

CNN-Based Fast HEVC Quantization Parameter Mode Decision

Liming Chen¹, Bosi Wang^{1,*}, Weijie Yu¹ and Xu Fan¹

Abstract: With the development of multimedia presentation technology, image acquisition technology and the Internet industry, long-distance communication methods have changed from the previous letter, the audio to the current audio/video. And the proportion of video in work, study and entertainment keeps increasing, high-definition video is getting more and more attention. Due to the limits of the network environment and storage capacity, the original video must be encoded to be efficiently transmitted and stored. High Efficient Video Coding (HEVC) requires a large amount of time to recursively traverse all possible quantization parameter values of the coding unit in the adaptive quantization process. The optimal quantization parameter is calculated by comparing the rate distortion cost. In this paper, we propose a fast decision method for HEVC quantization parameters selection based on convolutional neural network, which saves video's encoding time.

Keywords: HEVC, convolutional neural network, quantization parameter.

1 Introduction

With the development of multimedia imaging technology, image acquisition technology and the Internet industry, the methods of people's long-distance communication have changed from the previous letters to audio-video combination. Video plays an increasingly high proportion in work, learning and entertainment. In addition, high-definition video can provide a clearer and more realistic image, which plays a vital role in the development of various industries such as conference format, monitoring accuracy and so on Sullivan et al. [Sullivan, Ohm, Han et al. (2012)]. However, as video clarity and resolution increase, video content requires more bits for storage [Pourazad, Doutre, Azimi et al. (2012); Zhu, Li and Chen (2013)]. Efficient compression is the key for ultra-high definition video to be widely used in the market. In response to cope with the constantly changing Video Coding requirements, the International Joint Video Team released HEVC (High Efficiency Video Coding) [Sector ITU-T S (2013)].

The new generation of standards has been deeply updated in technology, such as using quadtree algorithm to improve the partitioning level of coding tree units in the coding structure, using larger resolution and asymmetric prediction units, multi-reference frame motion estimation and so on. In addition, block-based motion compensation is used in HEVC inter-frame prediction. HEVC can calculate the offset distance of the best

¹ School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China.

* Corresponding Author: Bosi Wang. Email: bosi@nuist.edu.cn.

matching blocks in the spatial and temporal domains between continuous frames. That is the motion vector MV. Predicting the current MV by choosing neighboring MVs in the airspace or time domain. Then the prediction residual of MV is encoded which can save a large amount of MV encoded bits. HEVC proposes two new technologies in MV prediction, including AMVP and Merge. Both new techniques establish a candidate MV list and then select the MV with the lowest rate distortion cost as the predicted MV of the current PU. The difference between the two is primary that the Merge mode mainly transmits the index of the candidate PU block and does not need to transmit the MV information. The AMVP mode needs to transmit the MV information of the current encoded PU for inter-frame prediction. Secondly, the candidate list lengths are different. The candidate list length of the Merge mode is 5, while the candidate list length of the AMVP mode is only 2. The video compression efficiency of H.265/HEVC has been significantly improved after ensuring that the quality of video compression is not different from that of the HD video coded by the advanced video coding standard H.264 [Shen, Li and Zhu (2013)].

The transform and quantization module in HEVC are mainly used to calculate the correlation coefficient according to image content, so as to reduce the redundancy of the image content and compress the video data more efficiently. For the video coding quantization process, the coding unit needs to spend a lot of time on the optimal adaptive quantization parameter selection. This process is modeled as the classification of quantization parameter of coding units, and the pre-trained network model directly predicts different types of coding blocks. Optimal adaptive quantization parameters. Finally, it can be realized that the video coding complexity is optimized under the condition that the video quality and the code rate after video compression are not significantly different from the original coding.

The purpose of this paper is to solve the problem of high computational complexity of optimal quantization parameter decision in video coding process. A fast quantization parameter decision-making method based on convolutional neural network is studied. Firstly, the basic quantization parameters of the original high-efficiency video coding standard code are specified. According to the method provided by the code, the optimal quantization parameter offset is obtained by recursively calculating the rate-distortion cost of all CUs contained in each frame of video. The CU image is in one-to-one correspondence with the corresponding offset as the training set of the convolutional neural network to participate in the classification training of the convolutional neural network. The original optimal quantization parameter calculation process is replaced by the trained model and related codes. The experimental results show that compared with the method of selecting the optimal quantization parameter in HEVC, the coding time of the proposed method is reduced by 34% on average, and the loss of code rate and PSNR is basically negligible.

The remainder of this paper is organized as follows. Section 2 introduces the related works. The details of the proposed work are presented in Section 3. Experimental results are given in Section 4. Finally, Section 5 concludes this paper.

2 Related works

Currently, video coding is attracting increasing interest from academia, research institutions and large companies. In the progress of video coding complexity optimization, according to different optimization modules, it can be divided into parts: HEVC intra module complexity optimization and HEVC inter module complexity optimization, which is used to eliminate redundant information between single frames or multiple frames in a video. There are two important directions which need people to optimize, one is to optimize the size of different prediction unit, and the other is to reduce the intra prediction direction. According to the spatiotemporal information of adjacent coding units, Tang et al. [Tang, Jing, Chen et al. (2017)] reduced the traversal range of the CU, and optimize the prediction mode selection of other CUs by judging whether the current best prediction mode is the planar mode. Tian et al. [Tian and Goto (2012)] proposed a high efficiency intra PU selection algorithm, which calculates the content information of the coding tree unit and its sub-coding units according to the algorithm. And then they determined whether to directly proceed to the next round of PU mode selection process. Belghith et al. [Belghith, Kibeya and Ayed (2016)] used the Sobel operator to detect the edge of the CU and analyze the texture content of the CU. If the content of the CU is simple, the encoding is performed directly at the current depth. Otherwise, if the content of the current CU is complex, the division will be continued. Yao et al. [Yao, Li and Lu (2016)] used the pixel arrangement information of the texture to select different coding modes for different PUs. Min et al. [Min and Cheung (2015)] split CUs of different sizes by analyzing the texture features of video frames. Qi et al. [Qi, Zhu and Yang (2014)] used the Sobel operator to calculate the texture direction information according to the value of the image pixel and the spatial correlation, and select the intra mode to predict. Shen et al. [Shen, Zhang and Liu (2014)] proposed a fast intra selection algorithm based on the spatiotemporal relationship between texture information and video images, and obtained texture information by calculating the mean absolute error. The optimization of inter prediction is to better select CU and PU. In Kim et al. [Kim, Yang, Won et al. (2012)], motion vector, relationship of PU under skip mode and corresponding residual are treated together as a model. In this way, the complexity of the PU decision process can be reduced. Shen et al. [Shen, Liu and Zhang (2013)] utilized the space-time domain information to determine the CU depth range of the coding tree unit. They dynamically adjust CTU levels by skipping or prematurely terminating infrequently used CU depths. By counting the rate distortion cost of CU encoded by skip mode, Kim et al. [Kim, Jeong and Cho (2012)] established a model to predict the current CU. Feng et al. [Feng, Dai, Zhao et al. (2017)] used the motion information of CU at the current depth to judge the CU division of the same area in adjacent frames. This approach can reduce the number of decision schemes of prediction mode by calculating the depth of different CU.

At present, the methods of video coding optimization can be divided into two categories: statistical based methods and machine learning based methods. Statistical based methods are to prematurely terminate or skip unnecessary patterns based on statistical information. Lee et al. [Lee, Shim, Park et al. (2015)] proposed to use the distortion characteristics of the merge mode to determine the use of skip mode to skip unnecessary mode. Zupancic et al. [Zupancic, Blasi, Peixoto et al. (2016)] proposed an adaptive method of reverse checking CU from the bottom up according to the coding information of higher CU depth.

Jung and Park adopted an adaptive method to accelerate the HEVC coding process by utilizing data of RD cost and bit rate. Jung et al. [Jung and Park (2016); Choi and Jang (2016)] used a fast TU decision algorithm based on non-zero discrete cosine transform coefficients, which reduces complexity by trimming quadtrees. Lee et al. [Lee, Kim, Lim et al. (2015)] proposed a fast CU decision algorithm, which referred to skip mode decision, CU skip estimation and early CU termination algorithms, and used Bayesian decision theory to determine the CU termination threshold. Xiong et al. [Xiong, Li, Meng et al. (2015)] proposed a rapid decision-making algorithm based on absolute difference estimation. Ahn et al. [Ahn, Lee and Kim (2015)] proposed a fast and efficient CU coding method, which uses texture parameters such as sample adaptive offset, MV and TU size to estimate texture complexity and time complexity. All of the above approaches are based on statistical analysis that prematurely terminates or overpasses unrelated checks of CUs/PUs/TUs. This may limit its applicability to other sequences.

From the perspective of machine learning, mode decision of video coding process can be regarded as a classification problem. For example, CU partitioning in HEVC can be considered as a binary classification task. Existing machine learning algorithms are used to predict the size of CU, PU, or TU in HEVC. Shen et al. [Shen, Zhang and Zhang (2015)] used Bayesian decision theory to map the variance of residual coefficient to TU size. Kim et al. [Kim and Park (2016)] proposed a CU premature termination algorithm based on Bayesian decision theory. In Correa et al. [Correa, Assuncao, Agostini et al. (2015)], Correa et al. used decision tree to predict CU size. Zhang et al. [Zhang, Kwong, Wang et al. (2015)] designed a triple output joint classifier and a flexible CU depth decision structure. Alencar et al. [Alencar and De Oliveira (2016)] proposed a fast CU decision method based on Pegasos algorithm, which terminated the CU division process through online learning. Zhu et al. [Zhu, Zhang, Li et al. (2016)] designed a decision function based on machine learning to control prediction accuracy. Peixoto et al. [Peixoto, Shanableh and Izquierdo (2014)] constructed a new H.264/AVC to HEVC conversion architecture. They used H.264/AVC coding parameters to determine the CU partition mode of HEVC coding standard. In detail, they mapped the H.264/AVC coding parameters to the CU partition of HEVC coding standard by using linear discriminant function. These approaches predicted CU, PU, and TU by using machine learning algorithms. However, these algorithms only use weak classifiers to implement mode decision. Too many wrong classifications may result in poor RD performance and no reduction in complexity.

Most of the complexity optimization algorithms mentioned above concentrated on the complexity optimization technologies of CUs/PUs in the intra/inter prediction module of HEVC. Researchers often rely on subjective inference to address complex computer vision problems. This behavior tends to ignore implicit but useful features. For the quantization module, the above method still uses the recursive search method to select the optimal quantization parameter in the quantization process. The process of calculating the optimal quantization parameter occupies a large proportion in the whole coding period. Since video coding efficiency is affected, there is a need to optimize the process of optimal quantization parameter selection.

3 Proposed method

In this section, we describe a fast decision method for the selection of H.265/HEVC quantization parameters based on convolutional neural network.

3.1 Problem formulation of QP selection

HEVC standard reference software-HM uses two ways to calculate quantitative parameters in the process of quantitative. One is to use traditional calculation method. Firstly, specify a basic QP based quantitative parameters, then calculate the offset value according to the complexity of different CUs. In final, quantization parameter is obtained by adding the basic QP and quantization parameter offset. This method is fast to calculate, but the subjective quality of encoded video is poor, the quantization parameters corresponding to the coding unit are not the best. The other one is to use the adaptive method calculating optimal quantization parameters. Modify the HM encoding configuration file, specify the range of the quantitative parameters of the offset value between $-7\sim 7$. Each CU from 64×64 to 8×8 recursive traversal of all possible quantitative parameters, calculate the rate-distortion cost. The optimal quantitative parameters are calculated by comparing the fifteen quantitative parameters of different rate-distortion cost. Not only the quality of the video encoded using this method is better than the former, but the bit rate reduced. However, this method needs much more time in calculate QPs.

3.2 Our method

This paper proposes to simplify the recursive traversal method of the optimal quantization parameter in HM to the image classification problem using convolutional neural network. The training model directly derives the quantization parameter value. So the final quantization parameter is obtained, and the original Quantization Parameter Calculation Module of HM is replaced, as shown in Fig. 1. Since most of the time of HEVC encoded video is concentrated on quantization parameters calculation, the final coding time in an ideal state can be saved by about 14/15.

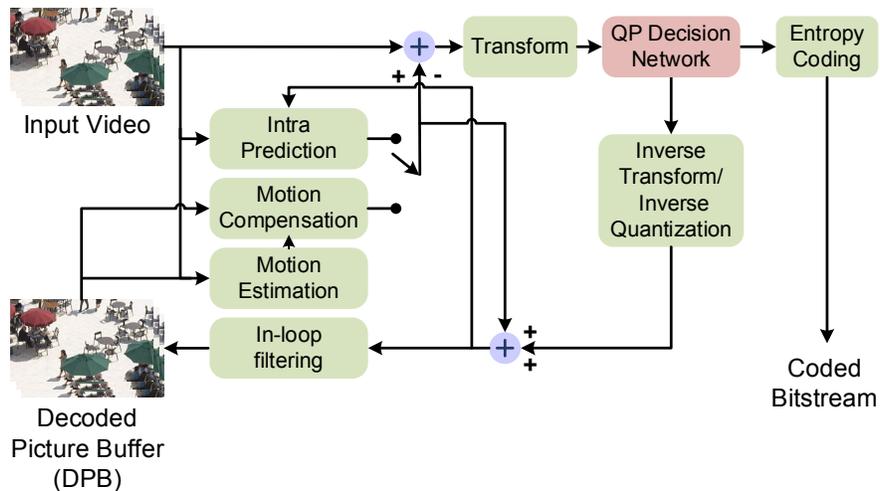


Figure 1: Improved method in this paper

3.3 The structure of our network

We consider that if we use complex network structures, it may bring new complexity problems to the quantization module. The training model uses a simple convolutional neural network, as shown in Fig. 2. Convolution layer 1 uses 64 convolution kernels ($3 \times 3 \times 3$), the stride is set to 1, padding is set to SAME, the activation function uses ReLU. Pooling layer 1 selects the max pooling, 3×3 filter, the stride is set to 2, and performs local response normalization after pooling. Convolution layer 2 uses 16 convolution kernels ($3 \times 3 \times 64$), padding is also set to SAME, and the activation function uses ReLU. The pooling layer 2 also selects the max pooling, using 3×3 filter, the stride is set to 1, and the local response normalization operation is performed after the pooling is completed. The fully connected layer 1 converts the data output by the pooled layer into a one-dimensional list by a reshape operation, the number of nodes is set to 128, and the activation function uses ReLU. The fully connected layer 2 is also 128 nodes, and the activation function uses ReLU. The softmax regression layer outputs the previous fully connected layer and performs linear regression, then calculates the score for each class. Loss uses cross entropy loss, and the learning rate is set to 0.0001.

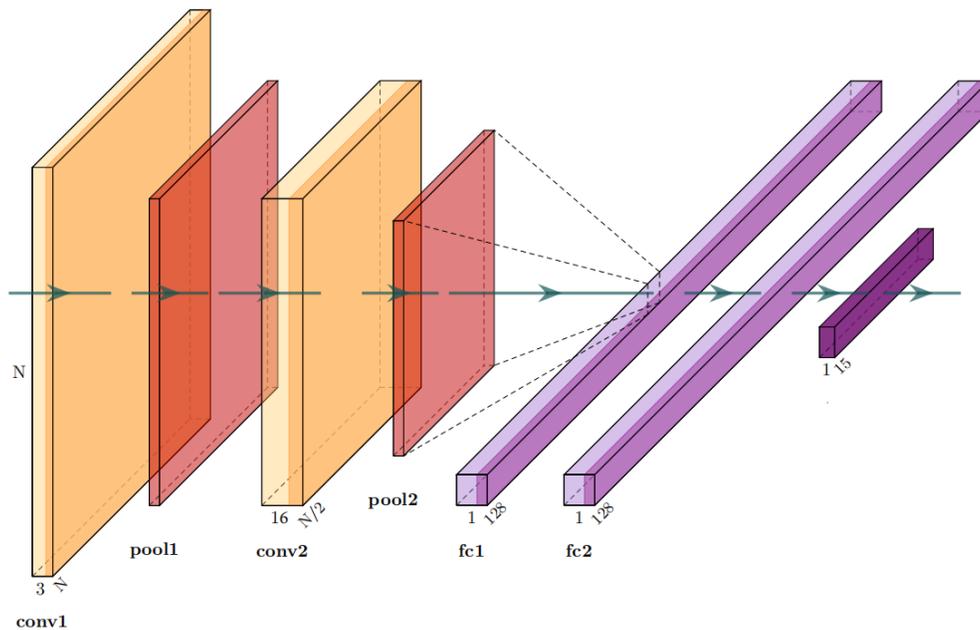


Figure 2: Network model diagram

Where N represents the size of the CU (64×64 , 32×32 , 16×16 , 8×8), and CUs of different sizes are separately trained and predicted.

3.4 Experimental results and analysis

This paper uses 15 video test sequence provided by the international video coding group to collect training data as shown in Tab. 1, and uses the intra-frame coding structure to test the performance of the HM reference software.

Table 1: Test sequence

Sequence	Resolution	Frame rate (fps)	Sequence	Resolution	Frame rate (fps)
BQTerrace	1920×1080	60	BasketballDrill	832×480	50
ParkScene	1920×1080	24	BQMall	832×480	60
BasketballDrive	1920×1080	50	PartyScene	832×480	50
Cactus	1920×1080	50	RaceHorses	832×480	30
Kimono1	1920×1080	24	BlowingBubbles	416×240	50
Johnny	1280×720	60	BasketballPass	416×240	50
FourPeople	1280×720	60	BQSquare	416×240	60
KristenAndSara	1280×720	60			

In order to increase the credibility of the results and the feasibility of the network model, the basic QP is set to 22, 27, 32, and 37 respectively. The first 200 frames of each video sequence are encoded by HM. The index of video frame, the position of each coding unit in one frame and the corresponding quantization parameter in the encode process is recorded. Since there is little difference in content between adjacent frames, Frames used to train is extracted every ten frames. Then the coding unit used to train included in the original image is clipped according to the position of the coding unit in the entire original video frame. Finally, a total of 16 different convolutional neural network models based on different basic QPs, different size of CUs, and different QP offset were trained, corresponding to the four different coding unit sizes under the four basic quantization parameters.

To ensure the authenticity and credibility of the experimental results, we replaced the original optimal quantization parameter module of HM16.0 with our proposed method results. The value of coding time, bit rate, and PSNR are based on the original HM reference software.

Video coding requires a trade-off between encoded quality, bit rate and coding time. These parameters are the basis of video coding method evaluation. In encoding process, the quantization parameters compared with optimal QPs are predicted by the trained model. It is judged whether quantization parameters calculated by our method corresponding to different coding units are the same as the HM16.0 quantization parameters. The average accuracy of optimal QPs trained by ours in this paper is 81.2%.

In order to verify the impact of our proposed method on the performance of HEVC coding, the encoding time, BDBR and BDPSNR are used as indicators for evaluating performance. The encoding time measured by considering the prediction time, hard disk reading and writing speed and other environmental factors is shown in Tab. 2.

It can be seen from Tab. 2 that the Johnny video sequence performs best in the whole test result compared with the original method. When the basic QP is set to 37, the overall time savings is about 34.56%. Our result is limited by the hardware environment, so the time only proves the feasibility of the method in this application. Considering the experimental environment, the average encoding time can save about 34.29%, which greatly improves the coding efficiency.

Table 2: Time comparison

sequence	Resolution	QP	Time(s)		Time Savings (%)
			Proposed	HM16.0	
BasketballDrill	832×480	22	381.71	551.01	30.73
		27	328.24	475.48	30.97
		32	284.87	411.79	30.82
		37	256.16	368.84	30.55
BasketballPass	416×240	22	102.62	138.98	26.16
		27	90.53	120.91	25.13
		32	80.40	105.70	23.94
		37	71.20	91.63	22.30
BlowingBubbles	416×240	22	107.43	142.92	24.83
		27	105.25	142.36	26.07
		32	95.44	128.93	25.97
		37	78.30	102.14	23.34
BQMall	832×480	22	466.15	692.50	32.69
		27	419.54	623.07	32.67
		32	372.30	552.62	32.63
		37	339.10	504.21	32.75
BQSquare	416×240	22	154.00	220.65	30.21
		27	142.72	204.50	30.21
		32	129.07	183.87	29.80
		37	116.73	164.93	29.23
Johnny	1280×720	22	920.69	1404.95	34.47
		27	841.13	1285.18	34.55
		32	772.63	1178.53	34.44
		37	746.15	1140.25	34.56
PartyScene	832×480	22	692.16	1049.00	34.02
		27	604.59	914.61	33.90
		32	545.93	828.59	34.11
		37	469.30	710.22	33.92
RaceHorses	832×480	22	562.18	852.40	34.05
		27	498.78	756.15	34.04
		32	390.63	583.16	33.02
		37	337.77	502.40	32.77
Average					34.29

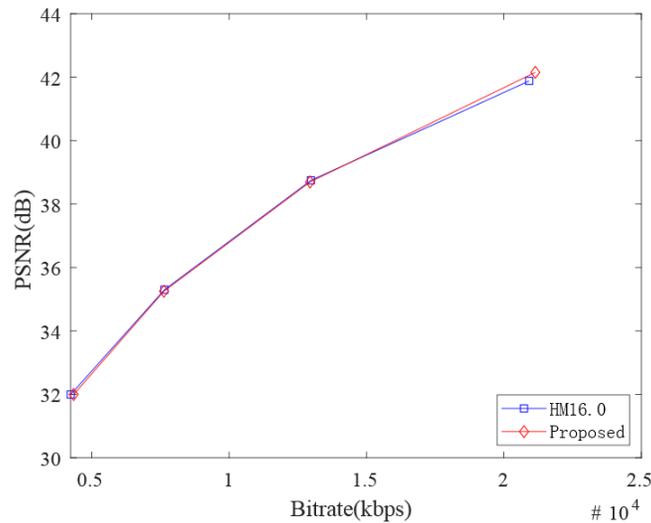
In order to verify the difference between the proposed method and the original method in terms of bit rate and encoded video quality, BDBR and BDPSNR are used as the measurement. The results are shown in Tab. 3.

Table 3: RD Performance

Video sequence	Resolution	Proposed vs. HM16.0	
		BDBR (%)	BDPSNR (dB)
Johnny	1280×720	1.70	-0.05
BQMall	832×480	0.86	-0.06
PartyScene	832×480	0.65	-0.02
RaceHorses	832×480	1.16	-0.05
BasketballDrill	832×480	1.01	-0.07
BasketballPass	416×240	0.94	-0.07
BlowingBubbles	416×240	0.72	-0.05
BQSquare	416×240	0.78	-0.07
Average		0.98	-0.05

It can be seen from the table that the loss of BDBR and BDPSNR has little to do with the video resolution. The average BDBR is increased by 0.98%, that is, the code rate is increased by 0.98%. The BDPSNR is reduced by 0.05 dB on average, that is, the video quality encoded reduced by 0.05 dB.

In order to see the difference more intuitively between code rate and video quality, the RD curve is given in this paper. As shown in Fig. 3, the blue line represents the effect of the HM16.0 optimal adaptive quantization parameter on the coding effect. The red line is the effect of the proposed method on the coding effect. It can be seen clearly that compared with the original method the loss of bit rate and video quality encoded by our proposed method are negligible.

**Figure 3:** RD curve

4 Conclusions

In order to solve the problem that UHD video cannot be widely used in daily life, the international joint coding group JCT-VC brainstormed and launched HEVC. Although the coding efficiency and other aspects of performance have exceeded the previous generation coding standard H.264. It still takes a lot of time for HEVC to select the optimal adaptive quantization parameter. In order to optimize the complexity of HEVC in the quantization parameter selection algorithm and improve the coding performance of HEVC, the convolution neural network is used in this paper to simplify the complex quantization parameter calculation problem into a convolution neural network image classification problem. The experimental results show that the fast decision method of quantization parameters proposed in this paper can save about 34% of the average video coding time compared with the adaptive optimal quantization parameter selection method in HEVC reference code, and other losses are basically neglected.

References

- Ahn, S.; Lee, B.; Kim, M.** (2015): A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 422-435.
- Alencar, M. S.; De Oliveira, J. F.** (2016): Online learning early skip decision method for the HEVC Inter process using the SVM-based Pegasos algorithm. *Electronics Letters*, vol. 52, no. 14, pp. 1227-1229.
- Belghith, F.; Kibeya, H.; Ayed, B.** (2016): Fast coding unit partitioning method based on edge detection for HEVC intra-coding. *Signal, Image and Video Processing*, vol. 10, no. 5, pp. 811-818.
- Choi, K.; Jang, E. S.** (2012): Early TU decision method for fast video encoding in high efficiency video coding. *Electronics Letters*, vol. 48, no. 12, pp. 689-691.
- Correa, G.; Assuncao, P. A.; Agostini, L. V.; Silva Cruz, L. A.** (2015): Fast HEVC encoding decisions using data mining. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 660-673.
- Feng, L.; Dai, M.; Zhao, C. L.; Xiong, J. Y.** (2017): Research of fast HEVC inter prediction coding based on motion features *Journal of Optoelectronics-Lase*, pp. 95-103.
- Jung, S.; Park, H. W.** (2016): A fast mode decision method in HEVC using adaptive ordering of modes. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 10, pp. 1846-1858.
- Kim, H.; Park, R.** (2016): Fast CU partitioning algorithm for HEVC using an online-learning-based Bayesian decision rule. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 130-138.
- Kim, J.; Jeong, S.; Cho, S.** (2012): Adaptive coding unit early termination algorithm for HEVC. *IEEE International Conference on Consumer Electronics*, pp. 182-185.
- Kim, J.; Yang, J.; Won, K.** (2012): Early determination of mode decision for HEVC. *IEEE Conference on Picture Coding Symposium*, pp. 449-452.
- Lee, H.; Shim, H. J.; Park, Y.; Jeon, B.** (2015): Early skip mode decision for HEVC

encoder with emphasis on coding quality. *IEEE Transactions on Broadcasting*, vol. 61, no. 3, pp. 388-397.

Lee, J.; Kim, S.; Lim, K.; Lee, S. (2015): A fast CU size decision algorithm for HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 411-421.

Min, B.; Cheung, R. C. C. (2015): A fast CU size decision algorithm for the HEVC intra encoder. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 892-896.

Peixoto, E.; Shanableh, T.; Izquierdo, E. (2014): H.264/AVC to HEVC video transcoder based on dynamic thresholding and content modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 1, pp. 99-111.

Pourazad, M. T.; Doutre, C.; Azimi, M.; Nasiopoulos, P. (2012): HEVC: the new gold standard for video compression: How does HEVC compare with H.264/AVC? *IEEE Consumer Electronics Magazine*, vol. 1, no. 3, pp. 36-46.

Qi, M. B.; Zhu, G. H.; Yang, Y. F. (2014): HEVC intra-prediction mode selection by using texture and spatial correlation. *Journal of Image and Graphics*, vol. 19, no. 8, pp. 1119-1125.

Sector ITU-T S (2013): *ITU-t Recommendation H. 265: High Efficiency Video Coding*. Geneva, Switzerland: Telecommunication Standardization Sector.

Shen, F.; Li, J.; Zhu, Z. (2013): High efficiency video coding. *Chinese Journal of Computers*, vol. 36, no. 11, pp. 2340-2355.

Shen, L.; Liu, Z.; Zhang, X. (2013): An effective CU size decision method for HEVC encoders. *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 465-470.

Shen, L.; Zhang, Z.; Liu, Z. (2014): Effective CU size decision for HEVC intracoding. *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4232-4241.

Shen, L.; Zhang, Z.; Zhang, X. (2015): Fast TU size decision algorithm for HEVC encoders using Bayesian theorem detection. *Signal Processing: Image Communication*, vol. 32, pp. 121-128.

Sullivan, G. J.; Ohm, J. R.; Han, W. J.; Wiegand, T. (2012): Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668.

Tang, H. Y.; Jing, Q.; Chen, Y. T.; Guo, N. (2017): Fast intra coding algorithm for HEVC. *Computer Engineering and Applications*, vol. 53, no. 18, pp. 182-185.

Tian, G.; Goto, S. (2012): Content adaptive prediction unit size decision algorithm for HEVC intra coding. *IEEE Conference on Picture Coding Symposium*, pp. 405-408.

Xiong, J.; Li, H.; Meng, F.; Wu, Q.; Ngan, K. N. (2015): Fast HEVC inter CU decision based on latent sad estimation. *IEEE Transactions on Multimedia*, vol. 17, no. 12, pp. 2147-2159.

Yao, Y.; Li, X.; Lu, Y. (2016): Fast intra mode decision algorithm for HEVC based on dominant edge assent distribution. *Multimedia Tools and Applications*, vol. 75, no. 4, pp. 1963-1981.

Zhang, Y.; Kwong, S.; Wang, X.; Yuan, H.; Pan, Z. et al. (2015): Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding. *IEEE Transactions on Image Processing*, vol. 24, no. 7, pp. 2225-2238.

Zhu, L.; Zhang, Y.; Li, N.; Jiang, G.; Kwong, S. (2016): Machine learning based fast H.264/AVC to HEVC transcoding exploiting block partition similarity. *Journal of Visual Communication and Image Representation*, vol. 38, pp. 824-837.

Zhu, X.; Li, X.; Chen, J. (2013): Next generation video coding standard-HEVC. *Journal of Nanjing University of Posts and Telecommunications (Natural Science)*, vol. 33, no. 3, pp. 1-11.

Zupancic, I.; Blasi, S. G.; Peixoto, E.; Izquierdo, E. (2016): Inter-prediction optimizations for video coding using adaptive coding unit visiting order. *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1677-1690.