The International Journal of Engineering and Science (IJES)
|| Volume || 14 || Issue || 2 || Pages || PP 22-31 || 2025 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

A Fast Partition Method for H.266/VVC Intra Prediction Based on Spatial Correlation and Deep Learning

Yun-Jung Li, Chi-Feng Hsiao, Chou-Chen Wang
Department of Electronic Engineering, I-Shou University, Kaohsiung, Taiwan
Corresponding Author: Chou-Chen Wang

ABSTRACT
H.266/VVC is the latest video coding standard which can support resolutions up to 8K and 16K as well as 360° videos. To obtain higher coding quality, VVC adopts the novel technology of quadtree with nested multi-type tree (QTMT) division scheme including the quadtree (QT), vertical and horizontal binary tree (BT) and ternary tree (TT) split for the coding tree unit (CTU) from 128×128 to 4×4 coding unit (CU). However, the computational complexity is considerably high since VVC selects the best division mode using recursive rate–distortion optimization (RDO). To speed up the VVC encoding process, Zhao et al. [7] proposed a size-adaptive convolutional neural network (SAE-CNN) using deep learning to improve the complexity for VVC intra prediction, recently. They first analyze the content complexities of CU through the entropy value. When the complexity falls in the uncertain interval defined between smooth and complexity, the SAE-CNN is used to make the split decision. However, they ignored the fact that spatial correlation existing frame being stronger when the higher resolutions (such as 4K or 8K). This will lead to reduce the encoding efficiency of their method. Therefore, in order to further speed up the VVC intra prediction time, we propose a spatial correlation searching algorithm and a simplified SAE-CNN structure to improve the encoding efficiency of Zhao's method. The experimental results show that the proposed method can achieve an average time improving ratio (TIR) about 48.04% with an average 1.13% of the increase in Bjøntegaard delta bit rate (BDBR) as compared to VVC (VTM7.0). In addition, the normalized encoding efficiency (η) of the proposed method can reach 43.5. On the other hand, the proposed method can further raise the average $TIR = 9.38\%$ and $\eta = 0.22$ with an average 0.21% of the increase in BDBR when compared with Zhao's method. Therefore, the proposed fast VVC intra frame prediction can indeed further improve the coding efficiency, especially when the image resolution is higher.
KEYWORDS– *Versatile video coding, deep learning, convolutional neural network.*

Date of Submission: 27-02-2025 Date of acceptance: 16-03-2025

I. INTRODUCTION

H.266/VVC (versatile video coding) is the latest video coding standard [1], which can further reduce video compressed sizes by 23% compared to H.265/HEVC while maintaining the same visual quality [2-3]. However, this improvement comes at the cost of significantly higher complexity of encoding system and computational requirements. To obtain higher coding quality, H.266/VVC (hereafter interchange H.266/VVC and VVC) adopts the novel technology of quadtree with nested multi-type tree (QTMT) division scheme including the quadtree (QT), vertical and horizontal binary tree (BT) and ternary tree (TT) split for the coding tree unit (CTU) from 128×128 to 4×4 coding unit (CU). Unlike H.265/HEVC, which uses only a quad-tree (QT) structure for partitioning, H.266/VVC introduces additional BT and TT partitioning structures. These allow CTUs to be divided into CUs of varying sizes and depths. Subsequently, each CU undergoes intra-prediction, inter-prediction, transformation, and quantization to determine the optimal prediction mode for encoding. The design of the rectangle type coding unit can better fit the shape of objects and make the selection of coding units more flexible.

However, the introduction of new partitioning structures in VVC has caused a dramatic increase in the number of CUs. This results in a significant increase in computational complexity, and leading to a substantial decrease in encoding speed. Therefore, it will face significant challenges in achieving real-time video transmission and applications. In recent years, a deep convolutional neural network (CNN) has achieved a great success in multimedia information processing fields, pattern recognition technology, and image compression [4-6]. To improve the encoding speed of VVC, Zhao et al. [7] (hereafter referred to as Zhao) proposed a method using size-adaptive convolutional neural network (SAE-CNN) to determine a coding unit (CU) whether should be further partitioned in intra prediction module. This approach reduces the number of CUs to be encoded, thereby reducing computational complexity and accelerating the encoding process of intra frame.

The remainder of this paper is organized as follows. In Section II we briefly review the H.266/VVC and

SAE-CNN for intra prediction. Section III elaborates the proposed fast partition algorithm to speed up intra prediction in H.266/VVC. The experimental results are presented in Section IV. Finally, Section V summarizes our conclusions.

II. BRIEF REVIEWS OF H.266/VVC AND SAE-CNN INTRA PREDICTION

2.1 H.266/VVC Intra prediction

The H.266/VVC encoding system is primarily divided into four units: CTU, CU, Prediction Unit (PU), and Transform Unit (TU). Figure 1 illustrates the CTU partitioning structure in an .266/VVC frame. Each frame is initially divided into 128×128 CTUs. Each CTU is then further partitioned into CUs of varying sizes. To better capture the lines and features within the image, in addition to retaining the quad-tree (QT) partitioning structure from H.265/HEVC, H.266/VVC introduces QT plus multi-type tree (QT+MTT) including QT, BT and TT splitting methods.

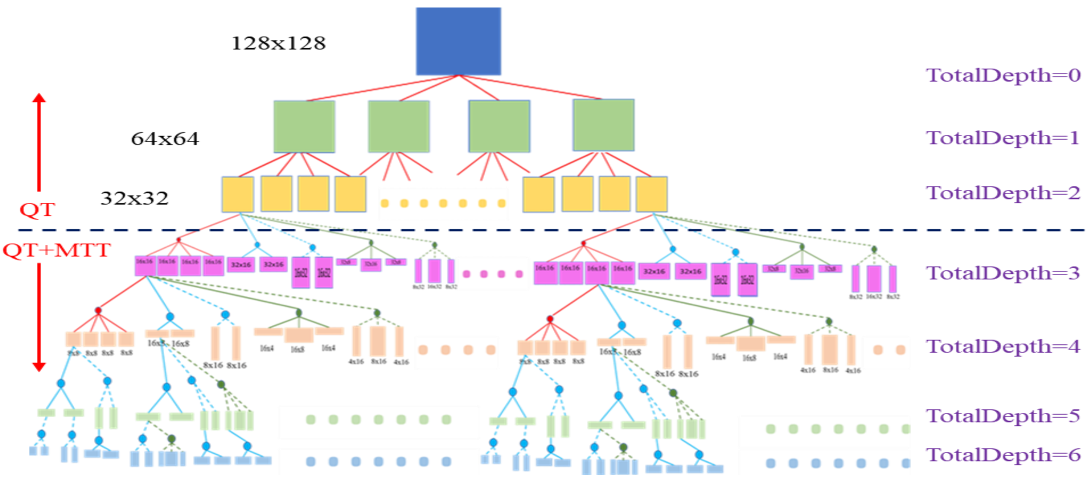


Fig.1 The CTU Partitioning Structure with the QT+MTT in H.266/VVC.

The BT splitting includes horizontal binary-tree (BT-H) and vertical binary-tree (BT-V), while the TT partitioning includes horizontal ternary-tree (TT-H) and vertical ternary-tree (TT-V), as illustrated in Figure 2. These newly introduced BT and TT partitioning methods are collectively referred to as MTT partitioning. This flexible MTT partitioning system allows H.266/VVC to adapt more effectively to diverse image content, capturing structural details and improving compression efficiency. However, the added splitting complexity also increases the computational demands of the encoding process, which remains a significant challenge for practical applications in real-time transmission.

(a) BT-H and BT-V (b) TT-H and TT-V

Fig.2 Splitting modes used in MTT structures.

Take an example as shown in Fig. 1, which the TotalDepth is defined as the sum of QTDepth and MTTDepth. A 128×128 CTU is first partitioned into four 64×64 CUs, and each 64×64 CU further partitioned into four32×32 CUs by using a QT structure. In other words, The TotalDepth equals to 2 due to QTDepth = 2 and MTTDepth = 0. Secondly, for each 32×32 CU can be split by using MTT to ensure that the CUs in VVC are more flexible and more suitable for the intra and inter prediction. MTT structures commonly involve four splitting modes: BT-H, BT-V, TT-H, and TT-V, as illustrated in Fig. 2. Finally, until it is partitioned into 4×4 CU. The CU shape exists square or rectangular in QTMT structure. If the leaf node size of QT is 128×128, it will not be further divided by BT and TT, because this size exceeds the root node size of maximum BT and TT (64×64). If the leaf node size of QT is not 128×128, the QT leaf node can be further divided by MTT. At this time, the QT leaf node is also the MTT root node which the depth is 0.

After a CTU is partitioned into several CUs, those CUs are used in intra prediction to select the optimal intra mode from 67 intra prediction modes. In other words, it traverses all possible division modes of the CU, and calculates the rate-distortion (RD) cost and selects the division mode with the minimal rate distortion cost (RD-cost). The calculation formula of the RD cost is defined as follows:

$$RD_{cost} = \mathrm{D} + \lambda \times R_{mode} \quad (1)$$

where RD_{cost} is the RD cost, λ is the Lagrange multiplier, R_{mode} represents the number of encoding bits and D is the reconstruction distortion.

During intra-prediction in H.266, each CU executes the 35 original prediction modes from the same as H.265, along with the newly added extended angular modes. As a result, each CU must process a total of 41 prediction modes, and the best mode is selected based on the RD cost. For example, in a 4K×2K resolution image, there are 507 CTUs of size 128×128, which are further subdivided into smaller CUs. The intra-prediction machine requires approximately 490 million RD cost calculations [8-9]. This leads to extremely high computational complexity. Although this process can ensure superior image quality, it fails to meet the requirements for real-time applications. Therefore, to find a way to reduce computational complexity without compromising image quality or significantly increasing the bit rate is a crucial research challenge. In this paper, we will propose an acceleration algorithm to solve the problem using spatial correlation searching algorithm and CNN method.

2.2 Size-adaptive convolutional neural network

The basic architecture of CNN is composed of one or more convolutional layers and pooling layers. For image applications, the CNN method can be regarded as an image-input and image-output structure. To reduce the computational complexity of H.266/VVC intra prediction, Zhao proposed an efficient decision method using a size-adaptive CNN (SAE-CNN) [7]. The SAE-CNN evaluates the complexity of residual CUs before performing intra prediction process, which makes pre-decision judgments using a pre-decision dictionary. The dictionary proposed by Zhao is first established based on statistical theory and analysis. Therefore, for a CU, the pre-decision judgments are made based on whether it is square or rectangular. Three situations including complexity (split), monotone (not split), and uncertain are judged according to statistical entropy. Only the uncertain CU residual block is sent to SAE-CNN, and it is judged whether a division operation on the CU should be performed according to the output result. If the output result is a split, then the subsequent division mode selection is performed to select a suitable division mode, and the division process is ended. If the result is a not-split, the division process is directly ended.

Since each CU exhibits varying levels of image complexity, Zhao first used entropy to calculate the complexity of each residual CU and classify them into three situations. The mathematical definition of entropy $H(x)$ is as follows:

$$H(x) = -\sum_{i}^{n} P(x_i) \log_2 P(x_i) \quad (2)$$

where $P(x_i)$ represents the probability of occurrence of x_i.

To investigate the relationship between entropy values and complexity, Zhao conducted many experiments and statistical analysis using different QP values. First, the entropy values of each residual CU were calculated after intra encoding with various QP values. Then, based on the corresponding entropy values, the division results of each residual CU were recorded and labeled as either split or not split. Table I shows the interval of entropy values to judge a residual CU as complexity or monotone or uncertain as determined by Zhao's pre-decision dictionary for QP values of 22, 27, 32, and 37. For an example of $\mathrm{QP} = 32$, a residual CU is judged as monotone block when $H(x) < 4.2$, and as complexity when $H(x) > 7.3$, and as uncertain when $4.2 \leq H(x) \leq 7.3$. The decision method for CU complexity classification proposed by Zhao is illustrated in Figure 3,

TABLE I. The interval of entropy values to judge a residual CU.

QP	22	27	32	37
$H(x)$	[3.1, 6.8]	[3.9, 7.1]	[4.2, 7.3]	[4.8, 7.4]

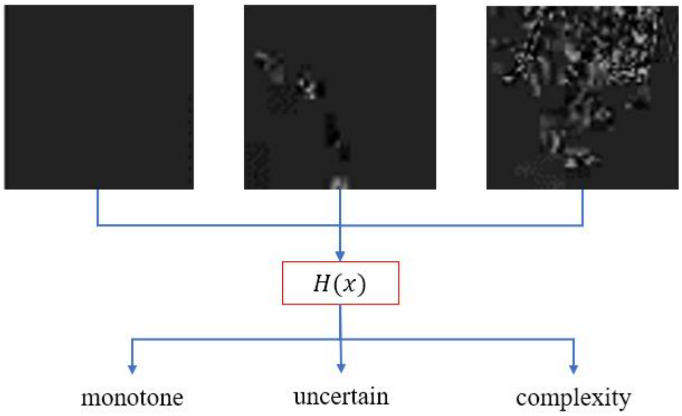


Fig.3 The decision method for residual CU classification.

The SAE-CNN network architecture proposed by Zhao is shown in Figure 4. It consists of 4 convolutional layers, 3 pooling layers, and 1 fully connected layer, followed by an output layer. The input residual CU sizes range from 32×32 to 8×8. First, the residual CU is fed into the network, where its features are extracted through convolutional layers. The size of the residual CU is reduced after passing through two pooling layers. In the pooling layers, the first two layers utilize size-adaptive pooling adjustments to account for the irregular shapes of residual CUs generated after MTT division. These shapes include not only squares but also rectangles. Constructing a dedicated network architecture for each residual CU size would be impractical. To allow a single architecture to handle residual CUs of different sizes effectively, the pooling layers are designed to normalize the dimensions of residual CUs before they proceed to the fourth convolutional layer for feature extraction. Since the main role of the pooling layers is to reduce the size of the feature maps, their influence on the overall recognition accuracy of the SAE-CNN is relatively minor. By adjusting the parameters of the pooling layers, the network ensures that all residual CU sizes are unified to the same dimensions before feature extraction in the fourth convolutional layer.

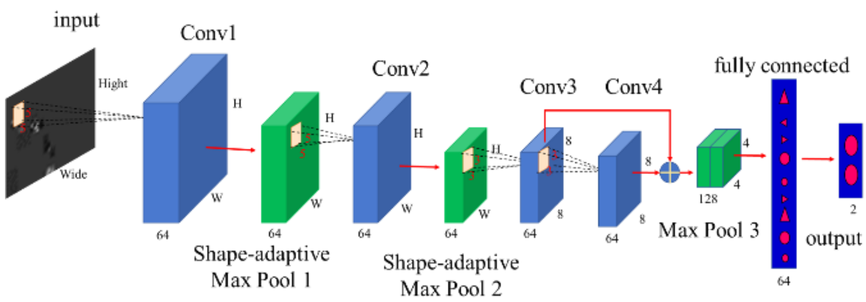


Fig.4 Diagram of the SAE-CNN structure.

From SAE-CNN method, it can be observed that each residual CU is first to perform a classification of complexity. If the complexity of residual CU is classified as uncertain type, the SAE-CNN is employed to make a division decision. On the other hand, when the complexity is classified as monotonic type, no division is performed, and when the complexity is classified as complex type, division is performed directly. However, SAE-CNN disregards the spatial correlation of CTU tree structures between their neighboring blocks in monotonic regions. In intra-frame coding, neighboring blocks are often used as reference points for predict coding. There is a very high spatial correlation between the CTU tree structures of neighboring blocks in the monotonic regions. Therefore, the best CU partition of the CTU may be the same as or similar to the split structures of the spatial four neighbor CTU tree structures. Therefore, if we can find the best CTU tree structure from the neighboring blocks, this will save the time required for each CU to perform prediction and determine the optimal CTU. This can effectively improve the encoding efficiency and reduce computation time.

III. PROPOSED METHOD

Although SAE-CNN can significantly reduce the encoding complexity of H.266/VVC, it does not consider the spatial correlation of neighboring block CTU trees within the frame. Therefore, this paper proposes a spatial correlation searching algorithm (SCSA). For monotonic regions, the best CTU tree structure is predicted by using the proposed SCSA method. For uncertain regions, a simplified SAE-CNN is applied to fast make a division decision. This proposed method aims to further optimize the encoding process by incorporating spatial correlation information, and improving both the efficiency and accuracy of the encoding decision by simplified SAE-CNN.

3.1 Spatial correlation of CTU tree structure

Figure 5 shows an image of the best CTU tree structure after intra encoding in the VVC reference software VTM7.0 [10]. We can deeply observe that the relations of CTU tree from many monotonic regions in Fig. 5. To further observe, we take the CU sizes of 32×32 in Figures 5(a)~5(h) as an example. The split structures of CTUs in the current frame, ex. Figs. 5(a)~5(d), are the same as the split structures of the spatial four neighboring CUs, or are similar to the split structures of the spatial four neighbor CUs, ex. Figs. 5(e)~5(f). Thus, when encoding the current frame, the current CTU can be predicted through the split structure of the spatial four neighboring CUs in the current frame.

From the analysis of the encoding results, it can be observed that the CTU tree structures of monotonic regular blocks are mostly divided to Depth 2 or Depth 3. In contrast, the CTU tree structures of monotonic regular blocks also have higher spatial correlation between neighboring CTU tree structures. Therefore, by referencing the CTU tree structure of neighboring blocks divided into Depth=2 or Depth=3, and using it as the CTU tree structure for the block to be encoded, it is possible to bypass the original division process of H.266/VVC for finding the optimal CTU. Considering that directly deciding the partition of 64×64 CU may lead to a huge loss in encoding performance, we do not take the partition decision of 64×64 into our account in this work. We thus decide to determine the partition of 32×32 CU by using fast algorithm. Therefore, the encoding quality and efficiency can achieve a nice balance.

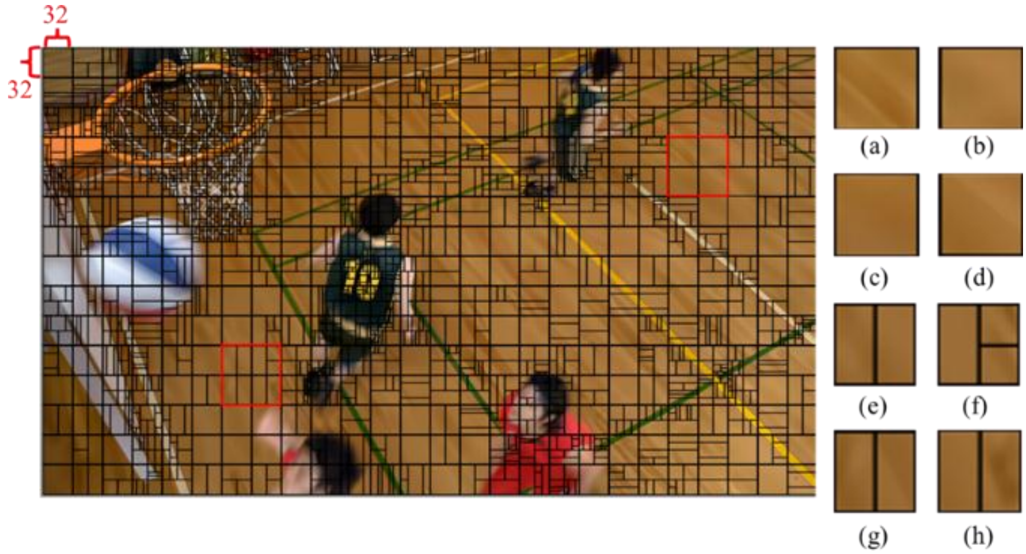


Fig. 5 Best CTU tree structure of intra coding for H.266/VVC.

Next, we calculate and analyze the occurring probability of the same CTU tree structures between the current encoding CU (X) and neighboring blocks at Depth=2 or Depth=3. In general, the encoding order of CU in H.266/VVC is from left to right and from top to bottom. Evidently, the neighboring blocks of current CU in the same depth are shown in Figure 6. Therefore, to employ the spatial correlation of CTU tree, we adopt four spatial four neighboring CTUs in the current frame. Figure 6 shows the four neighboring blocks of encoded CTU for current CU, including left (*Ref A*), left upper (*Ref B*), upper (*Ref D*) and right upper (*Ref E*), respectively.

We have observed the encoding results in a real-life video sequence, CUs in a slow-motion and low complexity frames are usually coded using larger partitions such as QT types i.e., 64×64, whereas CUs in a fast-motion or high-complexity frames are likely to be coded using smaller partitions such as MTT types, i.e., 16×8 or 16×8. From the statistical analysis in our experiments, we can find that there are different depth level distributions for sequences. Table II shows the average probability of the same CTU trees with four different quantization parameters (QP: 22, 27, 32, 37) values. From the table, it is observed that Ref A, Ref B, Ref C and Ref D have a matching tree structure about 41%, 32%, 39% and 22% on average, respectively. This demonstrates a high spatial correlation between the CTU tree structures of the current block and the neighboring blocks. Therefore, the proposed spatial correlation searching algorithm (SCSA) follows the searching priority order as *Ref A→ Ref D→ Ref B→ Ref E*. On the other hand, Figure 6 also shows the search priority order according to the correlation values determined by experiments.

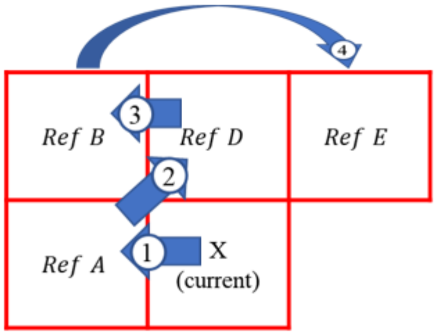


Fig.6 Four spatial neighboring reconstructed CTUs and searching priority order.

TABLE II. The the average probability of the same CTU trees with four different QPs.

Resolution	Sequence	*Ref A*	*Ref B*	*Ref D*	*Ref E*
1920×1080	Kimono1	50%	45%	49%	30%
	ParkScene	41%	36%	44%	25%
	Cactus	35%	26%	33%	18%
	BasketballDrive	38%	24%	27%	16%
	BQTerrace	45%	33%	38%	25%
1280×720	Vidyo1	35%	26%	34%	19%
	Vidyo3	43%	33%	44%	24%
	Vidyo4	41%	31%	41%	22%
	Average	41%	32%	39%	22%

Before proceeding with the reference CTU tree, it is first determined whether the current CU size is 32×32. If so, the neighboring reference block is selected, and the CTU tree structure of the reference block is checked to ensure it has been divided only to Depth=2 or Depth=3. Next, a threshold value is set to check whether the RD cost of the CTU tree structure of the reference block is lower than the threshold. If the RD cost is below the threshold, the CTU tree structure of the reference block is adopted as the CTU tree structure for the current CU, and the current CU's division process ends, moving on to the next CU encoding. If none of the neighboring blocks meet the conditions, the algorithm proceeds to the next step.

3.2 Threshold setting

The threshold value is a crucial parameter in the proposed SCSA. How to determine the threshold is the key issue to the proposed method in H.266/VVC. It is clear that the larger the thresholds are, the more neighboring CTU trees can be selected and the encoding complexity can be further reduced. However, more CTU trees are incorrectly selected at the same time, which results in more significant loss in image quality and more bits used in encoding prediction error CTU trees. We have conducted several experiments with different candidate sets of thresholds based on different degradation of video quality for some video sequences, and then select an appropriate threshold value which provides a good tradeoff between R-D performance and computational complexity in practice.

Considering that VVC is aimed at high-quality video, we selected five high-resolution (1920×1080) video sequences (Kimono1, ParkScene, Cactus, BasketballDrive, BQTerrace) and encoded them at four different QP values (22, 27, 32, 37). We conducted experiments with different threshold values and observed the results. To evaluate the performance of the VTM encoder, we used several measurement tools, including time increasing ratio (TIR), ΔBitrate, ΔY-PSNR, and the normalized encoding efficiency (η) defined as follows:

$$TIR = \frac{VTM_{Time} - Method_{Time}}{VTM_{Time}} \times 100\% \tag{3}$$

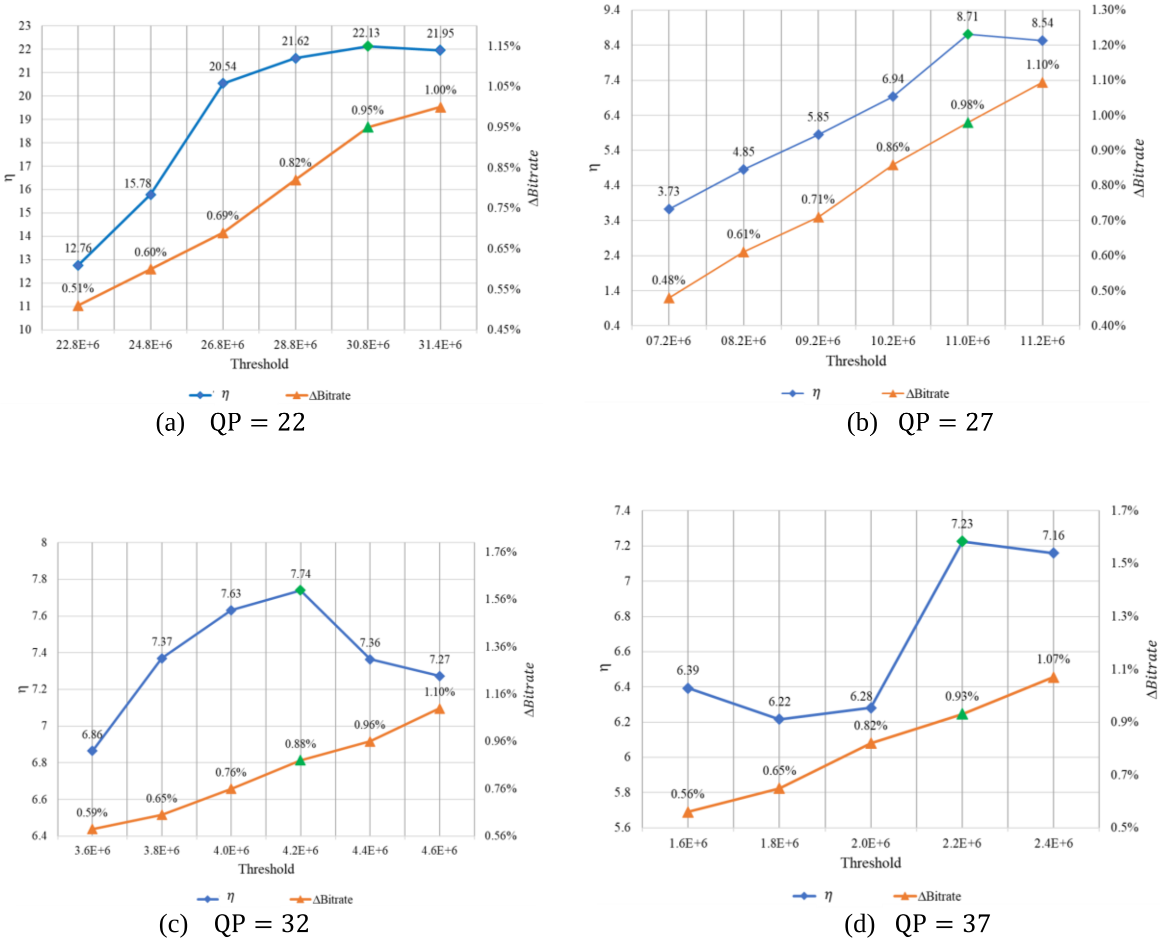
$$\Delta Bitrate = \frac{Method_{Bitrate} - VTM_{Bitrate}}{VTM_{Bitrate}} \times 100\% \tag{4}$$

$$\Delta Y\text{-}PSNR = Method_{Y\text{-}PSNR} - VTM_{Y\text{-}PSNR} \tag{5}$$

$$\eta = \frac{TIR}{\Delta Bitrate} \tag{6}$$

The experimental results for different QP values are shown in Figures 7(a)~7(d). From Fig.7, we can observe when the threshold value increases, the encoding efficiency (η) become larger. In other words, the encoding time savings become larger, but the bitrate also increases. When the threshold value is smaller, the encoding time savings are limited, and the bitrate increase is relatively smaller. Additionally, during VVC encoding, an increase in bitrate is used to maintain the decoded video quality. However, when the bitrate increases too much, it imposes a burden on network bandwidth and storage. Therefore, when setting the threshold value, the impact of the bitrate increase on network bandwidth needs to be considered.

Thus, the threshold value is set with the primary consideration being the change in bitrate. The goal is to keep the bitrate increase within a reasonable range while still saving encoding time. Based on this, we selected a threshold value where the overall average ΔBitrate for each QP increases by approximately 0.5% to 1%, and chose the threshold value that provides the best encoding performance for the current QP. The best threshold values for different QP are summarized in Table III based on the experimental results.



(a) QP = 22 (b) QP = 27

(c) QP = 32 (d) QP = 37

Fig.7 The experimental results for different QP values.

TABLE III. The interval of entropy values to judge a residual CU.

QP	22	27	32	37
Threshold	3.1×10^7	1.1×10^7	4.2×10^6	2.2×10^6

3.3 Simplified SAE-CNN

For the uncertain regions, we use a simplified SAE-CNN decision algorithm to partition CU. If no reference CTU tree structure is found in the neighboring blocks, we will retain the original H.266/VVC partitioning approach. Therefore, for blocks where no reference CTU tree structure is found, we refer to Zhao's SAE-CNN network architecture for partitioning decisions. However, Zhao's SAE-CNN architecture uses 4 convolution layers and 3 pooling layers. Since we have already pre-processed the blocks using spatial correlation from the neighboring CTU tree structures, the number of features in the CU is significantly reduced. To further save encoding time, we have designed a simplified SAE-CNN network architecture by reducing the architecture to 3 convolution layers and 3 pooling layers, making it more efficient while maintaining adequate decision-making capabilities.

3.4 Fast intra coding method for H.266/VVC

The paper proposes a fast H.266/VVC intra coding algorithm using the spatial correlation between CTU tree. For blocks in the monotonic region, the SCSA is used. If the residual CU is judged to be uncertain, it is handed over to the proposed simplified SAE-CNN for further partitioning decisions. For complex regions, partitioning is performed directly. This approach avoids wasting time by not sending every residual CU to the simplified SAE-CNN for decision-making, ensuring efficiency in the encoding process. The flowchart of the proposed fast intra coding method for H.266/VVC is shown in Fig. 8.

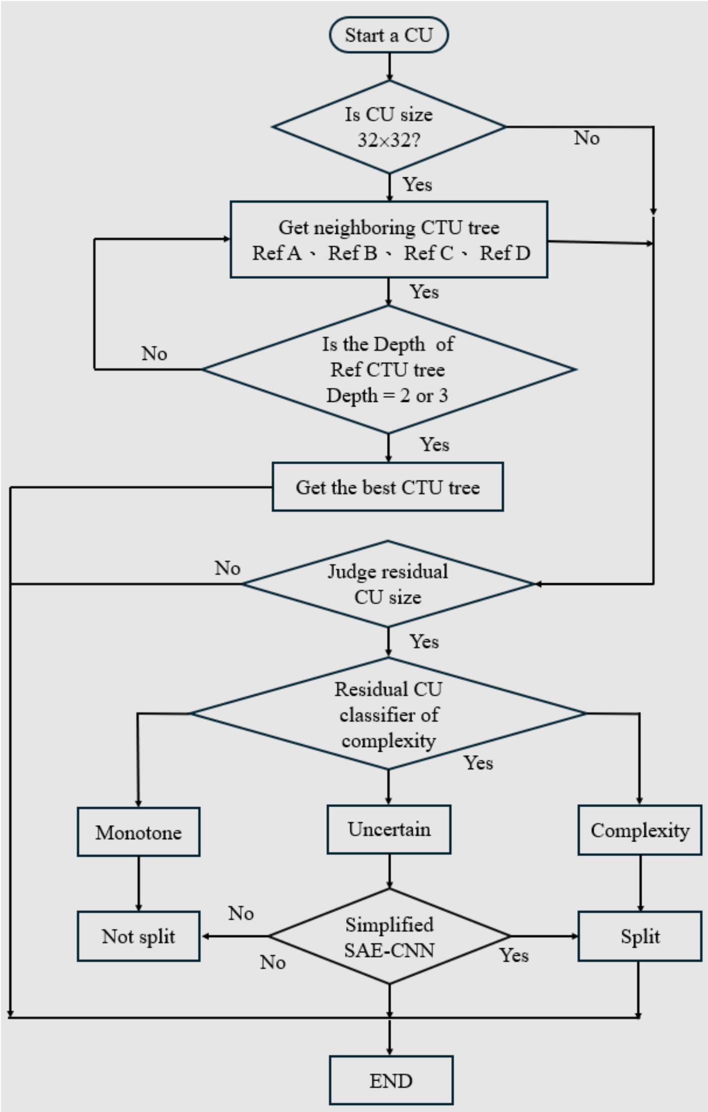


Fig.8 Fast intra coding method for H.266/VVC.

IV. EXPERIMENTAL RESULTS

In this paper, we have implemented an intra coding for H.266/VC using the proposed SCSA based on GPU in VTM 7.0 [6] encoder test model, the encoding configuration is summarized as follows:

(1) Scenario: all intra frames (IIII….).
(2) QP = 22、27、32、37
(3) To be encoded frames: 10 frames
(4) Standard test sequences:
Class B (1920×1080): Kimono1, ParkScene, BasketballDrive, BQTerrace
Class C (832×480): BasketballDrill, BQMall, PartyScene, RaceHorses
Class D (416×240): BQSquare, BlowingBubbles, RaceHorses
Class E (1280×720): Fourpeople, Johnny, KristenAndSara, vidyo1, vidyo4

Simulations are conducted on a desktop with
(1) OS: Windows 10 64-bit
(2) CPU: Intel(R)Xeon(R)E3-1230v3
(3) GPU: NVIDIA GeForce GTX 1060-6GB
(4) Memory: 32 GB
(5) Python version: Python3.6.13

Training video dataset selected in the SAE-CNN:
(1) Cactus (1920×1080)：25frames
(2) BasketballDrillText (832×480：25frames
(3) BasketballPass (416×240) ：25 frames
(4) Vidyo3(1280×720) ：25 frames

In our experiments, to ensure the fair results of the experiment and the effectiveness of the proposed simplified SAE-CNN model, the video sequences selected in the test video set is full different from the video sequence selected in the training dataset. For a fair comparison, we also selected the same training dataset in H.266/VVC intra frame coding and the same conditions of Zhao's method [7]. The coding performance is evaluated by the comparisons of BDBR (Bjontegaard delta bit rate) [11] and TIR. On the other hand, to evaluate the time improvement of the proposed method, we define the normalized encoding efficiency as follows:

$$\eta = \frac{TIR}{\mathrm{BDBR}} \tag{7}$$

The experimental results shown in Table IV, we find that the proposed method can achieve an average TIR about 48.04% with an average 1.13% of the increase in BDBR as compared to VTM7.0. In addition, the normalized encoding efficiency (η) of the proposed method can reach 43.5. On the other hand, the proposed method can further raise the average $\mathrm{TIR} = 9.38\%$ and $\eta = 0.22$ with an average 0.21% of the increase in BDBR when compared with Zhao's method. Therefore, the proposed fast H.266/VVC intra frame prediction can indeed further improve the coding efficiency, especially when the image resolution is higher.

Table IV. Performance comparisons between the proposed method and Zhao's method

Resolution	Sequence	BDBR(%)		TIR(%)		η	
		Zhao	Proposed	Zhao	Proposed	Zhao	Proposed
1920×1080	Kimono1	0.89	0.91	40.89	53.71	45.95	59.02
	ParkScene	0.78	0.81	40.08	45.97	51.39	56.75
	BasketballDrive	0.81	0.83	38.72	42.35	47.81	51.02
	BQTerrace	1.01	1.10	41.46	45.79	41.05	41.63
1280×720	FourPeople	1.15	1.39	44.89	59.52	39.03	42.82
	Johnny	0.91	1.46	45.25	60.23	49.72	41.25
	KristenAndSara	1.28	1.60	43.06	62.06	33.64	38.79
	vidyo1	1.44	1.63	46.87	65.75	32.55	40.33
	vidyo4	0.85	1.21	43.90	59.22	51.64	48.94
832×480	BasketballDrill	1.11	1.18	36.74	40.22	33.09	34.09
	BQMall	0.86	1.20	35.78	42.45	41.60	35.37
	PartyScene	0.53	0.67	31.47	37.59	59.38	56.10
	RaceHorses	0.78	1.12	33.76	43.53	43.29	38.87
416×240	BQSquare	0.71	0.85	31.51	34.48	44.37	40.57
	BlowingBubbles	0.92	1.11	32.92	39.06	35.78	35.19
	RaceHorses	0.74	1.04	31.27	36.72	42.26	35.31
Average		0.92	1.13	38.66	48.04	43.28	43.50

V. CONCLUSION

In this paper, we presented a fast partition method of intra prediction for H.266/VVC based on spatial correlation and CNN. Our results indicate that the proposed method outperforms Zhao's SEA-CNN method in time increasing ratio and RD performance.

REFERENCES

[1]. B. Bross et al., "Overview of the Versatile Video Coding (VVC) Standard and its Applications," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736-3764, Oct. 2021.

[2]. G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," in *IEEE Trans. Circuits Syste*m Video Technology, vol. 22, no. 12, pp. 1649- 1668, Dec. 2012.

[3]. Fraunhofer Institut zeigt 50% besseren HEVC Nachfolger VVC auf der.https://web.archive.org/ web/20181108065644/ https://www.slashcam.de/news/single/Fraunhofer-Institut-zeigt-50--besseren-HEVC-Nachfo 14589.html

[4]. Z. T. Zhang, C. H. Yeh, L. W. Kang, and M. H. Lin, "Efficient CTU-based intra frame coding for HEVC based on deep learning," *Proc. Asia Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, pp. 661-664, Dec. 2017.

[5]. C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 184-199, 2014.

[6]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 770-778, Jun. 2016.

[7]. J. Zhao, P. Dai, and Q. Zhang, "A Complexity Reduction Method for VVC Intra Prediction Based on Statistical Analysis and SAE-CNN," in Electronics, vol. 10, no. 24, pp. 3112, Dec. 2021.

[8]. C. Wang, Y. Chang, and K. Huang, "Efficient Coding Tree Unit (CTU) Decision Method for Scalable High-Efficiency Video Coding (SHVC) Encoder," in Recent Advances in Image and Video Coding. London, United Kingdom: IntechOpen, 2016.

[9]. G. Tang, M. Jing, X. Zeng and Y. Fan, "Adaptive CU Split Decision with Pooling-variable CNN for VVC Intra Encoding," in 2019 IEEE Visual Communications and Image Processing (VCIP), pp. 1-4, 2019

[10]. H.266 Test Model (VTM 7.0) documentation. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/

[11]. G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in Proc. 13th VCEG Meeting, pp. 1–5, Austin, TX, USA, Jan. 2001.