Multitask Learning-Based Early MTT Partition Decision for Versatile Video Coding

Wu Liu, Yue Li, and Mingxing Nie(✉)

University of South China, Hengyang 421009, China
niemx@usc.edu.cn

Abstract. Versatile Video Coding (VVC) introduces a new block partition structure called Multi-Type Tree (MTT), which includes four partitioning modes: horizontal-vertical binary tree partitioning, horizontal-vertical ternary tree partitioning. This new block partition structure significantly improves compression performance, but at the same time greatly increases the computational complexity of VVC. To reduce the computational complexity of MTT in VVC inter-frame coding, a Multitask learning-Based early MTT partition decision for Versatile Video Coding is proposed. Firstly, for each Coding Unit (CU), two types of features related to the optimal MTT partitioning are extracted, namely encoding parameter features and encoding intermediate information features. Secondly, to reduce the number of neural network parameters, the horizontal or vertical partitioning in MTT is jointly learned, and lightweight neural networks are constructed to decide whether to skip the horizontal or vertical partitioning of binary or ternary trees. Experimental results show that under the Random Access (RA) configuration, the proposed method can reduce the VVC inter-frame computational complexity by an average of 27.79%, while only increasing the Bjontegaard delta bit rate (BDBR) by 1.14%.

Keywords: Versatile video coding · Multi-type tree · Multi-task learning · Block partition

1 Introduction

With the rapid development of information acquisition technology, new video formats continue to emerge, such as 4K/8K and 360° panoramic video. Although the new video formats can give viewers a better visual experience, their data volume is very large, which brings new serious challenges to the field of video compression. In order to store and transmit video data more efficiently, in July 2020, Joint Video Explore Team (JVET) launched the new generation video

This research was supported in part by the National Natural Science Foundation of China under Grant 62001209; in part by the Research Foundation of Education Bureau of Hunan Province under Grant No. 21B0424.

L. Fang et al. (Eds.): CICAI 2023, LNAI 14474, pp. 488–499, 2024.
https://doi.org/10.1007/978-981-99-9119-8_44

compression standard H.266/VVC [1]. Compared with the previous generation of video compression standard H.265/High Efficiency Video Coding (HEVC), the coding efficiency has been improved by about 40% while maintaining the same subjective video quality [2].

Similar to the coding structure of HEVC, VVC is also encoded based on hybrid coding framework. In order to further improve coding efficiency, VVC introduced many new coding techniques [3,4]. For example, in order to support more flexible block partitioning shapes, VVC used a nested multi-type tree based on Quadtree with nested multi-type tree (QTMT), which increases the number of partition modes for each CU to six [5]: Non-partition (NT), Quadtree partition (QT), Horizontal binary tree partition (H_BT), Vertical binary tree partition (V_BT), Horizontal ternary tree partition (H_TT) and Vertical ternary tree partition (V_TT), as shown in Fig. 1. Under the RA configuration, the QTMT partitioning structure can reduce the coding rate by 8.5% [6], but it leads to about 1.7 times more computational complexity for VVC than HEVC [7]. Currently, the high complexity has become a major obstacle to deploying VVC in real-time applications on devices that require low power consumption, such as smartphones and unmanned aerial vehicles. Therefore, it is necessary to study fast QTMT decision method to reduce the complexity of VVC.

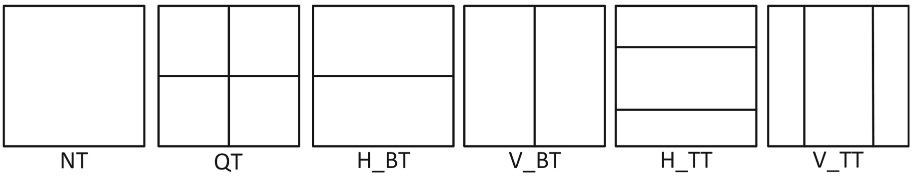

Fig. 1. Six partition modes.

In this paper, a multi-task learning-based early MTT partitioning decision method for VVC is proposed, which cleverly combines multi-task learning with the MTT module of VVC for the first time, and effectively solves the problems of a large number of model parameters and low prediction accuracy. The main contribution of this paper is as follows:

(1) Some new features related to MTT partitioning have been proposed, experimental results show that these features have good prediction effect, and the proposed method can effectively reduce the computational complexity.

(2) A lightweight neural network based on multi-task learning is proposed to reduce the computational complexity of MTT, the lightweight neural network model has fewer parameters and low training difficulty.

2 Related Work

2.1 Fast Algorithm in HEVC

The QTMT module of VVC is extended from the Quadtree module of HEVC. The existing fast algorithms in HEVC can be mainly divided into two categories:

the methods based on Machine Learning (ML) [8–12] and the methods based on encoding intermediate information [13–15]. For example, Bouaafia *et al.* [8] proposed two fast CU partitioning methods based on ML. The first is the online Support Vector Machine (SVM) fast algorithm. Another method is to design a deep convolutional neural network to predict the optimal size of each CU. Lee *et al.* [11] used characteristic information based on Sobel operator and rate distortion to determine the optimal size of each CU in advance. In the method based on intermediate information. For example, Tan *et al.* [13] predicted residual error through statistical analysis and designed a residual threshold to determine whether the CU needs further division.

2.2 Fast Algorithm in VVC

Since QTMT partitioning in VVC is more complex and flexible than QT partitioning in HEVC, the above method cannot be used directly in VVC. Fast methods in VVC also fall into two categories: the methods based on ML [16–22] and the methods based on intermediate information [23–26]. In the method based on ML, methods [16–20] is used for RA configuration inter-frame coding. For example, Pan *et al.* [16] designed a Multi-information Convolutional Neural Network (MF-CNN) model, which jointly uses multi-domain information to terminate the CU partitioning process in advance. Methods [21,22] are used for All Intra (AI) configuration intra-frame coding. For example, Tissier *et al.* [21] proposed a two-stage learning method is proposed to reduce the computational complexity of CUs in VVC encoders, including CNN and Decision Tree.

In the method using intermediate coding information, methods [23,24] is used for inter-frame coding. For example, Won *et al.* [23] proposed a fast partitioning algorithm of binary and ternary trees based on Mean Absolute Error (MAE) function, using the MAE value to compare with a threshold value to determine whether to further partition. Methods [25,26] is used for intra-frame encoding. For example, Peng *et al.* [26] sets adaptive threshold to classify CUs into simple, ordinary and complex types according to texture features, and skips the calculation of all partition modes of simple CU.

3 Background and Motivation

In the VVC encoding process, the current frame is first divided into multiple Coding Tree Units (CTUs) of the same size. Then, each CTU is divided into CU leaf nodes, and then CUs is recursively divided. Due to the addition of a variety of partitioning modes and partitioning rules, the partitioning results of a frame image become diverse. In order to obtain the best result of the current frame partitioning, it is necessary to traverse all possible partitioning cases for each CU and calculate the Rate-Distortion cost (RDcost) for each CU partitioning mode. Finally, the mode with the lowest RDcost is selected as the best CU partitioning mode. The RDcost is calculated as follows:

$$RDcost = D + \lambda \times K_m \tag{1}$$

where D is the distortion, K_m represents the number of bits of mode m, m includes the six partition modes show in Fig. 1, and λ is the Lagrange multiplier.

Although the exhaustive search method in VVC can obtain the optimal partition mode of CUs, it increases the RDcost calculation several times, which brings a sharp increase in computational complexity. Figure 2 shows an example of optimal CU partitioning in a frame of BQSquare sequence in RA configuration, where the left subgraph is a 128×128 CU partition, Only one partition mode is selected as the optimal mode for a CU. Therefore, if we can accurately predict the optimal partition mode of CUs in advance and skip the RDcost calculation of the remaining partition modes, the complexity will be reduced effectively.

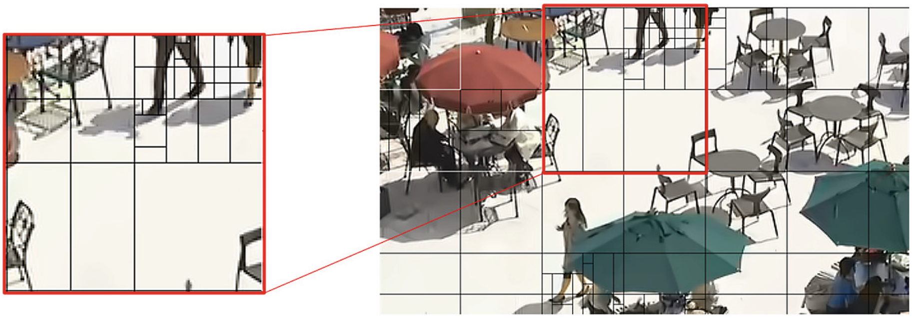

Fig. 2. A Partition Example.

4 Proposed Approach

4.1 Multi-task Learning Model

Multi-task Learning (MTL) can combine datasets from multiple tasks, and thereby alleviating the problem of data sparsity by utilizing useful information from other related learning tasks. In addition, when multiple tasks learn together, the unrelated parts of the tasks act as trace noise, and adding trace noise can improve the generalization ability of the model.

In VVC, since the binary tree partitioning of CUs in the same direction is closely related to the ternary tree partitioning, MTL can be applied to the MTT module of VVC based on this feature. Therefore, in this paper, the binary tree horizontal partitioning skip and ternary tree horizontal partitioning skip of the same CU are combined into a multi-task problem, while the vertical orientation constitutes another multi-task problem. Then, two types of multitask learning models are constructed: Horizontal Multitask Model (HMTL) and Vertical Multitask Model (VMTL). In order to reduce the number of parameters in the model, this paper employs lightweight neural network to build multi-task learning model. The specific structure of the model is shown in Fig. 3. At the input

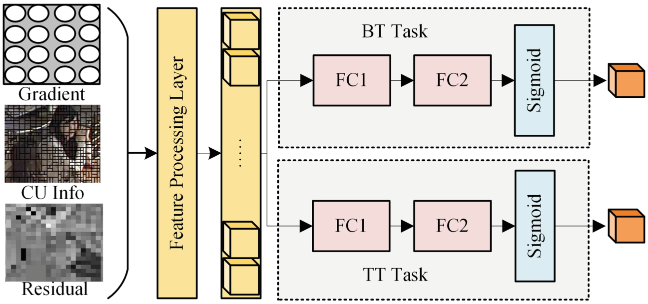

Fig. 3. Network Model.

layer, the residuals, CU information and gradients of two single tasks are input into the model for feature processing according to the calculation method in Sect. 4.2, and the obtained features are then processed using a simple two-layer fully connected (FC) network. The final output is the prediction result of two single tasks. The multi-task learning model utilizes Mean Square Error (MSE) as the loss function, which is defined as follows:

$$MSE(y, y') = \frac{\sum_{i=1}^{n}(y_i - y_i^{'})^2}{n} \tag{2}$$

where y' is the predicted value, y is the actual value, and n is the dimension.

4.2 Feature Analysis

In order to obtain the features most relevant to the optimal partitioning mode, the coding information of each CU and the corresponding optimal partitioning mode are extracted as data sets in the original VVC encoding process. In this paper, eight types of coding information are selected for correlation research. Then, according to the correlation from high to low, six kinds of encoded information are chosen as the input features for the model, Fig. 4 illustrates the analysis of thermal map characteristics. The dataset is obtained by encoding the BlowingBubbles sequence, Although the data set is extracted from only one sequence, the experimental results demonstrate that the model also exhibits good prediction performance on other sequences, which also proves that the method proposed in this paper has good generalization. The following is a detailed explanation of the selected features:

1) Maximum subblock residual variance (Max_res): In inter-frame coding, the residual value represents the changes of pixel value. However, block partition tends to divide pixels with similar changes into the same block, so the partitioning mode becomes more necessary when the variance value of subblock residuals obtained after partitioning is smaller. The residual value of pixel points is calculated as follows:

$$R_{i,j} = |P_{i,j} - O_{i,j}| \tag{3}$$

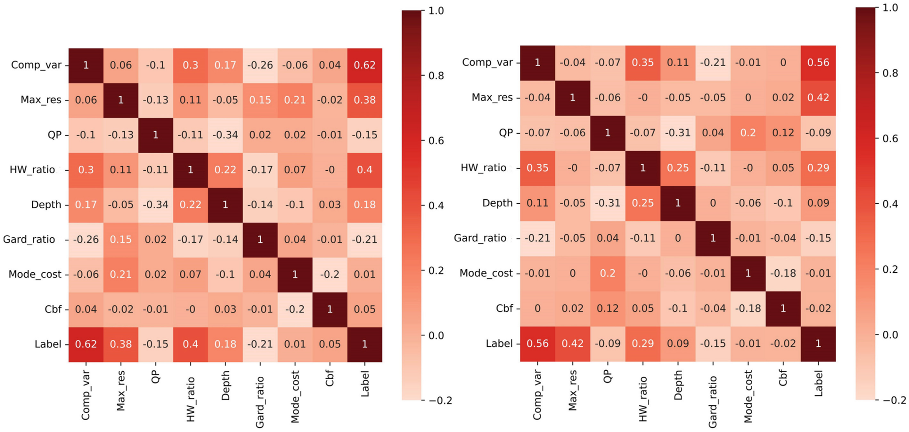

Fig. 4. The analysis of thermal map characteristics. The left is the HMTL data set and the right is the VMTL data set.

where $R_{i,j}$ represents the residual value of point (i,j) in the subblock, $P_{i,j}$ represents the predicted luma value, and $O_{i,j}$ represents the original luma value. In order to ensure that the subblocks of variance calculation are of the same size, the binary tree partitioning mode is considered to have two subblocks of equal size, and the ternary tree partitioning mode is considered to have four subblocks of equal size. Finally, the residual variance values of all CU subblocks are calculated based on the current partitioning mode, and the maximum value is normalized as a feature. The variance calculation is as follows:

$$Var = \frac{\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}(R_{i,j} - \bar{R})^2}{H \times W} \tag{4}$$

where Var is the residual variance value of subblock, H is the height of subblock, W is the width, $R_{i,j}$ is the residual value of subblock point (i, j), $\bar{R}$ is the average residual value of subblock.

2) Comparison value of variance of subblock residuals (Comp_var): Judging from only one direction will result in significant prediction errors. Therefore, within the same partition tree, we can compare the partition modes in two different directions to obtain the maximum residual variance value of subblock, and then skip the partition mode with large residual variance value of subblock through comparison. This feature is calculated as follows:

$$S = \begin{cases} 1 & Var\_H > Var\_V \\ 0 & Var\_H < Var\_V \end{cases} \quad D = \begin{cases} 1 & Var\_V > Var\_H \\ 0 & Var\_V < Var\_H \end{cases} \tag{5}$$

where Var_H represents the maximum residual variance value of current CU horizontal subblock, Var_V represents the maximum residual variance value of vertical subblock, S represents the horizontal binary tree and ternary tree features, and D represents the vertical binary tree and ternary tree features.

3) Quantization parameter (QP): QP reflects the compression level of spatial details. A smaller QP value indicates a higher retention of details, leading to a tendency to divide the data into smaller blocks.

4) Aspect ratio (HW_ratio): When the width is larger than the height, CU tends to be divided vertically. The specific calculation formula of this feature is as follows:

$$HW\_ratio = \begin{cases} \dfrac{H}{H+W} & M\epsilon\{H\_BT, H\_TT\} \\ \dfrac{W}{H+W} & M\epsilon\{V\_BT, V\_TT\} \end{cases} \tag{6}$$

where H is the height of the current CU, W is the width, and M is the partition mode of the current CU.

5) QTMT Depth (Depth): the smaller the depth is, the more likely it is to be divided, conversely, the larger the depth, the more likely it is not to be divided.

6) Horizontal and vertical gradient ratio (Gard_ratio): The gradient value can effectively represent the motion in a specific direction. In this paper, HMTL model uses G_h/G_v, VMTL model uses G_v/G_h, and the specific calculation formula of gradient is as follows:

$$G_h = \sum_{i=0}^{H-1}\sum_{j=0}^{W-1}|R_{i,j+1} - R_{i,j}| \quad G_v = \sum_{i=0}^{H-1}\sum_{j=0}^{W-1}|R_{i+1,j} - R_{i,j}| \tag{7}$$

where G_h and G_v respectively represent horizontal and vertical gradients. H is the height of the current CU, W is the width, and $R_{i,j}$ represents the residual value of the point $(i,\ j)$.

4.3 Model Training

In order to obtain the lightweight neural network structure with the best performance, we tested five different fully connected network structures. The test results are shown in Table 1,"Quantity" represents the number of parameters and "Accuracy" represents the model accuracy. The structure 6×20×20 achieved the highest prediction accuracy, and both tasks use the same structure.

Table 1. Model Architecture Testing.

Structure	6×10×50	6×30	6×20	6×30×30	6×20×20
Quantity	671	241	161	1171	581
Accuracy	80.14%	78.43%	78.13%	81.48%	83.32%

After data cleaning and redundancy removal, a total of 284,497 data sets were used to train the previously constructed multi-task model, including 110,589 data sets for the HMTL model and 173,908 data sets for the VMTL model. Train the model precision convergence about 500 times, and both models achieved an accuracy of over 80%. Figure 5 illustrates the training process.

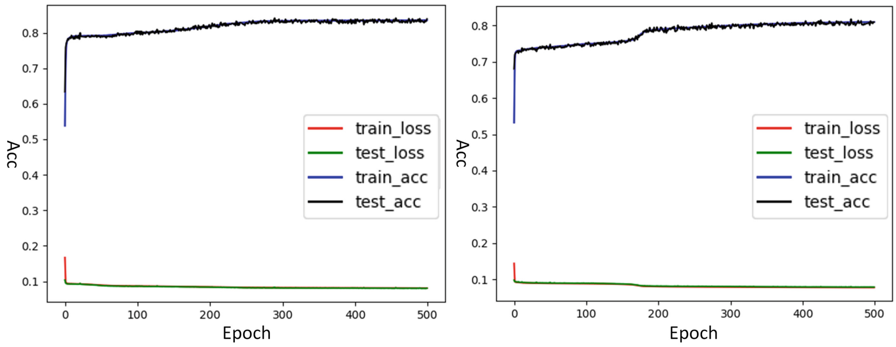

Fig. 5. Train Process. The left is the HMTL model and the right is the VMTL model, "Acc" represents the accuracy and "Epoch" represents the number of iterations

4.4 The Whole Algorithm Proposed

In this paper, the objective is to identify unnecessary partitioning modes using the algorithm, skip the calculation of RDcost, and reduce the computational complexity in the search process for the optimal CU partitioning mode. Additionally, skip flags are introduced to minimize the impact of incorrect predictions. For the binary tree or ternary tree partitioning of the same CU, if the HMTL model predicts the horizontal direction and skips it, the VMTL model will not make predictions for the vertical direction. The overall flow of the proposed algorithm is presented in Algorithm 1, Where "Skip" indicates that the calculation of the current mode is skipped ahead of time.

5 Experimental Results and Discussion

5.1 Experimental Conditions

In order to evaluate the performance of the proposed method, the latest test software VTM19.2 and the test software VTM6.0 of VVC were tested respectively with the original VTM as the anchor point. The experiment employed a total of 21 recommended general test videos, ranging from Class A1 to Class E, with RA configuration and QPs of 22, 27, 32, and 37. To mitigate the impact of incorrect predictions, the decision to skip the partitioning mode was based on a confidence level exceeding 95% in the model prediction. Therefore, the threshold (th) is set to 0.05. Encoding performance was evaluated using encoding time saving TS and BDBR [27]. Typically, better performance is indicated by greater encoding time reduction and smaller BDBR increase. To quantify coding performance, we use a performance metric similar to what is called a "Factor" in [28]. A higher Factor value denotes superior performance, The formulas are defined as follows:

$$TS = \frac{Time_{org} - Time_{pre}}{Time_{org}} \tag{8}$$

Algorithm 1: Proposed Algorithm

Input: Current mode M, Threshold value th, Binary tree horizontal skip flag B_flag, Ternary tree horizontal skip flag T_flag
initialization:B_flag=0, T_flag=0, p_bh=1, p_bv=1, p_th=1, p_tv=1
if M==H_BT **then**
    HMTL prediction—>p_bh,p_th;
    if $p\_bh < th$ **then**
        Skip and B_flag=1
    else
        B_flag=0
if M==V_BT && B_flag=0 **then**
    VMTL prediction—>p_bv,p_tv;
    if $p\_bv < th$ **then**
        Skip
if M==H_TT && $p\_th < th$ **then**
    Skip and T_flag=1;
if M==V_TT && $p\_tv < th$ && T_flag=0 **then**
    Skip;
end

$$Factor = \frac{TS}{BDBR} \tag{9}$$

where $Time_{org}$ represents the total encoding time of the original VTM encoder, and $Time_{pre}$ represents the total encoding time with the proposed algorithm added. The computer configuration for the experiment is: "11th Gen Intel(R) Core(TM) i7-11700F @ 2.50GHz, 16GB-RAM"

5.2 Coding Performance Evaluation

Table 2 shows the overall performance of the proposed method. The fast MTT partitioning method proposed in VTM19.2 can save 13.92%-41.63% encoding time, with an average saving of 27.79%. The corresponding BDBR increases by 0.56%-1.79%, with an average increase of only 1.14%. To better demonstrate the effectiveness of the algorithm proposed in this paper, a comparison is made with the methods proposed by Pan [16] and Li [24]. In order to make the experimental comparison fair, the same test platform version is used. The algorithm proposed in this paper is implemented on VTM6.0, and the comparison data with Pan's method is obtained. Similarly, Li's algorithm implemented on VTM19.2 is compared with the experimental results of the algorithm proposed in this paper. On VTM6.0, Pan's method achieves an average time reduction of 25.42% with an average BDBR increase of 2.53%. In contrast, the proposed method achieves an average time reduction of 26.68% with an average BDBR increase of 0.98%. On VTM19.2, Li's algorithm saves an average of 23.95% of time, and BDBR increases an average of 1.21%. The results indicate that, on average, the proposed method outperforms both Pan's and Li's algorithms in terms of TS and

BDBR. In other words, the method in this paper achieves a greater reduction in coding time with a smaller increase in BDBR, and get a higher Factor value.

Table 2. Experimental Result.

Class	Sequence	Pan[16]		Li[24]		Proposed(V6.0)		Proposed(V19.2)	
		BDBR	TS	BDBR	TS	BDBR	TS	BDBR	TS
A1	Campfire	2.80	30.08	1.84	33.35	1.24	30.23	1.31	30.37
	FoodMarket4	1.59	42.90	0.70	25.33	0.92	35.82	0.80	41.63
	Tango2	3.68	34.05	0.87	21.19	1.39	30.90	1.73	35.66
A2	CatRobot1	5.59	30.62	0.85	16.28	0.61	25.21	0.79	29.11
	DaylightRoad2	4.43	29.20	0.91	17.00	1.13	27.21	1.60	31.90
	ParkRunning3	1.61	21.30	0.87	27.35	0.73	28.50	0.87	29.54
B	MarketPlace	3.22	36.47	1.20	21.54	1.35	30.47	1.38	30.50
	RitualDance	2.97	31.23	1.89	29.53	1.52	30.71	1.79	26.86
	BasketballDrive	2.96	32.39	1.29	27.14	1.40	30.96	1.59	31.23
	BQTerrace	0.98	13.80	0.86	26.45	0.22	23.42	0.56	26.42
	Cactus	5.20	25.42	1.12	25.41	0.94	26.19	0.75	27.54
C	BasketballDrill	1.59	24.38	1.60	32.98	1.25	28.60	1.54	26.14
	PartyScene	1.84	14.94	1.36	33.65	0.69	24.94	0.86	25.33
	RaceHorsesC	2.23	22.55	1.92	32.63	1.05	25.87	1.29	26.02
D	BasketballPass	1.56	21.18	1.49	22.71	0.75	22.58	0.86	20.58
	BlowingBubbles	2.29	16.97	1.44	22.94	0.79	23.63	1.05	23.91
	BQSquare	0.84	9.69	1.04	18.78	0.35	15.30	0.65	13.92
	RaceHorses	2.24	20.33	1.96	26.83	1.40	25.36	1.38	23.80
E	FourPeople	1.76	25.26	0.93	15.63	0.88	23.04	0.97	26.11
	Johnny	1.69	24.92	0.63	12.65	0.63	25.77	1.20	28.95
	KristenAndSara	2.11	26.21	0.65	13.53	0.92	25.53	0.93	28.13
	Average	2.53	25.42	1.21	23.95	0.98	26.68	1.14	27.79
	Factor	10.45		19.79		27.22		24.38	

5.3 Model Performance Evaluation

In order to provide a clearer analysis of the number of model parameters, a comparison is made between the network structure in this paper and Pan's [16] as shown in Table 3. The number of model parameters used in this paper is only 1162, which is far less than Pan's model with 25.6M. In addition, the additional consumption brought by the model is tested under four different QPS. The result is to take the average of three sequences (BasketballDrill, BlowingBubbles and FourPeople). The additional time added in this paper is only 0.98% on average, while the additional time added by the Pan's model is 5.21%. Combined with the experimental test results, it is shown that the neural network model constructed in this paper can bring better prediction effect with fewer parameters.

Table 3. Model Parameter Quantity.

	Structure	Quantity	Size	QP22	QP27	QP32	QP37	Average
Proposed	Full-6*20*20	1162	4.528KB	0.73%	0.78%	1.06%	1.38%	0.98%
Pan [16]	ResNet-50	25.6M	102.4MB	4.23%	5.18%	5.91%	5.53%	5.21%

6 Conclusion

In order to reduce the computational complexity of VVC inter-frame coding, this paper proposes a Multitask learning-Based early MTT partition decision for VVC inter-frame coding. The proposed multi-task learning model is simple in structure, easy to be integrated into VVC test software, and can effectively reduce the complexity of coding computation. Experimental results show that the proposed method can achieve good coding performance on different versions of the test platform. In the latest test platform VTM19.2, the average BDBR increase is only 1.14%, and the encoding time can be reduced by 27.79%.

References

1. Bross, B.: Versatile video coding (draft 1), jvet-j1001 (2018)
2. Zhang, Q., Wang, Y., Huang, L., Jiang, B.: Fast CU partition and intra mode decision method for H. 266/VVC. IEEE Access **8**, 117539–117550 (2020)
3. An, J., Huang, H., Zhang, K., Huang, Y., Lei, S.: Quadtree plus binary tree structure integration with JEM tools. JVET-B0023, Joint Video Exploration Team (JVET) (2016)
4. Saldanha, M., Sanchez, G., Marcon, C., Agostini, L.: VVC intra-frame prediction. In: Versatile Video Coding (VVC) Machine Learning and Heuristics, pp. 23–33. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-11640-7_3
5. Bross, B., Chen, J., Liu, S., Wang, Y.K.: Versatile video coding (draft 5). Joint Video Experts Team (JVET) of Itu-T Sg 16, 3–12 (2019)
6. François, E., et al.: VVC per-tool performance evaluation compared to HEVC. In: IBC (2020)
7. Siqueira, Í., Correa, G., Grellert, M.: Rate-distortion and complexity comparison of HEVC and VVC video encoders. In: 2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS), pp. 1–4. IEEE (2020)
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 Proc. **17**, 185–196 (2020)
9. Duanmu, F., Ma, Z., Wang, Y.: Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension. IEEE J. Emerg. Sel. Top. Circuits Syst. **6**(4), 517–531 (2016)

10. Momcilovic, S., Roma, N., Sousa, L., Milentijevic, I.: Run-time machine learning for HEVC/H. 265 fast partitioning decision. In: 2015 IEEE International Symposium on Multimedia (ISM), pp. 347–350. IEEE (2015)
11. Lee, D., Jeong, J.: Fast CU size decision algorithm using machine learning for HEVC intra coding. Signal Proc. Image Commun. **62**, 33–41 (2018)
12. Bakkouri, S., Elyousfi, A.: Machine learning-based fast CU size decision algorithm for 3D-HEVC inter-coding. J. Real-Time Image Proc. **18**, 983–995 (2021)
13. Tan, H.L., Ko, C.C., Rahardja, S.: Fast coding quad-tree decisions using prediction residuals statistics for high efficiency video coding (HEVC). IEEE Trans. Broadcast. **62**(1), 128–133 (2016)
14. Shen, L., Zhang, Z., An, P.: Fast CU size decision and mode decision algorithm for HEVC intra coding. IEEE Trans. Consum. Electron. **59**(1), 207–213 (2013)
15. Huade, S., Fan, L., Huanbang, C.: A fast CU size decision algorithm based on adaptive depth selection for HEVC encoder. In: 2014 International Conference on Audio, Language and Image Processing, pp. 143–146. IEEE (2014)
16. Pan, Z., Zhang, P., Peng, B., Ling, N., Lei, J.: A CNN-based fast inter coding method for VVC. IEEE Signal Process. Lett. **28**, 1260–1264 (2021)
17. Li, J., Zhang, S., Yang, F.: Random forest accelerated CU partition for inter prediction in H. 266/VVC. In: 2022 IEEE International Conference on Multimedia and Expo (ICME), pp. 01–06. IEEE (2022)
18. Liu, Y., Abdoli, M., Guionnet, T., Guillemot, C., Roumy, A.: Light-weight CNN-based VVC inter partitioning acceleration. In: 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), pp. 1–5. IEEE (2022)
19. Amestoy, T., Mercat, A., Hamidouche, W., Menard, D., Bergeron, C.: Tunable VVC frame partitioning based on lightweight machine learning. IEEE Trans. Image Process. **29**, 1313–1328 (2019)
20. Yeo, W.H., Kim, B.G., et al.: CNN-based fast split mode decision algorithm for versatile video coding (VVC) inter prediction. J. Multimedia Inf. Syst. **8**(3), 147–158 (2021)
21. Tissier, A., Hamidouche, W., Mdalsi, S.B.D., Vanne, J., Galpin, F., Menard, D.: Machine learning based efficient QT-MTT partitioning scheme for VVC intra encoders. IEEE Trans. Circuits Syst. Video Technol. **33**, 4279–4293 (2023)
22. 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. **30**, 5377–5390 (2021)
23. Won, D.J., Moon, J.H.: Fast inter CU partitioning algorithm using MAE-based prediction accuracy functions for VVC. J. Broadcast Eng. **27**(3), 361–368 (2022)
24. Li, Y., Luo, F., Zhu, Y.: Temporal prediction model-based fast inter CU partition for versatile video coding. Sensors **22**(20), 7741 (2022)
25. Fu, T., Zhang, H., Mu, F., Chen, H.: Fast CU partitioning algorithm for H. 266/VVC intra-frame coding. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 55–60. IEEE (2019)
26. Peng, S., Peng, Z., Ren, Y., Chen, F.: Fast intra-frame coding algorithm for versatile video coding based on texture feature. In: 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 65–68. IEEE (2019)
27. Barman, N., Martini, M.G., Reznik, Y.: Revisiting Bjontegaard delta bitrate (BD-BR) computation for codec compression efficiency comparison. In: Proceedings of the 1st Mile-High Video Conference, pp. 113–114 (2022)
28. Kuang, W., Chan, Y.L., Tsang, S.H., Siu, W.C.: Fast HEVC to SCC transcoder by early CU partitioning termination and decision tree-based flexible mode decision for intra-frame coding. IEEE Access **7**, 8773–8788 (2019)