022**, *1*. [\[CrossRef\]](#)
2. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A. Overview of the H. 264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 560–576. [\[CrossRef\]](#)
3. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [\[CrossRef\]](#)
4. Bross, B.; Chen, J.; Ohm, J.R.; Sullivan, G.J.; Wang, Y.K. Developments in international video coding standardization after avc, with an overview of versatile video coding (vvc). *Proc. IEEE* **2021**, *109*, 1463–1493. [\[CrossRef\]](#)
5. Mercat, A.; Mäkinen, A.; Sainio, J.; Lemmetti, A.; Viitanen, M.; Vanne, J. Comparative rate-distortion-complexity analysis of VVC and HEVC video codecs. *IEEE Access* **2021**, *9*, 67813–67828. [\[CrossRef\]](#)
6. Shen, L.; Zhang, Z.; Liu, Z. Effective CU size decision for HEVC intra coding. *IEEE Trans. Image Process.* **2014**, *23*, 4232–4241. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Lee, J.; Kim, S.; Lim, K.; Lee, S. A fast CU size decision algorithm for HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *25*, 411–421.

8. Bouaafia, S.; Khemiri, R.; Sayadi, F.E.; Atri, M. Fast CU partition-based machine learning approach for reducing HEVC complexity. *J. Real-Time Image Process.* **2020**, *17*, 185–196. [[CrossRef](#)]
9. Kuanar, S.; Rao, K.R.; Bilas, M.; Bredow, J. Adaptive CU mode selection in HEVC intra prediction: A deep learning approach. *Circuits Syst. Signal Process.* **2019**, *38*, 5081–5102. [[CrossRef](#)]
10. Kuo, Y.T.; Chen, P.Y.; Lin, H.C. A spatiotemporal content-based CU size decision algorithm for HEVC. *IEEE Trans. Broadcast.* **2020**, *66*, 100–112. [[CrossRef](#)]
11. Kuang, W.; Chan, Y.L.; Tsang, S.H.; Siu, W.C. Online-learning-based Bayesian decision rule for fast intra mode and CU partitioning algorithm in HEVC screen content coding. *IEEE Trans. Broadcast.* **2019**, *29*, 170–185. [[CrossRef](#)] [[PubMed](#)]
12. Hari, P.; Jadhav, V.; Rao, B.S. CTU Partition for Intra-Mode HEVC using Convolutional Neural Network. In Proceedings of the 2022 IEEE International Symposium on Smart Electronic Systems (iSES), Warangal, India, 18–22 December 2022; pp. 548–551.
13. Bakkouri, S.; Elyousfi, A. Machine learning-based fast CU size decision algorithm for 3D-HEVC inter-coding. *J. Real-Time Image Process.* **2021**, *18*, 983–995. [[CrossRef](#)]
14. Fu, C.H.; Chen, H.; Chan, Y.L.; Tsang, S.H.; Hong, H.; Zhu, X. Fast depth intra coding based on decision tree in 3D-HEVC. *IEEE Access* **2019**, *7*, 173138–173147. [[CrossRef](#)]
15. Zhang, Q.; Zhao, Y.; Jiang, B.; Huang, L.; Wei, T. Fast CU partition decision method based on texture characteristics for H. 266/VVC. *IEEE Access* **2020**, *8*, 203516–203524. [[CrossRef](#)]
16. Tang, N.; Cao, J.; Liang, F.; Wang, J.; Liu, H.; Wang, X.; Du, X. Fast CTU partition decision algorithm for VVC intra and inter coding. In Proceedings of the 2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Bangkok, Thailand, 11–14 November 2019; pp. 361–364.
17. Fan, Y.; Sun, H.; Katto, J.; Ming'E, J. A fast QTMT partition decision strategy for VVC intra prediction. *IEEE Access* **2020**, *8*, 107900–107911. [[CrossRef](#)]
18. Wang, Y.; Dai, P.; Zhao, J.; Zhang, Q. Fast CU Partition Decision Algorithm for VVC Intra Coding Using an MET-CNN. *Electronics* **2022**, *11*, 3090. [[CrossRef](#)]
19. Ni, C.T.; Lin, S.H.; Chen, P.Y.; Chu, Y.T. High Efficiency Intra CU Partition and Mode Decision Method for VVC. *IEEE Access* **2022**, *10*, 77759–77771. [[CrossRef](#)]
20. Pan, Z.; Zhang, P.; Peng, B.; Ling, N.; Lei, J. A CNN-based fast inter coding method for VVC. *IEEE Signal Process. Lett.* **2021**, *28*, 1260–1264. [[CrossRef](#)]
21. Li, Y.; Yang, G.; Song, Y.; Zhang, H.; Ding, X.; Zhang, D. Early intra CU size decision for versatile video coding based on a tunable decision model. *IEEE Trans. Broadcast.* **2021**, *67*, 710–720. [[CrossRef](#)]
22. Tang, J.; Sun, S. Optimization of CU Partition Based on Texture Degree in H. 266/VVC. In Proceedings of the 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Chiang Mai, Thailand, 7–10 November 2022; pp. 402–408.
23. Jiang, W.; Ma, H.; Chen, Y. Gradient based fast mode decision algorithm for intra prediction in HEVC. In Proceedings of the 2012 2nd international conference on consumer electronics, communications and networks (CECNet), Yichang, China, 21–23 April 2012; pp. 1836–1840.
24. Zhang, Y.; Han, X.; Zhang, H.; Zhao, L. Edge detection algorithm of image fusion based on improved Sobel operator. In Proceedings of the 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 3–5 October 2017; pp. 457–461.
25. Li, Y.; Liu, Z.; Ji, X.; Wang, D. CNN based CU partition mode decision algorithm for HEVC inter coding. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 993–997.
26. Khan, M.U.K.; Shafique, M.; Henkel, J. An adaptive complexity reduction scheme with fast prediction unit decision for HEVC intra encoding. In Proceedings of the 2013 IEEE International Conference on Image Processing, Melbourne, VIC, Australia, 15–18 September 2013; pp. 1578–1582.
27. Zhang, Y.; Li, Z.; Li, B. Gradient-based fast decision for intra prediction in HEVC. In Proceedings of the 2012 Visual Communications and Image Processing, San Diego, CA, USA, 27–30 November 2012; pp. 1–6.
28. Zhang, Y.; Wang, G.; Tian, R.; Xu, M.; Kuo, C.J. Texture-classification accelerated CNN scheme for fast intra CU partition in HEVC. In Proceedings of the 2019 Data Compression Conference (DCC), Snowbird, UT, USA, 26–29 March 2019; pp. 241–249.
29. Li, T.; Xu, M.; Tang, R.; Chen, Y.; Xing, Q. DeepQTMT: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC. *IEEE Trans. Image Process.* **2021**, *30*, 5377–5390. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.