FAST QTBT PARTITIONING DECISION FOR INTERFRAME CODING WITH CONVOLUTION NEURAL NETWORK

Zhao Wang[*], *Shiqi Wang*[+], *Xinfeng Zhang*[#], *Shanshe Wang*[*], *Siwei Ma*[*]

*Institute of Digital Media, Peking University, Beijing 100871, China
+Department of Computer Science, City University of Hong Kong, Hong Kong, China
#University of Southern California, Los Angeles, California, USA
{zhaowang, sswang, swma}@pku.edu.cn; {sqwang1986,zhangxinf07}@gmail.com

ABSTRACT

In the latest Joint Video Exploration Team (JVET) development, the quadtree plus binary tree (QTBT) block partition structure is proposed for more flexible block partitioning. Compared to the quadtree partitioning in HEVC, QTBT can achieve better compression performance at the expense of significantly increased encoding complexity. To address this issue, we propose a convolution neural network (CNN) oriented fast QTBT partitioning decision algorithm for inter coding. We analyze the QTBT in a statistical way, which effectively guides us to design the architecture of the CNN. Furthermore, the false prediction risk is controlled based on temporal correlation to improve the robustness of the scheme. Experimental results show that the proposed algorithm can speed up QTBT block partition structure by reducing 35% encoding time on average with only 0.55% increase in bit rate, which enables its applications in practical scenarios.

Index Terms— Quadtree plus binary tree, block partitioning, CNN, video coding

1. INTRODUCTION

The block-based coding structure has been recognized as the core of the state-of-the-art video coding standards because of its capability in achieving high compression performance. To further investigate the flexibility of the block partitioning, a quadtree plus binary tree (QTBT) block partitioning structure was proposed during the recent Joint Video Exploration Team (JVET) development [1]. Fig. 1 illustrates an example of the QTBT block partition structure. For a CTU, it is first partitioned by the quadtree (QT), and then partitioned by the binary-tree (BT). For a block which has not been partitioned by the BT, it can be further partitioned by the QT, horizontal BT, or vertical BT, and the optimal mode minimizing the rate-distortion (RD) cost will be selected and signaled. After the BT partitioning, the final BT leaf nodes are termed as coding units (CUs) which are used for prediction and transform without any further splitting. During the encoding process, parameters *MinQSize*, *MaxBTSize* and *MaxBTDepth* restrict the minimal allowed QT leaf node size, maximal allowed BT root node size and maximal allowed BT depth. Though QTBT structure achieves higher compression performance compared to that in HEVC, the encoding complexity has also been dramatically increased due to the fact that more recursive splitting iterations are performed. In view of this, a fast QTBT partitioning algorithm is highly desired.

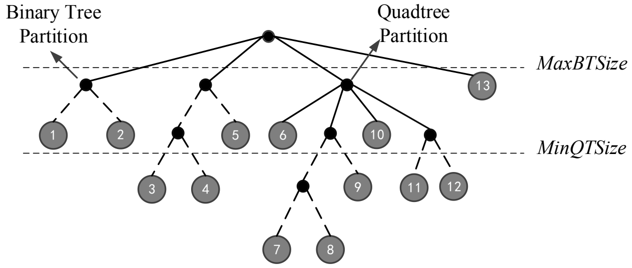

Fig. 1. Illustration of the QTBT block partitioning structure.

In the literature, the fast decision algorithms in video coding are mainly designed based on the statistical information. In [2], Gweon *et al.* proposed a Coded Block Flag (CBF) based early termination method. In particular, if there exists zero CBFs for all luma and chroma components, the remaining partitions of the current CU can be totally skipped. In [3], Shen *et al.* exploited the mode correlations among different depth levels. In [4], Wang *et al.* proposed a model to estimate the RD cost in a low-complexity way. In [5], Vanne *et al.* proposed an optimized scheme that conditionally evaluates certain set of inter splitting modes according to intermediate encoding information. The spatio-temporal correlations among neighboring blocks are also studied in some works [6-7]. In JVET- D0095, the authors proposed to adjust the *MaxBTDepth* adaptively for each frame according to the temporal level [8]. In [9], Wang *et al.* proposed a local constrained QTBT algorithm including dynamic partition parameters derivation method and the limited binary tree partitioning method. These works are efficient in reducing the computational complexity. However, they highly depend on the statistical information and hand-crafted features, which may

978-1-4799-7061-2/18/$31.00 ©2018 IEEE

Table 1. Percentage of max QTBT depth for 128x128 CUs.

Layer	QP	*MaxDepth*										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	22	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0	4.4	2.4	92.5
	27	0.4	0.0	0.4	0.0	0.0	0.0	1.2	2.4	5.2	4.0	86.5
	32	0.8	0.4	1.6	0.0	0.4	0.8	1.2	3.6	9.5	4.4	77.4
	37	3.2	0.0	2.8	0.0	1.2	2.0	2.8	4.4	10.7	9.1	63.9
1	22	1.8	0.4	0.4	0.0	0.4	2.2	3.1	2.7	5.4	5.8	77.7
	27	6.7	0.0	2.2	0.4	2.2	2.7	2.7	2.2	8.9	8.0	63.8
	32	12.9	0.9	2.2	0.0	4.5	1.8	5.8	2.2	15.6	11.6	42.4
	37	18.8	0.4	0.9	1.3	4.9	2.7	8.5	7.1	23.2	11.6	20.5
2	22	3.6	0.2	2.0	0.0	1.6	4.5	5.8	2.7	3.3	3.1	73.2
	27	7.6	0.9	4.0	4.0	5.4	1.3	1.1	1.3	7.6	8.0	58.7
	32	18.3	0.0	2.0	1.6	5.1	2.0	4.5	4.2	13.2	14.1	35.0
	37	22.1	0.0	3.1	1.8	6.0	2.5	7.6	9.4	20.3	15.6	11.6
3	22	11.3	0.3	3.3	2.3	5.9	2.5	1.7	1.2	6.1	7.4	57.9
	27	19.4	0.2	4.1	1.5	5.2	0.7	2.5	5.1	9.9	15.0	36.4
	32	24.1	0.6	4.4	0.7	6.1	3.9	8.8	8.8	14.3	15.8	13.2
	37	27.0	0.6	4.7	0.9	13.5	8.4	13.3	14.5	12.3	4.5	0.4
Average		**11.1**	**0.3**	**2.4**	**0.9**	**3.9**	**2.4**	**4.5**	**4.5**	**10.6**	**8.8**	**50.7**

Table 2. Percentage of max QTBT depth for 64x64 CUs.

Layer	QP	*MaxDepth*										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	22	0.0	0.0	9.6	0.0	0.2	0.6	1.5	3.1	11.5	5.0	68.6
	27	0.4	0.0	12.3	0.1	0.4	1.7	2.5	5.3	13.5	6.9	56.9
	32	0.8	0.4	16.1	0.4	2.5	2.3	3.0	5.1	16.1	6.1	47.4
	37	3.2	0.4	19.1	1.1	2.3	3.1	6.1	6.6	15.5	8.4	34.2
1	22	1.8	0.4	15.8	0.4	2.9	3.0	4.1	6.1	10.2	6.8	48.3
	27	6.7	0.0	22.0	0.9	4.1	3.1	3.8	5.9	13.7	7.6	32.1
	32	12.9	0.9	20.3	4.2	5.4	3.6	8.9	4.7	13.8	8.3	17.0
	37	18.8	0.4	20.8	3.8	7.0	4.9	11.7	6.7	13.8	6.1	5.9
2	22	3.6	0.2	20.8	0.2	3.8	5.2	5.2	4.6	9.4	5.8	41.1
	27	7.6	0.9	25.2	3.0	6.3	2.7	4.5	3.2	10.5	8.1	28.0
	32	18.3	0.1	22.9	2.5	7.0	3.1	7.0	5.6	12.1	8.2	13.1
	37	22.1	0.1	24.6	3.3	8.4	5.1	8.7	6.8	10.8	6.6	3.5
3	22	11.3	0.4	24.6	3.0	7.4	2.7	4.2	2.7	9.0	7.5	27.1
	27	19.4	0.2	25.9	1.9	7.4	3.3	5.1	4.6	10.0	8.9	13.3
	32	24.1	0.8	25.8	2.7	9.7	5.9	7.3	5.9	7.8	6.5	3.5
	37	27.0	1.5	27.0	3.9	14.3	7.4	7.7	5.7	4.2	1.1	0.1
Average		**11.1**	**0.4**	**20.8**	**2.0**	**5.6**	**3.6**	**5.7**	**5.2**	**11.4**	**6.7**	**27.5**

not be able to comprehensively reveal the statistics of natural video and the behaviors of the codec.

Recently, the convolution neural network (CNN) has been found to be an effective method in determining the behavior of the codec. In [10], a CNN based CU partition mode decision algorithm for HEVC intra coding was proposed. For the purpose of sharing the same CNN architecture, 32x32 and 16x16 CUs are subsampled to 8x8 matrices. In [11], Xu *et al.* represented the CU partition of an entire CTU in the form of a hierarchical CU partition map (HCPM), and established an early-terminated CNN architecture for learning to predict the HCPM. However, due to the new philosophy of the QTBT structure and the elimination of further splitting after the CU partitioning, existing fast algorithms for HEVC cannot be directly applied to QTBT structure. These challenges motivate us to develop a novel fast QTBT decision algorithm based on convolution neural network (CNN). To be best of our knowledge, it is the first framework to speed up QTBT interframe coding based on CNN.

Table 3. Percentage of max QTBT depth for 32x32 CUs.

Layer	QP	*MaxDepth*										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	22	0.0	0.0	9.6	0.1	9.9	2.6	3.5	6.6	16.6	5.3	45.8
	27	0.6	0.0	11.9	0.7	11.6	5.6	5.9	7.8	16.2	6.1	33.6
	32	0.8	0.4	16.2	0.8	16.7	4.7	5.4	7.6	16.0	6.2	25.2
	37	3.2	0.4	19.1	2.3	17.0	5.9	6.7	8.6	15.5	5.5	15.7
1	22	1.8	0.4	15.8	0.7	18.9	4.8	5.3	7.8	13.4	5.7	25.3
	27	6.4	0.0	24.6	3.1	20.9	5.5	5.3	5.6	10.0	5.2	13.3
	32	12.9	0.9	20.3	7.0	19.8	5.9	8.0	5.8	9.5	4.2	5.7
	37	18.8	0.4	20.8	8.0	20.2	7.1	8.8	5.0	7.0	2.2	1.7
2	22	3.6	0.2	20.8	0.5	21.6	6.3	5.8	6.3	10.8	4.8	19.3
	27	9.6	1.4	27.0	6.5	21.3	3.4	3.9	4.4	8.3	4.3	10.0
	32	18.3	0.1	22.9	5.4	21.5	5.3	6.3	5.2	7.3	3.4	4.2
	37	22.1	0.1	24.6	6.9	22.5	5.3	6.1	4.4	4.8	2.1	1.0
3	22	11.3	0.4	24.7	5.3	22.3	4.4	5.2	3.9	7.5	4.4	10.5
	27	23.6	0.4	27.5	3.0	22.0	3.3	4.1	3.7	5.4	3.5	3.7
	32	24.1	0.8	25.8	5.9	23.6	5.0	4.8	3.4	3.4	2.2	0.9
	37	27.0	1.5	27.4	8.1	23.6	4.8	3.9	2.0	1.3	0.3	0.0
Average		**11.5**	**0.5**	**21.2**	**4.0**	**19.6**	**5.0**	**5.6**	**5.5**	**9.6**	**4.1**	**13.5**

Table 4 Designed classifier for prediction the depth range.

Class labels	*MaxDepth*	Texture description
0	0	Very flat
1	2	Flat
2	4	Medium
3	6	A certain texture
4	8	High texture
5	10	Deep texture

2. PROPOSED CNN BASED FAST QTBT PARTITION

In this section, we first analyze the QTBT in a statistical way, which effectively guides us to design the architecture of the CNN. Subsequently, the CNN architecture and the training method are detailed. Finally, the false prediction risk is controlled based on temporal correlation to further improve the robustness of the scheme.

2.1. Statistical analyses of QTBT

During the encoding process, QTBT produces more partition shapes than HEVC, including 128x128, 128x64, 64x64, 64x32, 32x32, 32x16, 128x32, 64x16, etc. However, the CNN based classification may not be appropriate for all CU sizes, as meaningful features need to be extracted in a data-driven manner. Rather than predicting the splitting mode on each depth level, we model the depth of QTBT partitioning as a multi-classification problem. In particular, the QTBT depth is computed by

$$Depth_{cu} = 2 \times QTDepth_{cu} + BTDepth_{cu}. \qquad (1)$$

According to the QTBT parameters setting, *MinQSize* = 16x16, *MaxBTDpth* = 4, the QTBT depth varies in (0, 10).

The distribution of the maximal QTBT depth (*MaxDepth*) for 128x128, 64x64 and 32x32 CUs are showed in Tables 1-3, where "QP" and "Layer" are the quantization parameter and the temporal layer characterizing the reference relationship, respectively. From Table 1-3, it is obviously observed

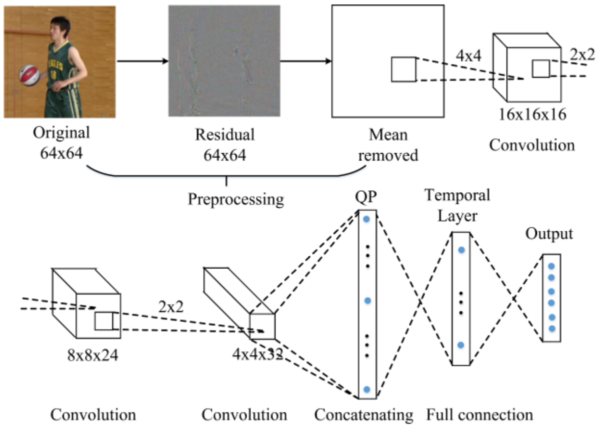

Fig. 2. Architecture of the CNN for QTBT fast partitioning.

that the CU tends to be partitioned into deeper depth for lower QP or lower temporal layer, and vice versa. One can also discern that majority CUs are partitioned with less than depth 10, which implies that time saving can be achieved by restraining the depth range and excluding the unnecessary partition iterations. However, if we directly predict the depth range of a whole CTU (128x128), it is observed from Table 1 that the average proportion of *MaxDepth* has a serious unbalanced distribution. In particular, the scarce distribution for *MaxDepth* ranging from 2 to 7 make the straightforward training with those data unpractical. It is also found that almost 50.7% CTUs demand the max QTBT depth which cannot provide any time saving. Table 2 and 3 show the percentage of *MaxDepth* for 64x64 and 32x32 CUs, respectively. It is clearly observed that the distribution is relatively homogenous, which are more appropriate to be trained. However, it is worth noting that the partitioning iterations between 128x128 and 32x32 cannot to be simplified if 32x32 CUs are trained for classification. Based on the above considerations, 64x64 CU is finally selected for CNN classification, which serves as the foundation of the proposed scheme. To avoid *over-fitting* for the categories with scarce percentage, such as “1” and “3”, we merge these categories into the adjacent categories to generate more reliable data, as shown in Table 4.

2.2. CNN architecture and training

In this work, we design a single network to predict the depth range of QTBT partitioning for 64x64 CUs. The architecture of the network is shown in Fig. 2. We first apply motion compensation for the original block to convert it into residual block. The reason of such process lies in that the partitioning of inter coding mainly depends on the correlations between the current block and the reference blocks rather than the original signal. Then, the residual block is subtracted by the mean intensity values. After pre-processing, 4x4 kernels at the first convolutional layer is used to extract the low level features. For the second and third layers, feature maps are further convoluted twice with 2x2 kernels. The final feature map is concatenated together and flatten into a vector. In the following full connection layers, the features vector, the supplemental factors QP, and temporal layer are nonlinearly fused in the fully connected layer for the final classification. The loss function evaluated based on the classification accuracy is formulated to optimize the neural network,

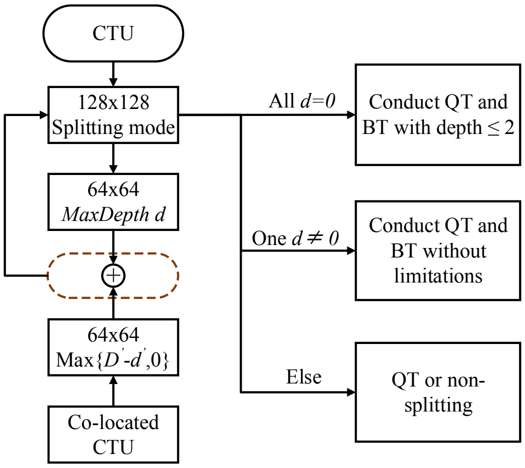

Fig. 3. Flowchart of the proposed scheme.

$$w = \underset{w}{\operatorname{argmin}}\{\frac{1}{2}w^T w + (L - l)^2\} \qquad (2)$$

where w represents the weight matrix of the network, L and l are the actual and predicted labels, respectively.

Training samples are collected with five sequences (*BasketballPass*, *BQMall*, *Johnny, Cactus*, and *ParkRunning3*) of different resolutions and characteristics. They were encoded with the JEM7.0 [12] reference software. Moreover, we eliminate such samples for which there is little RD cost difference between the optimal result and non-splitting case, since such samples may make the nets get trapped in ill-conditions during the network training.

2.3. Fast QTBT partitioning decision scheme

In the proposed scheme, the QTBT partitioning decision is modelled as a multi-classification problem, and the depth range of 64x64 CUs is predicted by the developed network. To control the risk of false prediction, we utilize the temporal correlations among consecutive frames to modify the predicted depth range. The flowchart of the proposed scheme is shown in Fig. 3.

Step 1: The current CTU is divided into four 64x64 patches directly, and the *MaxDepth* d_i ($i = 0,1,2,3$) is predicted for each patch with the neural network.

Step 2: Same processing in Step 1 is applied to the co-located CTU in the reference frame. As such, the actual and predicted *MaxDepth* of each co-located patch which are denoted as D_i' and d_i' can be obtained.

Step 3: Modify the predicted *MaxDepth* as

$$d_i = d_i + \max\{D_i' - d_i', 0\} \qquad (3)$$

When the co-located patch is precisely predicted, it is inferred

Table 5. Performance comparisons with the state-of-the-art methods.

Class	Sequence	Proposed			JVET-D0095		LC－QTBT [9]	
		BD-Rate	ΔET	NetT	BD-Rate	ΔET	BD-Rate	ΔET
A	*Campfire*	+0.66%	-40.6%	2.4%	+0.53%	-18.2%	+0.72%	-26.4%
	CatRobot1	+0.76%	-37.2%	5.1%	+0.57%	-19.4%	+0.61%	-22.3%
	FoodMarket4	+0.40%	-27.2%	2.0%	+0.51%	-20.2%	+0.37%	-16.9%
B	*BasketballDrive*	+0.53%	-35.5%	1.7%	+0.37%	-16.5%	+0.50%	-20.7%
	BQTerrace	+0.73%	-39.9%	1.8%	+0.40%	-17.6%	+0.48%	-21.0%
	MarketPlace	+0.68%	-38.8%	3.6%	+0.37%	-16.8%	+0.61%	-24.2%
	RitualDance	+0.55%	-32.6%	4.2%	+0.72%	-23.4%	+0.44%	-18.2%
C	*BasketballDrive*	+0.73%	-41.4%	1.5%	+0.24%	-10.2%	+0.52%	-23.8%
	PartyScene	+0.54%	-34.6%	2.2%	+0.37%	-18.6%	+0.48%	-22.0%
	RaceHorsesC	+0.47%	-30.7%	1.7%	+0.72%	-21.5%	+0.66%	-30.1%
D	*BQSquare*	+0.44%	-26.0%	2.1%	+0.13%	-15.4%	+0.63%	-28.4%
	BlowingBubbles	+0.60%	-35.7%	1.9%	+0.40%	-17.2%	+0.71%	-32.1%
	RaceHorses	+0.51%	-36.3%	2.7%	+0.55%	-18.2%	+0.25%	-10.8%
E	*FourPeople*	+0.32%	-28.9%	4.1%	+0.52%	-17.3%	+0.32%	-14.2%
	KristenAndSara	+0.38%	-33.8%	3.6%	+0.56%	-18.7%	+0.41%	-16.3%
	Average	**+0.55%**	**-35.0%**	**2.7%**	**+0.46%**	**-17.6%**	**+0.51%**	**-21.7%**

that the network is reliable and d_i keeps unchanged. If d_i' is predicted to be larger than its actual value, d_i of the current patch is also unchanged to ensure enough partitioning depth. Otherwise, when d_i' is predicted to be smaller than its actual value, the prediction difference will be added to d_i'. In this manner, the risk of false prediction, especially predict smaller value which results in decreasing the coding performance, can be controlled.

Step 4: According to the outputs of four patches, the 128x128 CU iterates different splitting modes. If all d_i are zero, 128x128 CU conducts further partitions with depth less than 2, and the optimal shape will be selected. If only one $d_i \neq 0$, the 128x128 CU will conduct all iterations while the patches with $d_i = 0$ will be early terminated. Otherwise, the CTU will be directly partitioned by QT and each 64x64 CU iterates according to its corresponding depth range.

3. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, the proposed scheme is implemented into JVET reference software JEM-7.0, and tested using Random Access (RA) configuration. The experiments are conducted on all the common test sequences during the JVET development expect for the sequences used for training. The coding performance is evaluated using the Bjontegaard-Delta (BD) and delta encoding time (ΔET) is used to measure the time saving. Meanwhile, the time proportion of CNN classifier (NetT) to the whole encoding time is also measured.

The simulation results are provided in Table 5. Compared with the anchor JEM-7.0, it is observed that an averaged 35% time saving can be achieved with 0.55% negligible BD-rate increase. One can also discern that consistent performance is obtained for different sequences, which proves the effectiveness and robustness of the proposed algorithm. With respect to the computational burden of CNN classifier in the proposed framework, we can clearly see that the "NetT" time is 2.7% on average. This computational burden is fully acceptable compared with the overall time.

To further verify the performance of the proposed framework, we compare our scheme with the methods that reduce the complexity of QTBT in JVET-D0095 [8] and LC－QTBT [9]. The comparison results show that our proposed algorithm can reduce the encoding complexity by around twice as much as the other methods. With more time saving obtained, similar BD coding performance loss is observed compared to these works. These results further provide evidence that the proposed method is effective in optimizing the QTBT process.

4. CONCLUSION

In this paper, we propose a CNN oriented fast QTBT partitioning decision algorithm for inter coding. We analyze the QTBT in a statistical way, which effectively guides us to design the architecture of the CNN. Furthermore, the false prediction risk is controlled based on temporal correlation to improve the robustness of the scheme. Experimental results show that the proposed scheme can consistently achieve promising performance for various video contents.

ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China (61571017, 61632001), Natural Science Foundation of Guangdong Province, China (2017A030310576)，Top-Notch Young Talents Program of China, which are gratefully acknowledged.

REFERENCES

[1] J. An, H. Huang, K. Zhang. Quadtree plus binary tree structure integration with JEM tools, JVET-B0023, Joint Video Exploration Team (JVET). Feb. 2016.

[2] R. H. Gweon, Y.-L Lee, and J. Lim. Early termination of CU encoding to reduce HEVC complexity, JVTVC-F045, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC). Jul. 2011.

[3] Shen L, Zhang Z, An P. "Fast CU size decision and mode decision algorithm for HEVC intra coding," IEEE Transactions on Consumer Electronics, 2013, 59(1): 207-213.

[4] Z. Wang, S. Wang, J. Zhang, S. Ma. "Adaptive Progressive Motion Vector Resolution Selection Based on Rate–Distortion Optimization," IEEE Transactions on Image Processing, 26(1): 400-413, 2017.

[5] J. Vanne, M. Viitanen, T. D. Hämäläinen. "Efficient mode decision schemes for HEVC inter prediction," IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, pp. 1579-1593, 2014.

[6] M. Zhang, C. Zhao, J. Xu. An adaptive fast intra mode decision in HEVC. Conn.: IEEE International Conference on Image Processing (ICIP), 2012.

[7] L. Shen, Z. Zhang, Z. Liu. "Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations," IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, pp. 1709-1722, 2014.

[8] Y. Yamamoto. AHG5: Fast QTBT encoding configuration, JVET-D0095, Joint Video Exploration Team (JVET). Oct. 2016.

[9] Wang Z, Wang S, Zhang J, et al. Local-constrained quadtree plus binary tree block partition structure for enhanced video coding. Conn.: IEEE Visual Communications and Image Processing (VCIP), 2016: 1-4.

[10] Liu Z, Yu X, Gao Y, et al. "CU partition mode decision for HEVC hardwired intra encoder using convolution neural network," IEEE Transactions on Image Processing, 2016, 25(11): 5088-5103.

[11] Xu M, Li T, Wang Z, et al. "Reducing Complexity of HEVC: A Deep Learning Approach," arXiv preprint arXiv:1710.01218, 2017.

[12] JVET software repository. Available online: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/