Journal of Computers Vol. 33 No. 4, August 2022, pp. 57-67
doi: 10.53106/199115992022083304005

A Complexity-Reducing HEVC Intra-Mode Method Based on VGGNet

Li-Ming Qin[1], Zhong-Jie Zhu[2*], Yong-Qiang Bai[2], Guang-Long Liao[2], Ting-Na Liu[2]

[1] Department of Physics, Zhengzhou University, Zhengzhou 450001, China
1004318975@qq.com
[2] Ningbo Key Lab of DSP, Zhejiang Wanli University, Ningbo 315000, China
zhongjiezhu@hotmail.com, byq-163@163.com, 623459525@qq.com, 2411828375@qq.com

Received 4 September 2021; Revised 12 January 2022; Accepted 13 February 2022

Abstract. High-efficiency video coding (HEVC) has improved the coding performance by 50% compared with the previous H.264 coding standard. However, it has also introduced an extremely high coding complexity. The quad-tree partition used by the coding unit (CU) is one of the key factors leading to the increase in complexity. Therefore, this paper proposes a CU partition method based on a convolutional neural network (CNN). Aiming at the complex recursive calculation of CU partition, an improved VGGNet network structure is proposed to replace the brute-force search strategy, which effectively reduces the computational complexity of intra frame coding. Finally, to enhance the effectiveness of the network model in this paper, the feature pyramid network is added to the CNN model to improve the accuracy of feature extraction. The experimental results show that the proposed method can reduce the intra coding time by 59.71% while maintaining the coding performance.

Keywords: video coding, intra prediction, deep learning, convolutional neural network

1 Introduction

Video services have been widely used with the development of visual sensing technology and multimedia technology but have also put forward higher requirements for the encoding and transmission of video information. At present, high-efficiency video coding (HEVC) [1] is the most widely used video coding standard. It has greatly improved the coding performance through core technologies, such as flexible quad-tree partition structures, diversified prediction unit (PU) and transform unit (TU), and 35 different intra prediction modes [2-4]. But these technologies also lead to an increase of approximately 253% in the entire coding complexity compared with that of H.264 [5], and make it difficult to meet the real-time video coding requirements of many low-end devices (such as smartphones, drones, and digital cameras). Therefore, how to reduce the coding complexity without affecting the coding performance has become a current research hotspot.

The quad-tree partition structure can effectively divide coding unit (CU), but it needs to rely on brute force to perform a recursive rate-distortion optimization (RDO) search and traverse all depths to obtain an optimal CU partition result. It is the key factor leading to the increase of intra coding complexity. To solve this problem, some traditional exploratory methods have emerged in the past few years [6-10]. These methods summarize the laws and intermediate characteristics through mathematical statistics, so as to skip the unnecessary recursive search process. However, these methods improve the coding speed, but they cannot guarantee high compression efficiency. In recent years, with the development of deep learning, learning-based methods are proposed to predict CU partitions and reduce the coding complexity [11-17]. These methods automatically extract features and train them through a convolutional network model, then the trained model can directly predict the final CU partition result and greatly reduce the coding time. But the depth of the convolutional network structure adopted by these methods is too shallow and has a great impact on the coding performance, which cannot guarantee video encoding quality while saving time.

In this paper, we propose an improved network model based on VGGNet [18] to predict the CU partition, which can reduce the complexity of intra-frame coding while ensuring the coding performance. The quality of the final prediction result is directly determined by the selection of the convolutional network model determines. Therefore, the VGGNet model with excellent feature extraction performance and higher scalability is chosen as the basis in this paper, and then a method is designed to quickly predict the CU partition. Classic CNN models, such as LeNet-5 [19] and AlexNet [20], have network depths that are too shallow, while the network structures of

* Corresponding Author

Inception V3 [21] and ResNet [22] are too complex. They cannot fit the CU partition prediction in this paper and lack the potential for network deepening. The structure of the entire network of VGGNet is very simple, and the balance between the network depth and performance is achieved through repeated stacking. Finally, to adapt to the CU partitions of different-resolution video sequences, a pyramid network structure for feature enhancement is added to the network in this paper. The final experimental results show that the method proposed in this paper has excellent performance in reducing complexity and ensuring coding performance.

In brief, the contributions that we have made in this paper can be summarized as follows:

(1) We propose a network structure based on a convolutional neural network model, which can automatically extract relevant features for learning, thereby replacing the RDO process in traditional CU partitioning.

(2) We select VGGNet with moderate network depth and excellent performance as the basis for improvement, which can not only save coding time, but also can ensure coding quality.

(3) We introduce a feature pyramid network structure into the network structure to enhance the accuracy of feature extraction, which can make it adaptable to video coding at different resolutions.

The remainder of this paper is organized as follows. Section 2 provides a brief overview of related works of HEVC coding unit partitioning. Then the proposed network structure is described in detail in Section 3. Section 4 presents the implementation process and experimental results of our model. Finally, we conclude this paper in Section 5.

2 The Related Work

At present, the improved methods for CU partitioning can be generally classified into traditional methods and learning-based methods. Traditional methods usually simplify the recursive RDO search by skipping unnecessary partition processes to save coding time. Among these techniques, Xiong et al. proposed a fast CU partition method based on pyramid motion divergence by studying the implicit relationship between motion divergence and the rate distortion (RD) cost [6]. Kim et al. proposed an adaptive CU depth decision-making scheme based on key points [7]. Shen et al. proposed making CU size decisions by using important and computationally friendly features, and a Bayesian decision rule was defined to help accurately and quickly select the CU size by minimizing the Bayesian risk at the same time [8]. In addition, based on the temporal and spatial characteristics of the coded image, Wang et al. introduced a pruning strategy based on texture complexity and motion characteristics. Then, the motion information of adjacent coded blocks is further used to quickly determine the reference image, thereby reducing the coding time [9]. Liu et al. proposed an adaptive decision method for the prediction mode based on texture complexity and image orientation [10]. These traditional methods have improved the efficiency of CU partitioning to a certain extent but have also had considerable impacts on the coding performance with limited coding time savings.

In recent years, fast learning-based partition methods have been proposed. Such methods extract and learn advanced features related to CU partitioning through the database, which further improves the partition efficiency and coding performance. First, researchers proposed a method based on a support vector machine (SVM) for classification and prediction. Liu et al. proposed a three-classification structure based on an SVM by extracting some effective image features to determine the complexity of CUs, which can quickly determine the size of the CU to further reduce the computational complexity of the encoder [11]. However, the accuracy of this method is low because it still requires manual feature extraction. In contrast, a convolutional neural network (CNN) can automatically extract the required features from the original image and quickly predict the CU partition. Based on the most classic LeNet-5 network model, Ting et al. improved the intra-frame mode decision problem as a classification problem and reduced the computational complexity of RDO to a tolerable limit [12]. This is a good idea, but the network model structure it adopts is too simple to learn more complex features. To this end, Liu et al. developed a fast algorithm based on CNN that reduces the complexity of the encoder by reducing the CU partition mode in each coding tree unit (CTU) and then performs the RDO processing [13]. In addition, Feng et al. proposed a fast CTU depth decision algorithm based on texture features and convolutional neural network classification technology [14]. This method can skip the RD cost outside of the predicted depth range but still needs to calculate the RD cost within the divided depth range. Based on this, Kim et al. designed a convolutional network model with better performance, which can quickly predict the current CTU depth and skip the complex RDO process [15]. It can save encoding time, but this method has a greater impact on the encoding quality. Subsequently, Xu et al. proposed a hierarchical CNN structure to predict the intra-mode CU partition and introduced the long- and short-term memory (LSTM) network to realize the inter-mode CU partition by learning the time correlation of the CU partition [16]. This method can skip the complex RD cost to reduce the coding complexity, but its network depth is too shallow, and the learning ability is limited, so it is not able to cope with the complicated CU partition process. In the later works, Wang et al. designed a single-stage decision network (OSDN) model to predict the CU partition results and 35 different intra-frame prediction modes [17]. Compared with the previous works, it saves more coding time. But the training time of the model is too long, and the encoding quality is still needs to be improved.

Among these methods, machine learning-based methods rely too much on manually extracted features and cannot accurately extract the features required for model training. By contrast, a CNN-based network model for the fast prediction of CUs was designed in [12-17], because deep learning-based methods can automatically extract features. However, these CNN-based network structures are relatively simple and offer a limited improvement in the coding performance; they also cannot be migrated to different coding frameworks. But the VGGNet model proposed in this paper performs better in improving the accuracy and reducing the amount of parameters, and it has better generalization abilities in network migration. It is not only suitable for the HM16.5 coding platform used in this paper but also well-adapted to other coding platforms.

3 The Proposed Method

This paper first analyzes the partition complexity of CUs and then proposes a CNN structure based on VGGNet to predict the partition structure of CUs. Finally, a fast decision model based on VGGNet is proposed. The CNN model proposed in this paper is embedded in HM16.5, and the proposed CNN model is used to replace the brute-force RDO process in the traditional way to greatly reduce the coding complexity. The flowchart is shown in Fig. 1.

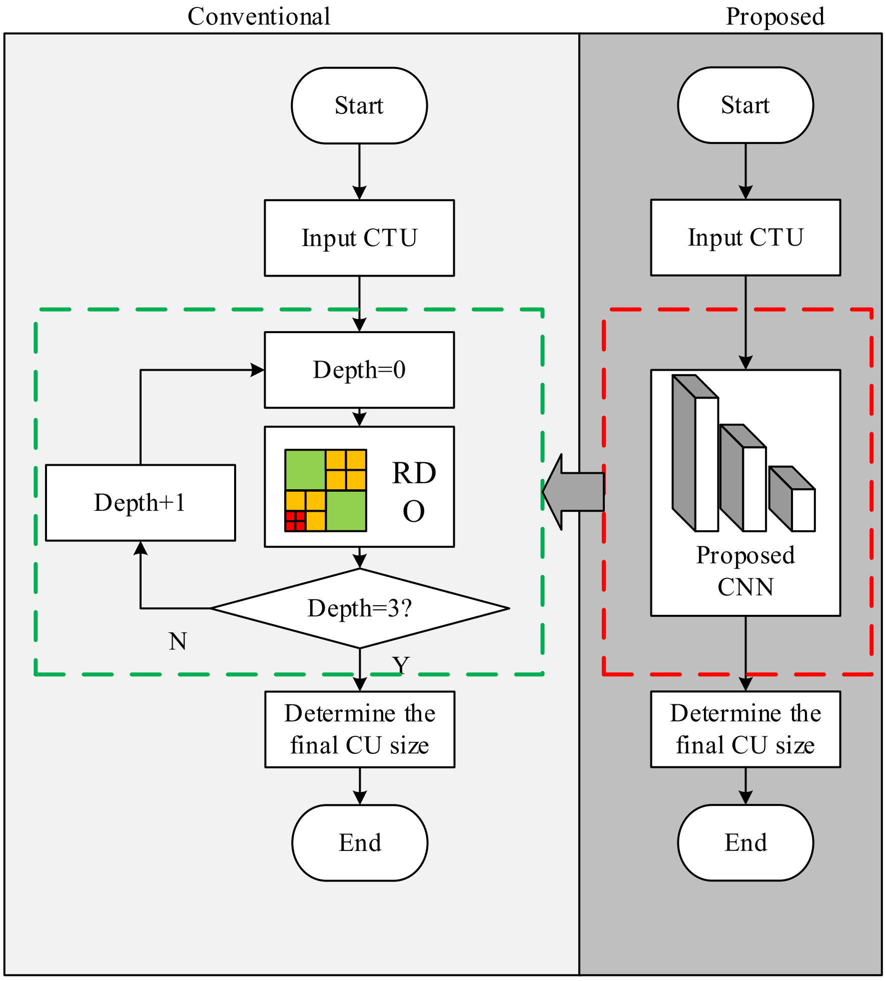


Fig. 1. Comparison between the conventional flowchart and proposed flowchart

3.1 Proposed Network Structure

In the traditional method, one CTU recursively searches 4 quad-tree depth levels ranging from 64×64 to 8×8. For each depth of CUs, there are split (=1) and no split (=0) labels, and the current depth of CUs is determined by minimizing the RD cost. To simplify the training model, this paper expresses the split labels between every two depths as a level of prediction, including three different levels of prediction. Then, a CNN model based on the VGGNet is designed to predict the optimal partition of CUs in this paper. The overall structure of the proposed model is shown in Fig. 2. First, the input CTU of the network only includes one Y channel because only the brightness information in the video sequence is used in this paper. The input CTU size is 64×64, and the preprocessing operation of averaging is performed on it. Specifically, the input CTU is subtracted from the average intensity value in each branch to reduce the variation in the input CTU samples. The preprocessed CTU is separately input into three parallel branches B_j (j=1, 2, 3), and finally, the CU partition results of the three different levels are output.

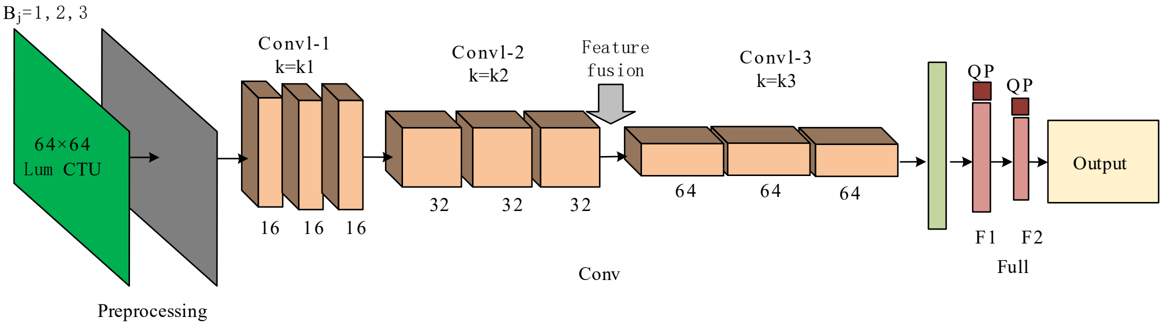


Fig. 2. Improved VGGNet CNN structure

To extract the features related to the CU partition, the convolutional layer uses VGGNet as the basis for improvement. Its greatest unique feature is that it continuously deepens the network structure to improve performance through repeated stacking. This method of stacking smaller convolution kernels can achieve the effect of convolutional layers with larger receptive fields. Since the activation function is used once after each convolution, it is equivalent to performing multiple nonlinear transformations in the calculation process, which can greatly enhance the CNN's ability to extract features. In the convolutional layer structure of this paper, the preprocessed CTU is input into three branches. Each branch includes nine convolutional layers, which are stacked by convolutional layers of different sizes. In each branch, the CU partition results of the three levels are separately predicted by changing the size of the convolutional layer. To ensure that the receptive fields of each pixel in the output feature map do not overlap, the convolutional layer sets the step size to the width of the corresponding kernel. For example, the first branch, B_1, contains convolutional layer kernel sizes k_1, k_2, and k_3 in the network structure, which are 8×8, 4×4 and 2×2, respectively. Finally, a feature map with a size of 1×1 is output. Due to the non-overlapping property of convolution, the output 1×1 feature map represents the 64×64 global feature in the original image. Similarly, the size of the output feature map is 2×2 and 4×4 by setting the size of the convolution kernel of the B_2 and B_3 branches, respectively. Each pixel has non-overlapping 32×32 and 16×16 receptive fields in the original input image. Therefore, these feature maps can be regarded as a description of the four sub-CUs of the upper-level CU. For CUs of different levels, the kernel sizes in the network are different to ensure that the pixels in feature maps extracted by $\text{Conv}_{1\text{-}3}$ have receptive fields corresponding to the sub-CUs.

Table 1. The sizes of the convolution kernels on different branches

Branch	$\text{Conv}_{1\text{-}1}$	$\text{Conv}_{1\text{-}2}$	$\text{Conv}_{1\text{-}3}$
1	k_1=8	k_2=4	k_3=2
2	k_1=4	k_2=4	k_3=2
3	k_1=4	k_2=2	k_3=2

Finally, the fusion feature maps extracted by the convolutional layer are converted into a one-dimensional vector and input to the fully connected layer. The fully connected layer includes two hidden layers and an output layer, and the hidden layer randomly removes the spliced feature vectors by setting different removal rates. The output layers of different branches output different numbers of feature maps corresponding to the number of labels of the three different CU classification levels. Finally, the output layer is activated through the *softmax* function, and the three branches output the three levels of CU partition probabilities. An early termination mechanism is also introduced in this paper. When the prediction result of the first branch is 0, the full connection of the second and third branches can be terminated. Similarly, the full connection of the third branch can be terminated when the prediction result of the first branch is 1 and that of the second branch is 0. In addition, since the quantization parameter (QP) also has an impact on the partition of CUs, the QP is added as an external feature to the feature vector. The CNN can better adapt to different QPs when predicting the CU partitions. The larger the QP is, the greater the tendency to use a larger CU.

3.2 Feature Fusion Network

The feature pyramid network (FPN) [23] can fuse low-level, high-resolution features and highly semantic information of high-level features and can effectively solve multiscale problems in model training and reduce the loss of details in the feature extraction process. Therefore, a network structure similar to the FPN is added in this paper, and its basic structure is shown in Fig. 3. The whole structure includes a bottom-up and a top-down process, where the bottom-up process represents feature maps of different scales extracted from the convolutional layer and the top-down process is to upsample the high-level features to their size equal to the low-level features. In addition, a 1×1 convolution is used to change the number of low-level feature maps to fuse it with high-level features. A total of nine convolutional layers are included in the network of this paper, and then we test the fusion of the extracted feature maps of different convolutional layers. The results show that fusing the extracted features of the sixth convolutional layer and the ninth convolutional layer can effectively improve the accuracy of the features without affecting the training time. Specifically, the feature extracted by the ninth convolutional layer is upsampled so that its size is consistent with the feature size of the sixth convolutional layer. At the same time, the sixth convolutional layer features are changed by convolution to make the number of channels the same as the ninth convolutional layer features. Finally, the fused features are input into the fully connected layer for prediction. In short, the feature pyramid network added to the CNN model of this paper can effectively improve the accuracy of the feature map and thus better adapt to the CU partition of video sequences of different resolutions.

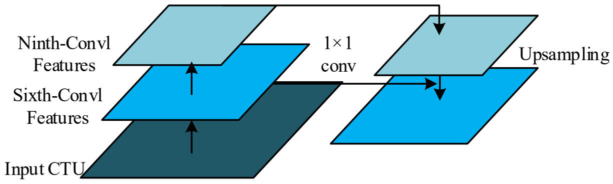


Fig. 3. Feature fusion network

3.3 Loss Function

For the loss function of the training model, the cross-entropy is often used to describe the difference between the predicted label and the correct value of the distribution. The greater the cross-entropy is, the greater the difference, and vice versa. Therefore, the sum of the cross-entropy is used as the loss function for training the VGGNet model designed in this paper. Supposing the labels of the predicted output are defined as, $\{\hat{y}_1(i)\}_{i=1}^{1}$, $\{\hat{y}_2(i)\}_{i=1}^{4}$, and $\{\hat{y}_3(i)\}_{i=1}^{16}$, the true values are defined as $\{y_1(i)\}_{i=1}^{1}$, $\{y_2(i)\}_{i=1}^{4}$, and $\{y_3(i)\}_{i=1}^{16}$, respectively. The overall number of samples is set to N, and for a single sample n, the loss function L_n is calculated as follows:

$$L_n = H(y_1(i), \hat{y}_1(i)) + \sum_{i=1}^{4} H(y_2(i), y_2(i)) + \sum_{i=1}^{16} H(y_3(i), y_3(i)) \ , \tag{1}$$

where $H(*)$ is the cross-entropy operator between the real value and the predicted label. Finally, the predicted CU partition is more accurate by minimizing the loss function L of the training model:

$$L = \frac{1}{N} \sum_{n=1}^{N} L_n \ . \tag{2}$$

4 Experimental Results

4.1 Experiment Setup

The experimental configuration and evaluation criteria of this paper are as follows: The experimental platform is HEVC reference software HM 16.5, and the hardware configuration is an Intel(R) Xeon (R) E5-1603 CPU @ 2.8 GHz with 8 GB of memory, an NVIDIA Quadro K600 GPU and a Linux 64-bit operating system. The database used for training is the database in [16], which includes 2000 images of different resolutions. The test QP value includes the four values of 22, 27, 32, and 37, the encoder configuration file is *encoder_intra_main.cfg* [24], and the test video sequence is the JCT-VC standard test set [25]. When training the model, all trainable parameters of the network model in this paper are initialized randomly and obey the truncated normal distribution with a mean value of 0 and a standard deviation of 0.1. The batch size N used for training is 64, and the initial learning rate is set to 0.01, which decreases by 1% every 2000 iterations for a total of 1 million iterations. At the same time, the evaluation criteria adopt ΔT, BD-BR and BD-PSNR: ΔT represents the time saved in encoding compared to the original HM and measures the degree of reduction in complexity, while BD-BR and BD-PSNR are used to evaluate the RD performance and represent the average bit-rate difference and average peak signal-to-noise ratio difference of the encoding, respectively. The smaller the increase in BD-BR is, the smaller the decrease in BD-PSNR, which represents a smaller RD performance loss.

4.2 Experimental Results

To verify the effectiveness and stability of the proposed method, we test video sequences with different resolutions of A, B, C, D and E in this paper. The resolutions are 2560×1600, 1920×1080, 832×480, 416×240, 1280×720, respectively. It is compared and analyzed comprehensively with excellent methods such as Reference [11] and Reference [16]. The detailed performance test results and overall performance comparison results for each sequence of JCT-VT are shown in Table 2, Table 3, Table 4, Table 5, Table 6, and Table 7. As shown in these tables, the performance of proposed method in this paper is best either saving time or RD performance.

Table 2. Comparison results for the A-Sequence of JCT-VT

					ΔT (%)			
class	sequence	Appr.	BD-BR (%)	BD-PSNR (dB)	QP= 22	QP= 27	QP= 32	QP= 37
A	PeopleOnStreet	[11]	9.777	-0.493	-49.12	-47.63	-36.79	-32.81
		[16]	2.632	-0.139	-54.71	-55.79	-60.82	-60.05
		Our	**2.148**	**-0.116**	**-55.91**	**-56.51**	**-60.91**	**-61.52**
	Traffic	[11]	6.561	-0.305	-34.11	-22.36	-18.63	-31.38
		[16]	2.776	-0.135	-59.71	-62.76	-69.06	-69.74
		Our	**2.355**	**-0.115**	**-60.42**	**-64.48**	**-68.16**	**-70.81**

Table 3. Comparison results for the B-Sequence of JCT-VT

					ΔT (%)			
class	sequence	Appr.	BD-BR (%)	BD-PSNR (dB)	QP= 22	QP= 27	QP= 32	QP= 37
B	BasketballDrive	[11]	9.073	-0.245	-45.53	-34.13	-40.05	-44.89
		[16]	4.419	-0.125	-56.87	-66.68	-73.41	-76.07
		Our	3.558	-0.101	-56.48	-67.61	-72.87	-77.15
	BQTerrace	[11]	6.767	-0.296	-64.32	-60.09	-58.57	-34.48
		[16]	1.877	-0.08	-47.67	-53.97	-59.78	-62.57
		Our	1.654	-0.078	-46.57	-55.9	-59.33	-63.48
	Cactus	[11]	7.793	-0.248	-35.37	-37.83	-42.61	-49.23
		[16]	2.438	-0.082	-53.82	-61.19	-67.83	-71.24
		Our	2.106	-0.071	-55.5	-61.57	-67.5	-71.65
	Kimono	[11]	5.362	-0.172	-31.51	-40.21	-48.88	-61.58
		[16]	2.562	-0.085	-78.12	-81.92	-84.67	-84.8
		Our	2.098	-0.08	-80.66	-81.71	-84.4	-84.87
	ParkScene	[11]	3.78	-0.15	-38.69	-41.79	-58.98	-62.92
		[16]	2.129	-0.088	-58.77	-63.85	-70.04	-72.3
		Our	1.921	-0.08	-58.51	-63.3	-69.43	-73.25

Table 4. Comparison results for the C-Sequence of JCT-VT

					ΔT (%)			
class	sequence	Appr.	BD-BR (%)	BD-PSNR (dB)	QP= 22	QP= 27	QP= 32	QP= 37
C	BasketballDrill	[11]	9.968	-0.439	-43.65	-55.86	-46.66	-60.53
		[16]	2.944	-0.137	-39.35	-45.42	-56.63	-62.83
		Our	**2.723**	**-0.118**	**-38.63**	**-45.8**	**-55.8**	**-63.83**
	BQMall	[11]	9.816	-0.489	-49.62	-39.94	-34.52	-35.12
		[16]	2.384	-0.126	-46.98	-51.92	-57.89	-60.71
		Our	**1.866**	**-0.099**	**-47.03**	**-52.74**	**-57.8**	**-61.32**
	PartdScene	[11]	7.503	-0.469	-60.84	-46.82	-30.5	-24.88
		[16]	0.707	-0.047	-38.24	-37.9	-40.37	-44.09
		Our	**0.651**	**-0.044**	**-37.91**	**-37.95**	**-40.21**	**-44.14**
	RaceHorses	[11]	7.36	-0.38	-43.46	-37.13	-40.49	-48.28
		[16]	2.055	-0.113	-53.07	-55.6	-58.89	-65.87
		Our	**1.871**	**-0.102**	**-53.09**	**-55.24**	-58.97	**-65.89**

Table 5. Comparison results for the D-Sequence of JCT-VT

					ΔT (%)			
class	sequence	Appr.	BD-BR (%)	BD-PSNR (dB)	QP= 22	QP= 27	QP= 32	QP= 37
D	BasketballPass	[11]	10.214	-0.546	-40.69	-39.03	-36.46	-34.69
		[16]	1.96	-0.113	-54.14	-56.58	-61.81	-63.43
		Our	**1.735**	**-0.1**	**-54.04**	**-57.17**	**-59.55**	**-63.41**
	BlowingBubbles	[11]	6.328	-0.547	-54.13	-39.45	-24.73	-20.81
		[16]	0.678	-0.043	-33.53	-34.31	-37.13	-41.63
		Our	**0.578**	**-0.0465**	**-34.12**	**-34.32**	**-37.33**	**-40.53**

	BQSquare	[11]	12.482	-0.877	-58.45	-59.4	-57.99	-44.86
		[16]	0.869	-0.065	-39.31	-43.86	-45.4	-46.43
		Our	**0.813**	**-0.06**	**-38.4**	**-43.11**	**-45.23**	**-46.63**
	RaceHorses	[11]	8.999	-0.487	-40.12	-37.18	-38.54	-36.07
		[16]	1.507	-0.087	-47.43	-50.5	-55.6	-57.92
		Our	**1.369**	**-0.068**	**-47.14**	**-49.85**	**-53.67**	**-58.37**

Table 6. Comparison results for the E-Sequence of JCT-VT

					ΔT (%)			
class	sequence	Appr.	BD-BR (%)	BD-PSNR (dB)	QP= 22	QP= 27	QP= 32	QP= 37
E	FourPeople	[11]	9.227	-0.482	-50.52	-37.88	-25.12	-22.34
		[16]	3.45	-0.188	-55.91	-61.35	-66.93	-68.45
		Our	**2.939**	**-0.144**	**-58.18**	**-62.72**	**-66.12**	**-69.49**
	Johnny	[11]	12.332	-0.475	-55.29	-57.21	-62.98	-68.7
		[16]	4.004	-0.16	-71.98	-75.04	-77.43	-76.03
		Our	**3.22**	**-0.137**	**-72.73**	**-75.16**	**-77.4**	**-79.26**
	KristenAndSara	[11]	13.501	-0.628	-51.44	-53.94	-55.05	-60.61
		[16]	3.754	-0.181	-69.03	-71.75	-74.97	-76.17
		Our	**3.137**	**-0.153**	**-69.27**	**-69.69**	**-74.44**	**-76.84**

Table 7. Average comparison results for all sequences of JCT-VT

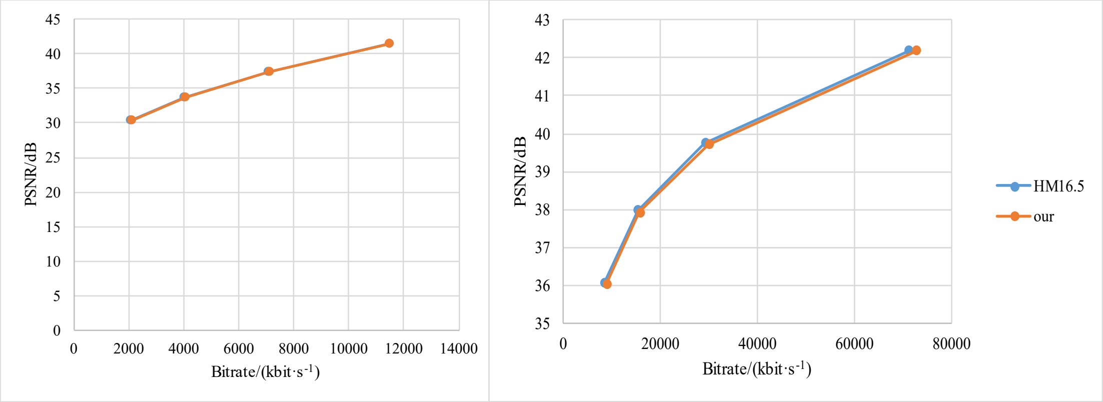
			ΔT (%)			
Appr.	BD-BR (%)	BD-PSNR (dB)	QP= 22	QP= 27	QP= 32	QP=37
[11]	8.713	-0.429	-47.05	-43.72	-41.09	-43.01
[16]	2.397	-0.111	-53.26	-57.24	-62.14	-64.46
Our	**2.041**	**-0.095**	**-53.59**	**-57.49**	**-62.62**	**-65.13**

4.3 Performance Evaluation

To evaluate the degree of complexity reduction, it can be seen from Table 3 that the method in this paper saves up to 84.87% of the time in the test sequence Kimono, and the BD-PSNR loss is only 0.08dB at this time, which shows that the method in this paper is excellent in saving time and maintaining coding performance. In addition, it can be seen from Table 2, Table 4, Table 5 and Table 6 that the time saving *ΔT* of our method is better than the results in [11] and [16]. Finally, it can be seen from Table 7 that in the case of QP = 22, 27, 32, 37, the method in this paper reduces the coding complexity *ΔT* on average by 53.53%, 57.49%, 62.62% and 65.13%, respectively, exceeding 47.05%, 43.72%, 41.09%, 43.01% in the reference [11] and 53.26%, 57.24%, 62.14%, 64.46% in [16]. It is shown that in reducing coding complexity, the method in this paper saves the most time and has the smallest loss of coding performance compared to the methods in reference [11] and reference [16].

To evaluate the RD performance loss in the encoding process, the BD-PSNR and BD-BR values under the three methods are compared in Table 7. The method of this paper produces an average BD-PSNR loss of 0.095 dB for the test sequence, which is better than the loss of 0.429 dB in reference [11] and the loss of 0.111 dB in [16]. Then the BD-BR stands for the average bit rate growth. The average growth rate of BD-BR in this paper is 2.041%, which is better than the average rates of 8.713% in reference [11] and 2.397% in reference [16]. Therefore, in the RD performance evaluation of BD-BR and BD-PSNR, the method of this paper is superior to the methods of reference [11] and reference [16]. Finally, to analyze the RD performance of the proposed method more intuitively, the RD curves of the minimum performance loss sequence BlowingBubbles and the maximum sequence BasketballDrive are given, as shown in Fig. 4. It can be seen from Fig. 4(a) that for the BlowingBubbles test sequence, the method in this paper basically coincides with the RD curve of HM16.5. As can be seen from

Fig. 4(b) that for the BasketballDrive sequence with a large performance loss, the method in this paper is basically similar to the original result in the worst case of the experimental result. Therefore, the network structure designed in this paper can effectively reduce the coding complexity without affecting the coding quality.



(a) BlowingBubbles

(a) BasketballDrive

Fig. 4. Comparison of the RD curve

To verify the stability of the method under different sequences, Formula (3) is used to calculate the variance values of BD-BR and BD-PSNR in the three methods. The smaller the variance is, the better the stability.

$$\sigma^2 = \frac{\sum (X-\mu)^2}{N} \quad . \tag{3}$$

where σ^2 is the overall variance, X is the overall variable, μ is the average, and N is the number of test sequences. The variance values of the five different types of video sequences are calculated, and the results are shown in Table 8.

Table 8. Variance results of the five sequences

sequence	Appr.	BD-BR	BD-PSNR
		Variance	Variance
A	[11]	2.586	0.008
	[16]	0.005	0.002
	Our	**0.01**	**0.001**
B	[11]	3.408	0.003
	[16]	0.809	0.000
	Our	**0.443**	**0.000**
C	[11]	1.519	0.002
	[16]	0.678	0.001
	Our	**0.545**	**0.001**
D	[11]	4.929	0.024
	[16]	0.261	0.001
	Our	**0.207**	**0.000**
E	[11]	0.005	0.024
	[16]	0.051	0.000
	Our	**0.014**	**0.000**

It can be seen from the Table 8 that the resolutions of the five different types of test sequences are 2560×1600, 1920×1080, 832×480, 416×240, and 1280×720. This shows that the method of this paper has good stability under various resolution sequences because it has the smallest variance values of BD-BR and BD-PSNR among the three methods. In summary, the method of this paper performs best in reducing the coding complexity and maintaining the coding RD performance among the three methods, and it can maintain good stability in video sequences of different resolutions.

5 Conclusion

A fast CU partition decision method based on VGGNet is proposed in this paper that can reduce the complexity of HEVC intra-mode coding. A feature fusion network is also proposed in this paper that successfully enhances the effectiveness of feature extraction, thereby ensuring the accuracy of the CU partition prediction results. Then, a fast CU partition network model with VGGNet as the core network is designed that realizes the combination of a convolutional neural network and HEVC and effectively reduces the coding complexity. The experimental results show that the proposed method reduces the coding time by 59.71% while ensuring the coding performance, which is better than the existing advanced algorithms.

The limitation of the proposed method in this paper is that the accuracy is low for some 4k, 8k and other super-resolution video sequences and some videos with higher bit depth. In addition, the training time of the model still has potential for improvement. In the future plans, our work of adding learning-based methods into coding is still a work with great advantages and potential. On the one hand, such methods can break through the limitations of traditional methods and achieve higher performance improvements. On the other hand, such methods can greatly save coding time and greatly promote the practical application of coding standards. In the future, firstly, we look forward to applying our method to related fields such as VR and 3D video in future work. Secondly, we also look forward to researching better network models in the future to make greater contributions to the development of the video field.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (No. 61671412), Zhejiang Provincial Natural Science Foundation of China (No. LY19F010002, No. LY21F010014), Natural Science Foundation of Ningbo, China (No. 202003N4323), School-level Research and Innovation Team of Zhejiang Wanli University, Fundamental Research Funds for Zhejiang Provincial Colleges and Universities, and General Scientific Research Project of Zhejiang Education Department (No. Y201941122).

References

[1] G.J. Sullivan, J.R. Ohm, W.J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, IEEE Transactions on Circuits and Systems for Video Technology 22(12)(2012) 1649-1668.

[2] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Transactions on Circuits and Systems for Video Technology 13(7)(2003) 560-576.

[3] I. Kim, J. Min, T. Lee, W.-J. Han, J.H. Park, Block partitioning structure in the HEVC standard, IEEE Transactions on Circuits and Systems for Video Technology 22(12)(2012) 1697-1706.

[4] J. Vanne, M. Viitanen, T.D. Hamalainen, A. Hallapuro, Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs, IEEE Transactions on Circuits and Systems for Video Technology 22(12)(2012) 1885-1898.

[5] G. Correa, P. Assuncao, L. Agostini, L. Cruz, Performance and computational complexity assessment of high-efficiency video encoders, IEEE Transactions on Circuits and Systems for Video Technology 22(12)(2012) 1899-1909.

[6] J. Xiong, H.L. Li, Q.B. Wu, F. Meng, A fast HEVC inter CU selection method based on pyramid motion divergence, IEEE Transactions on Multimedia 16(2)(2014) 559-564.

[7] N. Kim, S. Jeon, H. Shim, B. Jeon, S.-C. Lim, H. Ko, Adaptive keypoint-based CU depth decision for HEVC intra coding, in: Proc. 2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2016.

[8] X.L. Shen, L. Yu, J. Chen, Fast coding unit size selection for HEVC based on Bayesian decision rule, in: Proc. 2012 Picture Coding Symposium, 2012.

[9] X.J. Wang, Y.L. Xue, Fast HEVC inter prediction algorithm based on spatio-temporal block information, in: Proc. 2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, 2017.

[10]X.G. Liu, Y.B. Liu, P.C. Wang, C.-F. Lai, H.-C. Chao, An Adaptive Mode Decision Algorithm Based on Video Texture Characteristics for HEVC Intra Prediction, IEEE Transactions on Circuits and Systems for Video Technology 27(8)(2017)

1737-1748.

[11]D. Liu, X. Liu, Y. Li, Fast CU size decisions for HEVC intra frame coding based on support vector machines, in: Proc. 2016 IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, 14th International Conference on Pervasive Intelligence and Computing, 2nd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress, 2016.

[12]H.-C Ting, H.-L Fang, J.-S Wang, Complexity Reduction on HEVC Intra Mode Decision with modified LeNet-5, in: Proc. 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), 2019.

[13]Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, D. Wang, CU partition mode decision for HEVC hardwired intra encoder using convolution neural network, IEEE Transactions on Image Processing 25(11)(2016) 5088-5103.

[14]Z. Feng, P. Liu, K. Jia, K. Duan, Fast Intra CTU Depth Decision for HEVC, IEEE Access 6 (2018) 45262-45269.

[15]K. Kim, W.W. Ro, Fast CU Depth Decision for HEVC Using Neural Networks, IEEE Transactions on Circuits and Systems for Video Technology 29(5)(2019) 1462-1473.

[16]M. Xu, T.Y. Li, Z.L. Wang, X. Deng, R. Yang, Z. Guan, Reducing complexity of HEVC: a deep learning approach, IEEE Transactions on Image Processing 27(20)(2018) 5044-5059.

[17]Z.X. Wang, F. Li, Convolutional neural network based low complexity HEVC intra encoder, Multimedia Tools and Applications 80(2)(2021) 2441-2460.

[18]K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: Proc. 2015 3rd International Conference on Learning Representations, 2015.

[19]Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11)(1998) 2278-2324.

[20]A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: Proc. 2012 26th Annual Conference on Neural Information Processing Systems, 2012.

[21]C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[22]K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proc. 2016 IEEE Conference on Computer Vision & Pattern Recognition, 2016.

[23]T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature Pyramid Networks for Object Detection, in: Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[24]F. Bossen, Common test conditions and software reference configurations, in: Proc. 2011 Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 5th meeting, 2011.

[25]J.-R Ohm, G.J Sullivan, H. Schwarz, T.K. Tan, T. Wiegand, Comparison of the coding efficiency of video coding standards including high efficiency video coding (HEVC), IEEE Transactions on Circuits and Systems for Video Technology 22(12)(2012) 1669-1684.