

## Fast CU Partition for VVC Using Texture Complexity Classification Convolutional Neural Network

Yue Zhang<sup>1,3,4</sup>, Pengyu Liu<sup>1,2,3,4,\*</sup>, Xiaowei Jia<sup>5</sup>, Shanji Chen<sup>2</sup>, Tianyu Liu<sup>1,3,4</sup> and Chang Liu<sup>1,3,4</sup>

<sup>1</sup>The Information Department, Beijing University of Technology, Beijing, 100124, China

<sup>2</sup>School of Physics and Electronic Information Engineering, Qinghai Minzu University, Xining, 810000, China

<sup>3</sup>Advanced Information Network Beijing Laboratory, Beijing, 100124, China

<sup>4</sup>Computational Intelligence and Intelligent Systems Beijing Key Laboratory, Beijing, 100124, China

<sup>5</sup>Department of Computer Science, University of Pittsburgh, Pittsburgh, 15260, USA

\*Corresponding Author: Pengyu Liu. Email: liupengyu@bjut.edu.cn

Received: 05 February 2022; Accepted: 18 March 2022

**Abstract:** Versatile video coding (H.266/VVC), which was newly released by the Joint Video Exploration Team (JVET), introduces quad-tree plus multi-type tree (QTMT) partition structure on the basis of quad-tree (QT) partition structure in High Efficiency Video Coding (H.265/HEVC). More complicated coding unit (CU) partitioning processes in H.266/VVC significantly improve video compression efficiency, but greatly increase the computational complexity compared. The ultra-high encoding complexity has obstructed its real-time applications. In order to solve this problem, a CU partition algorithm using convolutional neural network (CNN) is proposed in this paper to speed up the H.266/VVC CU partition process. Firstly,  $64 \times 64$  CU is divided into smooth texture CU, mildly complex texture CU and complex texture CU according to the CU texture characteristics. Second, CU texture complexity classification convolutional neural network (CUTCC-CNN) is proposed to classify CUs. Finally, according to the classification results, the encoder is guided to skip different RDO search process. And optimal CU partition results will be determined. Experimental results show that the proposed method reduces the average coding time by 32.2% with only 0.55% BD-BR loss compared with VTM 10.2.

**Keywords:** Versatile video coding (VVC); coding unit partition; convolutional neural network (CNN)

### 1 Introduction

With the continuous development of multimedia technology, video is moving towards ultra-high-definition (UHD) with high definition (HD) and high frame rate (HFR). UHD has been widely used in social media, distance education, telemedicine, video conferencing and other fields, and has gradually become the main way for people to obtain external information. The growing amount of video data has greatly increased the demand for storage space and transmission bandwidth.



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is urgent and important to explore effective and efficient ways to compress video data. In January 2013, International Telecommunication Union Telecommunication Standardization Sector (ITU-T) and International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) jointly released the High Efficiency Video Coding (H.265/HEVC). Compared with the previous standard, Advanced Video Coding (H.264/AVC), H.265/HEVC saves nearly 50% bitrate. According to the latest Ericsson mobility report in 2021, video traffic is estimated to account for 69% of all mobile data traffic, a share that is forecast to increase to 79% in 2027 [1]. With the continuous improvement of video quality and the growing video data, these previous standards are difficult to meet the needs for more efficient video compression. In this context, the Joint Video Exploration Team (JVET) established in 2015 began to develop a new video coding standard. And Versatile Video Coding (H.266/VVC) was officially released in July 2020 [2].

From some related studies, we can find that compared with the H.265/HEVC reference software HM 16.22, H.266/VVC reference software VTM 10.2 improves video encoding quality 23.0% under All Intra (AI) configuration, but the computational complexity increases 34.0 times [3–5]. It is difficult for H.266/VVC to meet the requirements of high real-time performance in practical application scenarios. In order to support more flexible CU partition shapes, the quad-tree plus multi-type tree (QTMT) partition structure, which contains Quad-tree partition (QT), Binary Tree partition (BT) and Ternary Tree partition (TT) in vertical and horizontal directions, is introduced in H.266/VVC based on QT partition structure in H.265/HEVC. The more sophisticated and more flexible CU partition not only improves the coding efficiency greatly, but also increases the intra prediction time in H.266/VVC. Hence, there is an urgent need to reduce the complexity of CU partition in H.266/VVC.

The fast CU partition decision algorithm for video coding standards has been widely studied during the past decades. Prior works on this topic mainly includes two categories: traditional methods [6–10] and deep learning methods [11–16]. The core idea is to replace or simplify the rate-distortion optimization (RDO) search process in the encoder with more efficient CU partition methods. For the traditional methods, most of them are based on the spatial correlation between the current CU and the surrounding CU or the image texture features of CUs [6–10]. Considering the inherent advantage of video including rich image information, algorithms that exploit the texture characteristics of CU and reduce the encoding complexity of CU partition have been extensively researched. Chen et al. [7] used Sobel to explore CU texture and determine the CU partition. Zhang et al. [8] proposed gray level co-occurrence matrix (GLCM) to judge CU texture direction, so as to terminate BT/TT division in horizontal or vertical direction in advance. However, considering the size of GLCM depends on the number of brightness levels rather than the image size and the calculation process is complicated, Zhang et al. [9] proposed the angular second moment (ASM) to determine the texture complexity of the image on the basis of GLCM and the CU division depth will be determined. Although these algorithms can reduce the CU partitioning complexity, the encoding quality heavily depends on the result of texture complexity analysis and the specified threshold. With the great advantages of machine learning and deep learning in image/video information processing, reducing the complexity of CU partition process through machine learning and deep learning has become a research hotspot in video coding fields [11–16]. Tang et al. [13] proposed pooling-variable Convolutional Neural Networks (pooling-variable CNN) to realize the adaptive decision of CU partition considering the existence of rectangular CU with different aspect ratios in H.266/VVC. Li et al. [14] utilized Multi-stage Exit CNN (MSE-CNN) with an early Exit mechanism to determine CU partition. However, these methods only rely on the CU partition results output by VTM and ignore the rich texture information contained in the video. Jin et al. [15] modeled QTBT partitioning as a multi-class classification problem, using CNN

to decide the QT and BT depth of  $32 \times 32$  CU. However, its QTBT partitioning structure in JVET is not suitable for the more complex QTMT partition structure in H.266/VVC.

Inspired by the above algorithms, considering high accuracy of deep learning in the texture feature extraction of image and video [11,17–22], this paper proposes a fast CU partition method for VVC using convolutional neural network, named CU texture complexity classification convolutional neural network (CUTCC-CNN). Our aim is to lead the encoder to select an appropriate RDO search process for different CU to obtain the optimal CU partition structure with less encoding time. Experimental results show that our method can significantly improve the coding efficiency with only 0.55% objective compression performance.

The remainder of this paper is organized as follows. We introduce the CU partition structure and analyze the texture complexity of H.266/VVC in Section 2. The details about our proposed method are described in Section 3. Section 4 shows the experimental results and the corresponding analysis. Finally, we conclude this paper in Section 5.

## 2 Related Work

### 2.1 CU Partition Structure in H.266/VVC

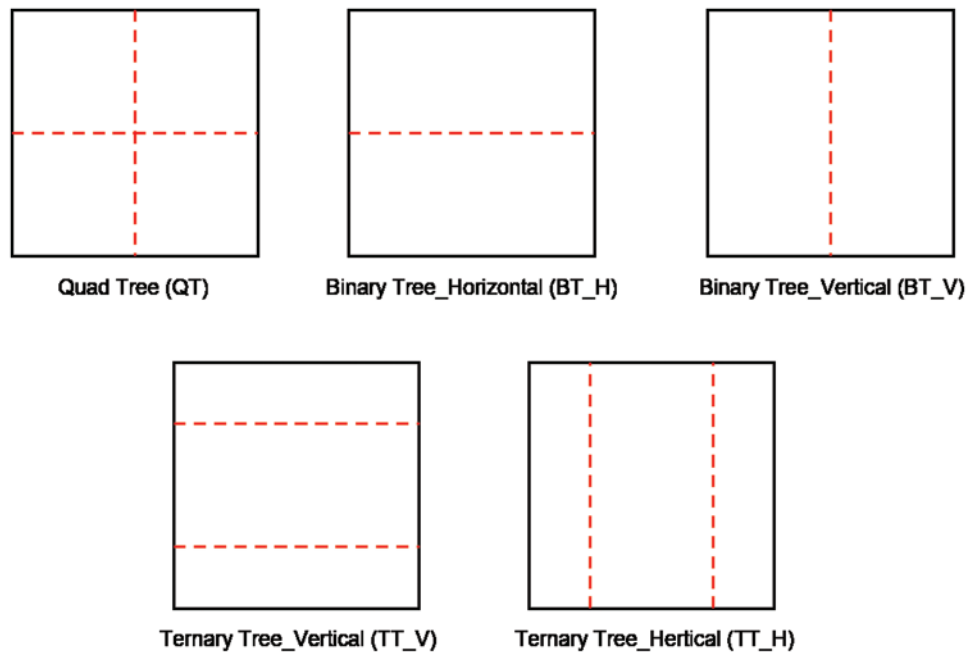
As the continuation and improvement of H.265/HEVC, H.266/VVC still adopts a block-based hybrid coding framework and divides each frame of the video into coding tree units (CTU). In order to encode videos with a higher resolution, the size of CTU expands from  $64 \times 64$  to  $128 \times 128$ , and the minimal CU size is  $4 \times 4$  in H.266/VVC. For intra-coded frames, H.266/VVC adopts independent CU partition scheme for luminance and chrominance components. This paper mainly studies the partition of luminance CU.

There are five CU partition modes in H.266/VVC, as is shown in Fig. 1. A  $128 \times 128$  CTU need to be divided into four CUs through QT. Then the RDO search process is performed to find the optimal CU partition structure, with not dividing and QT are supported for  $64 \times 64$  CUs. When QT is chosen, four  $32 \times 32$  CUs are obtained. CUs less than  $32 \times 32$  need to choose an optimal partition mode from all possible partition modes to achieve the minimal rate-distortion cost (RD cost), with not dividing, QT, BT\_H, BT\_V, TT\_H TT\_V supported. QTMT partition structure for CU partition may bring about extremely high computational complexity in H.266/VVC. Therefore, designing an appropriate method to simplify the RDO search process will reduce the encoding time.

Fig. 2 shows the CU partitioning process in H.266/VVC. We can see that after QT, BT, and TT division, rectangular CUs with aspect ratios of 1/8, 1/4, 1/2, 1, 2, 4, and 8 may be obtained. The flexible QTMT partition structure enables H.266/VVC to adapt to various complex video scenes, thus improving compression efficiency.

### 2.2 Relationship between CU Partition Results and Texture Complexity

Images in the actual video scenes often contain rich texture information. Existing studies have shown that an image is composed of areas with complex and smooth texture, and there is also a certain texture transition between complex and smooth areas [7,9,10]. From the perspective of texture characteristics of CU image, this paper classifies CU into smooth texture CU, mildly complex texture CU and complex texture CU. Different partitioning methods will be selected for the three kinds of CU to reduce unnecessary RDO search process. The detailed methods will be described in 3.3.



**Figure 1:** CU partition modes in H.266/VVC

Fig. 3 shows the CU partition result of a frame encoded by H.266/VVC for Johnny. It shows that the image texture in complex texture CUs is complex and diverse. The encoder will divide  $64 \times 64$  CUs into smaller size CUs. For mildly complex texture CUs, the encoder will end the division of  $64 \times 64$  CUs at larger size. The encoder will not divide smooth texture CUs.

This paper counts the CU information for all VVC test sequences to reveal the characteristics of CU partition in H.266/VVC, as shown in Fig. 4. H.266/VVC test sequences are encoded by the VTM 10.2 under AI configuration, and one frame is down-sampled every 10 frames. As can be seen from the graph, CUs with different sizes hold different proportions. And most CUs tend to choose not dividing. For example, not dividing CUs account for more than 20% of  $32 \times 32$  CUs. As for  $32 \times 4$  and  $4 \times 32$  CUs, the proportion of not dividing is close to 75%. Therefore, encoding time can be shortened through skipping unnecessary CU division modes in intra prediction according to the texture features of videos.

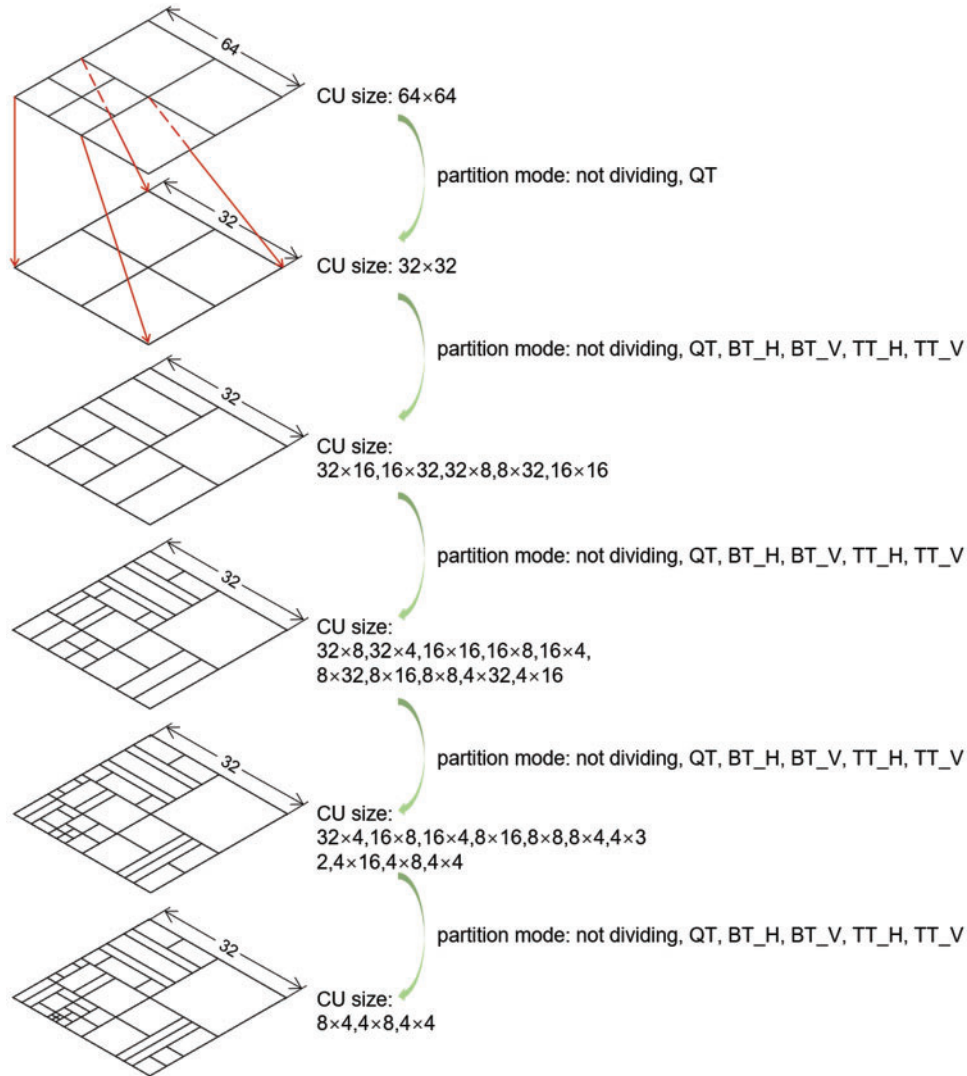
### 3 Proposed Method

In this section, our method to simplify the original RDO search process in H.266/VVC is presented. First, we introduce our CUTCC-CNN in 3.1. Then the dataset construction is given in 3.2. Finally, we will describe the process for fast CU partition based on the proposed CUTCC-CNN.

#### 3.1 CU Texture Complexity Classification CNN Structure

The structure of CUTCC-CNN is shown in Fig. 5.  $64 \times 64$  CU is fed into CUTCC-CNN which is composed of a down-sampling unit, 3 Light Convolutional units (LCU), a  $1 \times 1$  convolutional layer and a fully connected layer. Firstly,  $64 \times 64$  CU passes through a down-sampling unit, which is composed of a  $3 \times 3$  convolution layer and two branching structures to obtain a group of feature maps. Then, output feature maps of down-sampling unit flow into 3 Light Convolutional units (LCU). After

the third LCU, a  $1 \times 1$  convolutional layer and a fully connected layer is used to output CU texture complexity category. The details of down-sampling unit and LCU are as follows.

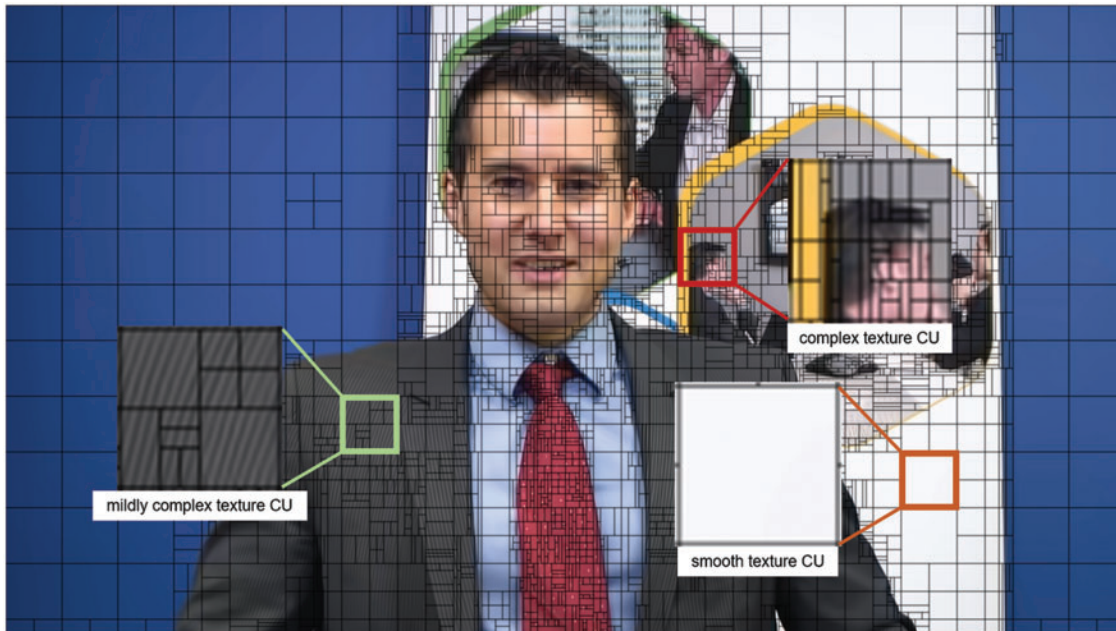


**Figure 2:** CU partition process in H.266/VVC

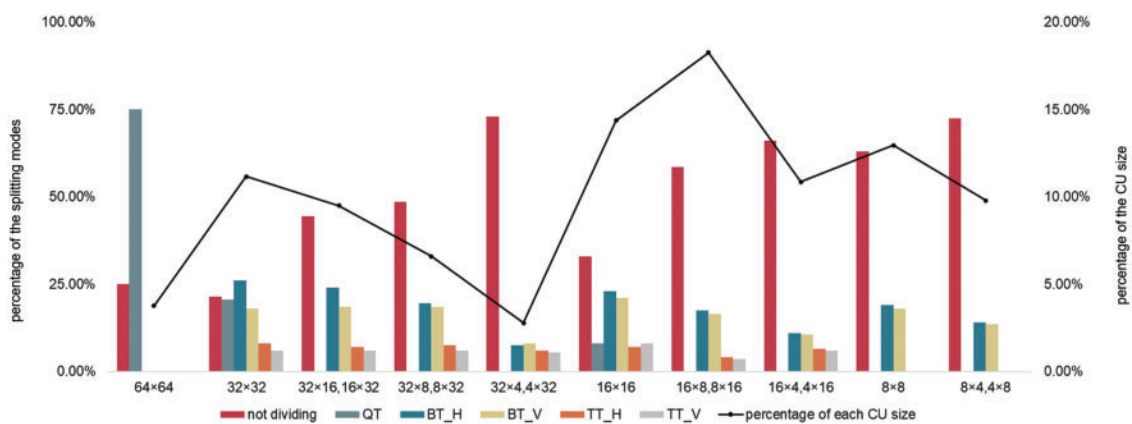
In down-sampling unit, a  $3 \times 3$  convolution layer is first used to improve the channel dimension and obtain partial shallow texture information. Secondly, the output shallow feature map will be fed into two different branches. One of the branches uses global average pooling layer to realize down-sampling operation of shallow feature map, and the other branch is composed of two  $1 \times 1$  convolution layers and a  $3 \times 3$  Depthwise Convolution [23]. The channel dimension is increased through the  $1 \times 1$  convolutional layer, the texture extraction and size reduction are conducted on the high-dimensional feature map through the  $3 \times 3$  Depthwise Convolution, and the channel dimensionality reduction is achieved through the  $1 \times 1$  convolutional layer. Finally, the feature maps of the two branches are added and fused to obtain the final feature map. The down-sampling unit merges the texture information in two different branches, which can effectively reduce the loss of texture information in the CU.



The introduction of the Depthwise Convolution can significantly improve the computational efficiency as it has a smaller number of parameters.

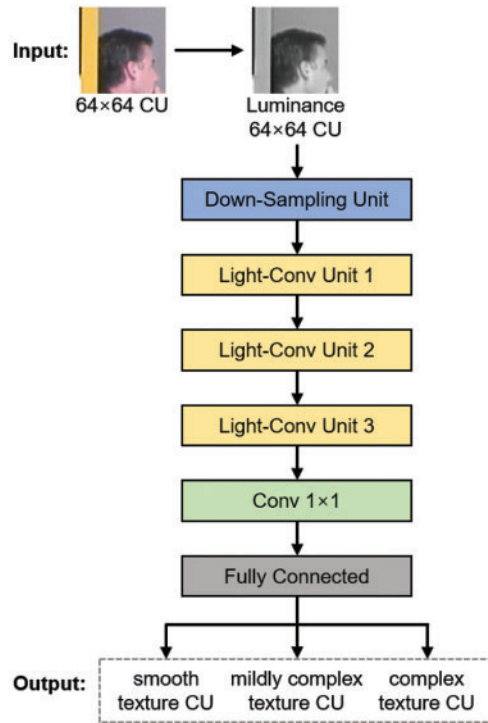


**Figure 3:** An example of CU partition results in H.266/VVC

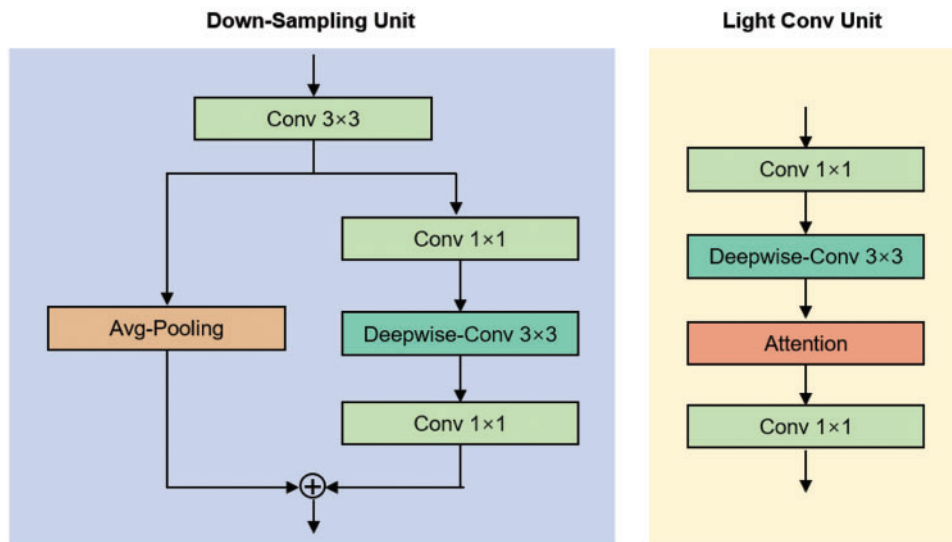


**Figure 4:** The percentage of each CU size and the corresponding splitting modes

LCU is composed of two  $1 \times 1$  convolution layers, a  $3 \times 3$  Depthwise Convolution and Attention mechanism. Attention mechanism which gives different weights to different channels can improve learning ability of the network [24]. Combing Depthwise Convolution with  $1 \times 1$  convolution will reduce the amounts of parameters and improves the ability to extract texture information compared with  $3 \times 3$  convolution layer.



(a) Overall CTUCC-CNN structure



(b) Details about Down-Sampling Unit and Light Conv Unit

**Figure 5:** Proposed CUTCC-CNN

### 3.2 Dataset Construction

The neural network performance is highly dependent on the selected samples and the size of the data set. In order to train CUTCC-CNN and ensure the accuracy of CU texture classification, we

choose 19 video sequences published by JVET with resolution ranging from  $1920 \times 1080$  to  $416 \times 240$ . With  $64 \times 64$  CUs of a frame in every 10 frames extracted, our dataset includes 228,750 samples. The details are listed in [Tab. 1](#). We randomly select 85% of them as the training set and 15% as the validation set.

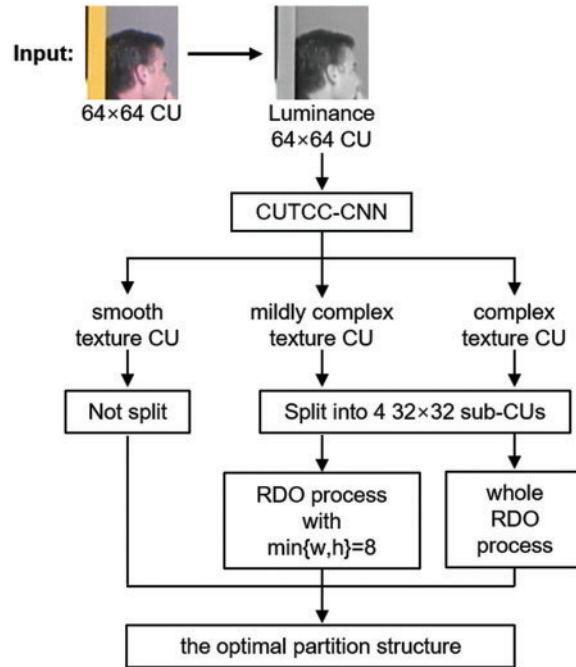
**Table 1:** Details about our dataset

Resolution ( $64 \times 64$ CU numbers/frame)	Sequence	Total frames	Samples
1920 $\times$ 1080 (480)	BasketballDrive	500	24000
	BQTerrace	600	28800
	Cactus	500	24000
	Kimono	240	11520
	ParkScene	240	11520
	MarketPlace	600	28800
	RitualDance	600	28800
1280 $\times$ 720 (220)	FourPeople	600	13200
	Johnny	600	13200
	KristenAndSara	600	13200
	BasketballDrillText	500	11000
832 $\times$ 480 (91)	BasketballDrill	500	4550
	RaceHorses	300	2730
	BQMall	600	5460
	PartyScene	500	4550
	416 $\times$ 240 (18)	BasketballPass	500
BlowingBubbles		500	900
BQSquare		600	1080
RaceHorses		300	540
Total number			228750

### 3.3 Main Process

The flow chart of our proposed fast CU partition for VVC using CTUCC-CNN is shown in [Fig. 6](#). In our method,  $64 \times 64$  CU is fed into CUTCC-CNN first. After the classification of CUTCC-CNN, this  $64 \times 64$  CU might be classified as smooth texture CU, mildly complex texture CU or complex texture CU. Then, appropriate RDO search process is selected according to the classification results output by CUTCC-CNN. If the current CU is a smooth texture CU, the partitioning process is terminated. Otherwise, CU will be divided by QT, resulting in four  $32 \times 32$  sub-CUs. When CU is judged as mildly complex CU, the encoder's RD Cost calculation will traverse the sub-CU with the minimum width/height limited to 8 pixels. When the current CU is classified as a complex texture CU, all possible RDO search processes will be performed. Finally, the encoder will decide the optimal partition structure for the  $64 \times 64$  CU.





**Figure 6:** Process of a fast CU partition method for H.266/VVC using CUTCC-CNN

#### 4 Experimental Results and Analysis

To verify the performance of our method, we embed our algorithm into VTM 10.2, and 16 video sequences from 4 different resolutions ( $1920 \times 1080$ ,  $1280 \times 720$ ,  $832 \times 480$  and  $416 \times 240$ ) are selected. The experiment is configured as AI encoding, and the Quantization Parameters (QP) are  $\{22, 27, 32, 37\}$ . The hardware configuration of the experimental platform is AMD Ryzen 7 4800H with Radeon Graphics, the main frequency is 2.90 GHz, and the memory is 16.0 GB. The software configuration is Microsoft Visual Studio 2019.

Based on the coding results of VTM 10.2, Bjøntegaard Delta Bit Rate (BD-BR) [25] is used to evaluate the objective compression performance. The percentage of coding time saving is the average of the four QPs relative to the VTM 10.2. The computational complexity of the relevant algorithms is measured by  $\Delta T$ , as shown in Eqs. (1) and (2)

$$\Delta T(QP_i) = \frac{T_a(QP_i) - T_{VTM}(QP_i)}{T_{VTM}(QP_i)} \quad QP_i = \{22, 27, 32, 37\} \quad (1)$$

$$\Delta T = \frac{\sum_i \Delta T(QP_i)}{T_{VTM}(QP_i)} \quad QP_i = \{22, 27, 32, 37\} \quad (2)$$

The algorithm presented in this paper is compared with the algorithm proposed in [13], and the experimental results are shown in Tab. 2. We can see that the proposed method reduces the coding time by 32.2% on average. For RitualDance, Johnny, KristenAndSara and other sequences with relatively smooth or regular textures, the proposed method can significantly reduce the computational complexity, and perform better than the algorithm in [13]. However, for sequences with complex or irregular textures such as PartyScene, BlowingBubbles and BQMall, the reduction of coding time is limited. The reduction of coding complexity is usually achieved by sacrificing coding quality. In

particular, in terms of objective compression performance, our method has an average increase of 0.55% in BD-BR compared with VTM 10.2, but is superior to the algorithm proposed in [13].

**Table 2:** Coding performance of the proposed fast CU partition method compared to [13]

Resolution	Sequence	Method in [13]		Proposed	
		BD-BR (%)	$\Delta T$ (%)	BD-BR (%)	$\Delta T$ (%)
1920 × 1080	BasketballDrive	0.52	-35.2	0.43	-39.8
	BQTerrace	1.18	-35.1	0.52	-33.6
	Cactus	0.65	-29.8	0.76	-27.2
	MarketPlace	0.61	-36.5	0.58	-25.7
	RitualDance	0.76	-39.2	0.42	-46.3
1280 × 720	FourPeople	1.11	-44.6	0.46	-30.6
	Johnny	1.00	-37.9	0.61	-54.3
	KristenAndSara	0.87	-40.0	0.53	-48.8
832 × 480	BasketballDrill	1.41	-40.8	0.59	-35.9
	RaceHorses	0.50	-32.6	0.48	-27.8
	BQMall	0.65	-40.5	0.89	-24.7
	PartyScene	0.37	-38.8	1.03	-16.3
416 × 240	BasketballPass	0.71	-37.7	0.56	-24.6
	BlowingBubbles	0.63	-40.2	0.31	-23.3
	BQSquare	0.52	-35.6	0.23	-29.2
	RaceHorses	0.54	-32.5	0.34	-27.4
Average		0.75	-37.3	0.55	-32.2

## 5 Conclusion

Considering high computational complexity caused by the newly introduced QTMT partition structure in H.266/VVC, this paper proposes a fast intra CU partition algorithm based on deep learning to analyze the CU texture complexity to simplify the original RDO search process. The main contribution of this paper is to build a CUTCC-CNN, which makes full use of texture information obtained in CU and classifies  $64 \times 64$  CUs into three types. Our proposed method can intelligently skip unnecessary RD Cost calculation and speed up the CU partition process. The experimental results show that our method can significantly improve the coding efficiency with only 0.55% objective compression performance.

**Acknowledgement:** First and foremost, I would like to thank my master's advisor Liu Pengyu. Since I was identified as Ms. Liu's student, she has given me a lot of care and guidance. In the academic and scientific research process, I communicated with Ms. Liu about the progress of my study and existing problems in time, and she gave a lot of constructive opinions and suggestions which helped me a lot. In the process of thesis writing, it is Ms. Liu who guides and embellishes my thesis diligently. At the

same time, I also want to thank the students in the multimedia Information Processing laboratory for their help and encouragement, and my parents and friends for their support and care.

**Funding Statement:** This paper is supported by the following funds: The National Key Research and Development Program of China (2018YFF01010100), Basic Research Program of Qinghai Province under Grants No. 2021-ZJ-704, The Beijing Natural Science Foundation (4212001) and Advanced information network Beijing laboratory (PXM2019\_014204\_5000 29).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] P. Jonsson, S. Carson, S. Davies, P. Lindberg, G. Blennerud *et al.*, *Ericsson mobility report*. Stockholm, Sweden, 2021. [Online]. Available: <https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/november-2021>.
- [2] Joint Video Exploration Team (JVET), *New versatile video coding standard to enable next-generation video compression*, 2021. [Online]. Available: <https://www.itu.int/en/mediacentre/Pages/pr13-2020-New-Versatile-Video-coding-standard-video-compression.aspx>.
- [3] A. Mercat, A. Mäkinen, J. Sainio, A. Lemmetti, M. Viitanen *et al.*, “Comparative rate-distortion-complexity analysis of VVC and HEVC video codecs,” *IEEE Access*, vol. 9, pp. 67813–67828, 2021.
- [4] M. A. Usman, M. R. Usman, R. A. Naqvi, B. Mcphilips, C. Romeika *et al.*, “Suitability of VVC and HEVC for video telehealth systems,” *Computers, Materials & Continua*, vol. 67, no. 1, pp. 529–547, 2021.
- [5] J. Tariq, A. Alfalou, A. Ijaz, H. Ali, I. Ashraf *et al.*, “Fast intra mode selection in HEVC using statistical model,” *Computers, Materials & Continua*, vol. 70, no. 2, pp. 3903–3918, 2022.
- [6] Y. Gao, P. Y. Liu, Y. Y. Wu and K. B. Jia, “Quadtree degeneration for HEVC,” *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2321–2330, 2016.
- [7] J. N. Chen, H. M. Sun, J. Katto, X. Y. Zeng and Y. B. Fan, “Fast QTMT partition decision algorithm in VVC intra coding based on variance and gradient,” in *Proc. VCIP*, Sydney, NSW, Australia, pp. 1–4, 2019.
- [8] H. C. Zhang, L. Yu, T. S. Li and H. K. Wang, “Fast GLCM-based intra block partition for VVC,” in *Proc. DCC*, Snowbird, UT, USA, pp. 382, 2021.
- [9] Q. W. Zhang, Y. B. Zhao, B. Jiang, L. X. Huang and T. Wei, “Fast CU partition decision method based on texture characteristics for H.266/VVC,” *IEEE Access*, vol. 8, pp. 203516–203524, 2020.
- [10] Q. W. Zhang, Y. B. Zhao, B. Jiang and Q. G. Wu, “Fast CU partition decision method based on Bayes and improved de-blocking filter for H.266/VVC,” *IEEE Access*, vol. 9, pp. 70382–70391, 2021.
- [11] C. Liu, K. B. Jia, P. Y. Liu and Z. H. Sun, “Fast depth intra coding based on layer-classification and CNN for 3D-HEVC,” in *Proc. DCC*, Snowbird, UT, USA, pp. 381, 2020.
- [12] M. Lei, F. L. Luo, X. Zhang, S. S. Wang and S. W. Ma, “Look-ahead prediction based coding unit size pruning for VVC intra coding,” in *Proc. ICIP*, Taipei, Taiwan, pp. 4120–4124, 2019.
- [13] G. W. Tang, M. G. Jing, X. Y. Zeng and Y. B. Fan, “Adaptive CU split decision with pooling-variable CNN for VVC intra encoding,” in *Proc. VCIP*, Sydney, NSW, Australia, pp. 1–4, 2019.
- [14] T. Y. Li, M. Xu, R. Z. Tang, Y. Chen and Q. L. Xing, “DeepQTMT: a deep learning approach for fast QTMT-based CU partition of intra-mode VVC,” *IEEE Transactions on Image Processing*, vol. 30, pp. 5377–5390, 2021.
- [15] Z. P. Jin, P. An, L. Q. Shen and C. Yang, “CNN oriented fast QTBT partition algorithm for JVET intra coding,” in *Proc. VCIP*, St. Petersburg, FL, USA, pp. 1–4, 2017.
- [16] X. Sun, P. Y. Liu, X. W. Jia, K. B. Jia, S. J. Chen *et al.*, “An effective CU depth decision method for HEVC using machine learning,” *Computer Systems Science and Engineering*, vol. 39, no. 2, pp. 275–286, 2021.

- [17] X. R. Zhang, X. Sun, X. M. Sun, W. Sun and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.
- [18] T. Lin and S. Maji, "Visualizing and understanding deep texture representations," in *Proc. CVPR*, Las Vegas, NV, USA, pp. 2791–2799, 2016.
- [19] L. Sharan, C. Liu, R. Rosenholtz and E. H. Adelson, "Recognizing materials using perceptually inspired features," *International Journal of Computer Vision*, vol. 103, no. 3, pp. 348–371, 2013.
- [20] X. R. Zhang, W. F. Zhang, W. Sun, X. M. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.
- [21] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, Lake Tahoe, NV, USA, pp. 1097–1105, 2012.
- [22] L. A. Gatys, A. S. Ecker and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. NIPS*, Montreal, Canada, pp. 262–270, 2015.
- [23] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. CVPR*, Honolulu, HI, USA, pp. 1800–1807, 2017.
- [24] J. Hu, L. Shen, S. Albanie, G. Sun and E. H. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.
- [25] G. Bjöntegaard, "Calculation of average PSNR differences between RD-curves," in *Proc. VCEGM*, Austin, TX, USA, 2001.