JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 32, 1289-1299 (2016)

Machine Learning-based Fast Intra Coding Unit Depth Decision for High Efficiency Video Coding

ZONG-YI CHEN[1], JIUNN-TSAIR FANG[2], YEN-CHUN LIU[1] AND PAO-CHI CHANG[1]
[1]*Department of Communication Engineering*
National Central University
Taoyuan City, 320 Taiwan
[2]*Department of Electronic Engineering*
Ming Chuan University
Taoyuan City, 333 Taiwan
E-mail: pcchang@ce.ncu.edu.tw

This paper proposes a fast coding unit (CU) depth decision algorithm for intra coding of high efficiency video coding using an artificial neural network (ANN) and a support vector machine (SVM). Machine learning provides a systematic approach for developing a fast algorithm for early CU splitting or termination to reduce intra coding computational complexity. Appropriate features for training SVM models were extracted from spatial and pixel domains of the current CU. These features were classified into three types for three SVM training models at each depth, and different weights were assigned on the basis of the ANN analysis. Experimental results showed that the proposed fast algorithm saves at most 48.5% and on average 33% encoding time with a 1.55% Bjøntegaard delta bit rate (BDBR) loss compared with HM 15.0.

Keywords: coding unit (CU), fast algorithm, high efficiency video coding (HEVC), intra coding, machine learning, support vector machine (SVM)

1. INTRODUCTION

High efficiency video coding (HEVC), the latest video coding standard, was finalized by the Joint Collaborative Team on Video Coding (JCT-VC) in 2013 to meet the increasing demands for high-quality and high-resolution videos. As an extension of H.264 advanced video coding (H.264/AVC), HEVC saves half the bit rate of H.264/AVC for the same subjective video quality, thus achieving higher coding efficiency. HEVC adopts a quadtree-based coding unit (CU) structure to provide high flexibility for encoding an area with smooth or complex content. CU is the basic encoding unit and is similar to the macroblocks (MBs) used in H.264/AVC; CU includes prediction units (PUs) and transform units (TUs), and its size varies from 64 × 64 to 8 × 8 pixels, corresponding to four depths from 0 to 3, as shown in Fig. 1.

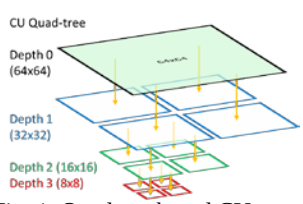


Fig. 1. Quadtree-based CU structure.

Received June 15, 2015; revised August 20, 2015; accepted September 21, 2015.
Communicated by Chung-Lin Huang.

The intra coding in H.264/AVC supports nine prediction modes: DC, planar, and seven directional modes; by contrast, HEVC intra coding supports up to 35 prediction modes: DC, planar, and 33 angular modes [1], thus considerably improving intra coding performance. However, HEVC encoders execute rate-distortion optimization (RDO) to calculate the RD costs of all CU sizes to determine the most favorable CU depth, and all possible PU/TU and intra modes are tested at each CU depth. This heuristic RDO increases computational complexity in HEVC intra coding.

Many useful features were used in existing fast algorithms [2-8] to reduce complexity and accelerate encoding in HEVC intra coding. The gradient of a CU was often utilized to measure block complexity [2, 3]. A block with complex texture was encoded with a small CU size. In addition, the mean absolute deviation (MAD) of a CU was an efficient measure of texture homogeneity [2, 8]. Shen *et al.* calculated the MAD in horizontal and vertical directions to enhance block homogeneity check [8]. Using the variance of pixels in a CU block, a fast CU size decision algorithm was developed [4]. A texture analysis method based on the variation of a pixel with respect to its local neighborhood was proposed for intra CU size selection [6]. According to strong correlations in the spatial domain, the neighboring CU depths were used to predict the possible range of CU depth for encoding [3, 5, 8]. In [7], global and local edge complexities in four directions were proposed and used to decide CU partitioning.

In most algorithms, observations and analyses are required to determine the criteria of fast decision in H.264/AVC and HEVC. Unlike a conventional rule-based method, machine learning provides a systematic approach for developing fast algorithms. Previous studies [9, 10] have used machine learning to establish criteria for MB size selection in H.264/AVC. Xiong *et al.* [11] proposed a learning-based method that uses a non-normalized histogram of oriented gradient as a feature to determine intra CU size in HEVC. In [12], a support vector machine (SVM) [13] was used for HEVC inter coding for a CU splitting early termination algorithm based on various input features, such as depth information, gradient magnitudes, sum of absolute transformed difference, and coded block flags. Moreover, RD penalties were introduced as weights for instances of misclassification during SVM model training to maintain RD performance. The learning-based method efficiently facilitated the development of a new fast encoding algorithm.

This study focused on using SVM to design a fast CU depth decision algorithm for HEVC intra coding. Appropriate features, including neighboring CU depth, boundary pixel difference, pixel variance, and number of edge points (*EPs*), were extracted from spatial and pixel domains of the current encoding CU. An artificial neural network (ANN) [14, 15] was used to analyze the effect of each extracted feature on CU size decision, and different weights were assigned to the SVM outputs. The concept of the proposed algorithm was introduced in [16]. This paper presented the details of the proposed algorithm. In addition, more simulation results were included.

The remainder of this paper is organized as follows. Section 2 presents the input features for SVM classification and the proposed SVM-based fast CU depth decision algorithm. Section 3 describes the experimental results, and Section 4 concludes the paper.

2. PROPOSED FAST INTRA CU DEPTH DECISION

This section first introduces the features used in SVM classification for predicting whether the current CU should be split. Subsequently, the effect of each feature is discussed and different weights are assigned to the features.

2.1 Feature Extraction

• *Depth of neighboring CUs*

Similar to [5], information from neighboring encoded CUs was used as a feature for training because of strong spatial correlation. The depths of left, upper, upper left, and upper right CUs were available for the current block because of the zigzag encoding order. However, according to experimental results, selecting the depths of all four neighboring CUs for SVM model training results in approximate SVM prediction accuracy compared with selecting only the depths of the left and upper CUs. Simultaneously, fewer input features accelerate the SVM prediction. Therefore, only the depths of the left (Dep_L) and upper (Dep_U) CUs were used as the features.

• *Average pixel difference on CU boundary*

Because HEVC intra coding utilizes the reference samples from the left and upper blocks for prediction, the characteristics of boundary pixels must be considered. The average absolute difference of boundary pixels was calculated to represent the similarity between current CU and reference pixels. Fig. 2 is an example of an 8 × 8 CU boundary; the differences in the left ($Diff_L$) and top ($Diff_T$) boundaries are calculated using Eqs. (1) and (2), respectively.

$$Diff_L = \frac{1}{N}\sum_{i=0}^{N-1}\left|P(x_0, y_i) - P'(x_{-1}, y_i)\right| \tag{1}$$

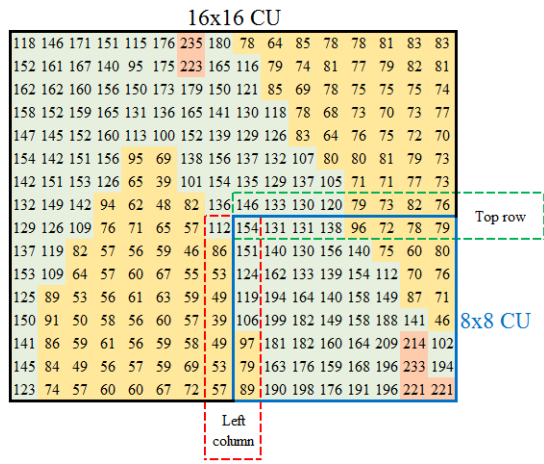


Fig. 2. Average pixel differences on CU boundaries illustrated using an 8 × 8 CU.

$$Diff_T = \frac{1}{N}\sum_{i=0}^{N-1}\left|P(x_i, y_0) - P'(x_i, y_{-1})\right| \quad (2)$$

• *Pixel variance of current CU*

As reported in [4], the variance of image pixels (*Var*) represents the content texture of each encoding block. A wider variance generally implies that more details are contained within a CU, which is likely to be split into a smaller CU size.

• *Variance of the mean of sub-CUs*

Because HEVC adopts a quadtree-based CU structure, the characteristics of the four sub-CUs may influence the splitting of the current CU. A CU tends to split if the variation among its four sub-CUs is wide. First, each sub-CU was represented using its mean. Then, the variance of these four means (Var_{sub}) was used as an SVM feature.

• *Number of edge points*

Similar to gradient calculation, edge detection by using a Sobel filter was employed to count the number of pixels which are detected as edge points. A larger number of *EPs* indicates that the CU is more complex and tends to be encoded with a smaller CU size.

2.2 Feature Analysis

An increase in the number of features increases the prediction accuracy, but it also extends the time required for predicting classification. Therefore, selecting useful features and discarding the rest is essential to maintain high prediction efficiency and accuracy. In addition, features affect the SVM prediction to varying degrees. An ANN [14, 15] was used to analyze the effect of features on classifications. Table 1 presents the results of the ANN analysis.

Table 1. Results of the ANN analysis.

Depth	0		1		2	
Class / Feature	ET	ES	ET	ES	ET	ES
Dep_L	–1.31	1.32	–1.61	1.61	–1.72	1.72
Dep_U	–1.77	1.78	–1.45	1.45	–1.55	1.55
$Diff_L$	–2.88	2.58	1.46	–1.46	–1.85	1.86
$Diff_T$	0.72	–1.02	-0.40	0.40	1.03	–1.03
Var	–16.09	15.56	–35.23	34.93	–14.28	14.24
Var_{sub}	–4.68	5.94	–1.35	1.96	6.96	–6.90
EPs	–6.64	6.79	0.27	–0.03	–0.64	0.64

ET: Early termination, ES: Early splitting

In this study, early termination means preventing the current CU from splitting into four sub-CUs and testing only the current CU in the RDO process, and early splitting means directly splitting the current CU into four sub-CUs and skipping the RD cost calculation of the current CU. The absolute values in Table 1 indicate the degree of

effect of each feature at each depth; higher absolute values indicate a higher degree of effect. The variance-type features (*Var* and Var_{sub}) are the most dominant, whereas the depth-type (Dep_L and Dep_U) and difference-type ($Diff_L$ and $Diff_T$) features are less influential. The number of *EPs* has a high effect only at depth 0; therefore, it is excluded from the SVM model training.

The characteristic of each feature clearly varied at different depths. A distinct SVM model can reasonably be applied to each depth to improve prediction accuracy. However, in some instances, the number of available features was different. For example, the current CU located at the frame boundary did not have a neighboring CU to extract features from. Thus, three SVM models with different types of input features were used to reduce SVM misclassification when the number of extracted features was not six.

Furthermore, because features varied in their degrees of effect on SVM prediction, the prediction results of the three SVM models were refined using different weights according to the results of the ANN analysis. The input features and corresponding weights are listed in Table 2. The final SVM result ($Result_{All}$) is formulated using Eq. (3).

$$\begin{aligned} &Result_{All} = 0.2 \times Result_{SVM_1} + 0.2 \times Result_{SVM_2} + 0.6 \times Result_{SVM_3}, \\ &\text{where } Result_{SVM_i} = \begin{cases} 0, & \text{if early termination} \\ 1, & \text{if early splitting} \end{cases}, i = 1, 2, 3. \end{aligned} \tag{3}$$

Table 2. Input features and weights of each SVM.

	SVM_1	SVM_2	SVM_3
Input features	Neighboring CU depth	Boundary pixel difference	Current CU & sub-CU pixel variance
Weights of SVM result	0.2	0.2	0.6

Because the number of *EPs* has a high effect only at depth 0, it was used individually for depth 0. An *EPs* threshold was set to determine whether depth 0 should be reserved for RDO, even if the final SVM result was early splitting. The threshold for *EPs* was determined to be a favorable trade-off between RD performance and time saving on the basis of extensive experiments. Several distinct thresholds (10, 20, 40, 60, 100, and 160) were set for six sequences in each class as in Table 4. The average BDBR and time saving performance are shown in Fig. 3. Smaller threshold will result in more early splitting, hence more time saving and poorer RD performance is achieved. From Fig. 3, the threshold is preferable to be 20 or 40. To achieve higher time saving, the threshold is thus set to be 20 in this study. Fig. 4 illustrates edge point checking, which is helpful in keeping the background region encoded with a large CU size as the partition selected in original HM 15.0.

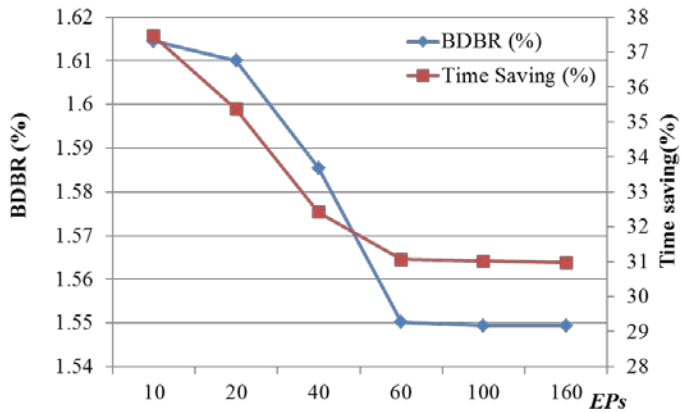
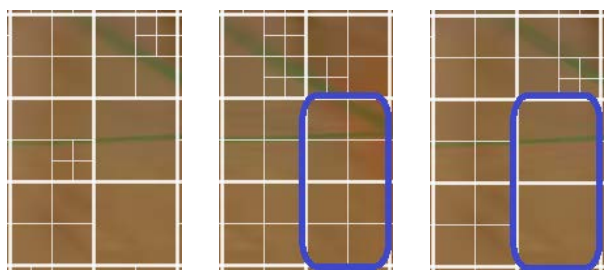


Fig. 3. The average BDBR and time saving performance for distinct EPs.



(a) (b) (c)

Fig. 4. Encoding results of BasketballDrive with QP = 32: (a) HM 15.0; (b) weighted SVM; and (c) weighted SVM with edge point checking at depth 0.

Finally, according to Table 3, whether to skip RDO under the listed conditions was decided. Fig. 5 is the flowchart of the proposed algorithm. The encoding process starts with LCU (depth 0) and extracts features from the spatial and pixel domains to determine whether early CU split occurred.

Table 3. Proposed fast CU depth decision.

Condition	Operation	
$Result_{All} \leq 0.2$	ET	Do RDO process at current depth and no splitting
$0.2 < Result_{All} < 1$	Unsure	Original HEVC encoding
$Result_{All} = 1$ && Depth = 0 && $EPs < 20$		
$Result_{All} = 1$	ES	Split into next depth without RDO process at current depth

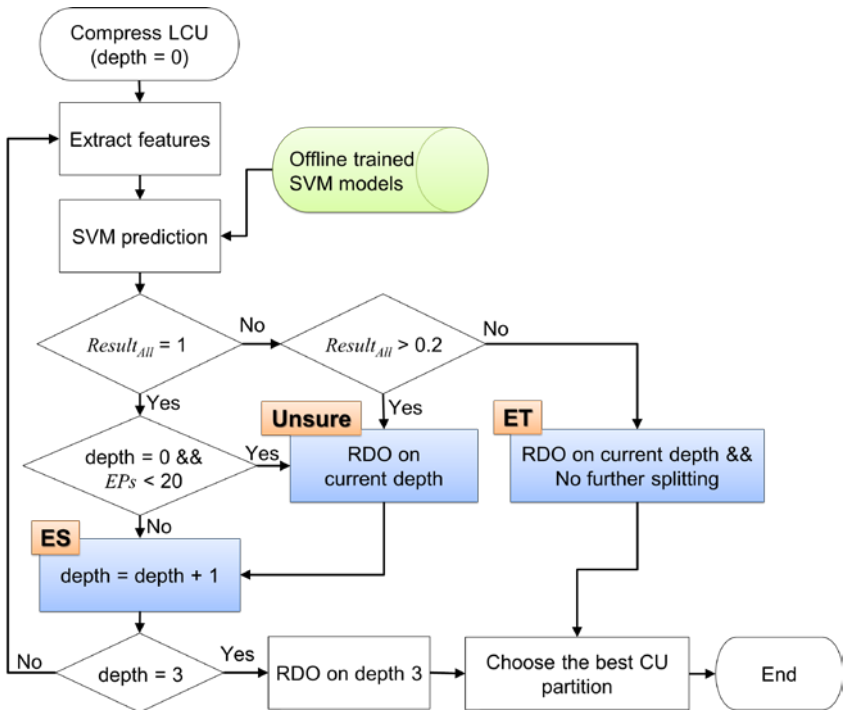


Fig. 5. Proposed learning-based fast intra CU depth decision for HEVC.

3. EXPERIMENTAL RESULTS

The proposed fast algorithm was implemented using HEVC reference software (HM 15.0) with intra_main configuration. The test platform was a PC with an Intel i7-2600 3.4-GHz CPU, 4-GB RAM, and Windows 7 professional operating system. The software for SVMs was LibSVM 3.17 [17]. LibSVM is a popular open source and commonly used in researches related to SVM. The training materials for SVM models are listed in Table 4. Twenty four frames of training data were used to train the SVM models with different input features at every depth. The kernel used in this study is radial basis function (RBF).

Table 4. Training materials for SVM models.

Reference software	LibSVM 3.17 [17]
QP	22, 27, 32, 37
Training materials (1 frame per QP per sequence)	ClassA_Traffic CalssB1_ParkScene ClassB2_BasketballDrive ClassC_RaceHorses ClassD_BQSquare ClassE_Vidyo1

Two fast CU depth decision algorithms [4, 5] were implemented on HM15.0 for comparison. For literature [5], only the algorithm of CU part was implemented. Table 5 displays the experimental results of the first ten frames of each sequence encoded with QPs 22, 27, 32, and 37. The coding efficiency was measured in terms of BDBR (%) [18]. The encoding time (including SVM prediction time) saved by the proposed algorithm (ΔT) compared with HM 15.0 is calculated using Eq. (4). The time consumed by SVM prediction during encoding is also reported in Table 5.

$$\Delta T(\%) = \frac{1}{4}\sum_{i=1}^{QP_i} \frac{Enc.Time_{HM15.0}^{QP_i} - Enc.Time_{proposed}^{QP_i}}{Enc.Time_{HM15.0}^{QP_i}} \times 100,\ QP_i = \{22,27,32,37\} \quad (4)$$

Table 5. BDBR and time saving performance.

Testing sequences	T. Nishikori [4]		L. Shen [5]		Proposed		
Class/Sequence	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	T_{SVM} (%)
A_Traffic	4.750	59.395	0.314	19.954	1.458	34.197	4.120
A_PeopleOnStreet	3.505	54.323	0.303	18.094	1.238	30.114	4.199
B1_ParkScene	4.105	60.916	0.305	22.056	1.511	37.522	3.642
B1_Kimono	17.831	69.379	0.359	35.585	1.323	31.344	4.562
B2_BasketballDrive	7.302	72.896	0.780	26.944	2.662	45.151	3.538
B2_BQTerrace	1.969	56.044	0.365	23.262	1.323	30.993	3.710
B2_Cactus	4.637	60.126	0.253	19.734	1.642	34.284	3.802
C_RaceHorsesC	2.207	51.082	0.151	15.685	1.002	28.216	3.921
C_PartyScene	0.545	41.026	0.005	17.930	0.599	23.511	3.305
C_BQMall	2.438	51.064	0.187	17.685	1.402	26.170	4.226
C_BasketballDrill	6.177	64.318	0.366	15.707	2.563	35.569	4.149
D_BQSquare	0.751	46.507	0.161	13.786	0.741	25.147	3.747
D_RaceHorses	1.957	46.630	0.085	12.470	0.686	25.414	4.867
D_BasketballPass	3.559	59.427	0.906	11.780	1.481	31.508	3.983
D_BlowingBubbles	1.121	45.305	0.015	12.155	0.872	22.872	3.964
E_Vidyo1	7.343	68.355	0.946	21.515	2.286	41.972	3.854
E_FourPeople	4.676	60.316	0.354	18.904	1.626	35.482	4.393
E_Johnny	7.465	72.802	0.825	30.073	3.280	48.474	3.379
E_KristenAndSara	5.776	69.606	0.638	25.314	1.840	45.091	3.405
Average	4.638	58.396	0.385	19.928	1.554	33.317	3.935

In all sequences, the proposed algorithm provides better BDBR in sequences, such as PartyScene and BQSquare, with complex texture uniformly distributed over the whole frame. In such sequences, the features extracted from neighboring CUs correspond to the features extracted within the current CU. In other words, all prediction results of different SVMs are likely to have the same tendency; therefore, the accuracy of the final SVM prediction is high and the encoding result is close to that of HM 15.0.

For sequences with strong motion in some parts of the frame, such as BasketballDrive, BasketballDrill, and Johnny, the current CU characteristics may be much different from those of the neighboring CUs if the current CU is located at the boundary between the moving object and the background. Furthermore, for a strong motion in a high-resolution video, the edges of the moving objects were smoothed and

the *Var*, Var_{sub}, and *EPs* values were smaller, which strongly affects the CU depth decision. Hence, the BDBR values of these sequences were slightly higher than those of other sequences.

The computational overhead of SVM prediction is about 4%, which is acceptable and is still can be improved by programming or establishing lookup tables for offline trained models.

Reference [4] used a fixed threshold for CU variance for early termination and early splitting, thus it can only perform well in some sequences. Besides, making decision without reserving "unsure" case achieves considerable time saving but substantially degrades the RD performance. Compared with the fast CU depth decision algorithm for intra coding in [5], applying machine learning substantially reduces computational complexity; however, the bitrate performance of the proposed algorithm slightly increases compared with the performance of conventional methods of observation and analysis.

4. CONCLUSION

This paper proposed an ANN- and SVM-based fast CU depth decision algorithm for intra coding of HEVC. Machine learning provided a systematic approach to develop a fast algorithm for early CU splitting or termination to reduce intra coding computational complexity. Appropriate features, including neighboring CU depth, boundary pixel difference, pixel variance, and number of edge points, were extracted from the spatial and pixel domains. An ANN was used to investigate the effect of each feature. The features were classified into three types to generate three SVM training models at each depth. Furthermore, different weights were assigned to the three SVM outputs on the basis of the ANN analysis to calculate the final SVM result. In addition, an edge point checking process was performed at depth 0 to improve the RD performance.

Experimental results show that the proposed algorithm saves approximately 33% encoding time on average with a 1.55% BDBR loss compared with HM 15.0. Applying machine learning can substantially reduce computational complexity with a favorable bitrate increase.

REFERENCES

1. J. Lainema, F. Bossen, W. J. Han, and J. H. Min, "Intra coding of the HEVC standard," *IEEE Transactions on Circuits Systems for Video Technology*, Vol. 22, 2012, pp. 1792-1801.
2. C. Bai and C. Yuan, "Fast coding tree unit decision for HEVC intra coding," in *Proceedings of IEEE ICCE-China Workshop*, 2013, pp. 28-31.
3. J. W. Qiu, F. Liang, and Y. L. Luo, "A fast coding unit selection algorithm for HEVC," in *Proceedings of IEEE International Conference on Multimedia and Expo Workshops*, 2013, pp. 1-5.
4. T. Nishikori, T. Nakamura, T. Yoshitome, and K. Mishiba, "A fast CU decision using image variance in HEVC intra coding," in *Proceedings of IEEE Symposium on Industrial Electronics and Applications*, 2013, pp. 52-56.
5. L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm

for HEVC intra coding," *IEEE Transactions on Consumer Electronics*, Vol. 59, 2013, pp. 207-213.
6. T. Mallikarachchi, A. Fernando, and H. K. Arachchi, "Efficient coding unit size selection based on texture analysis for HEVC intra prediction," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2014, pp. 1-6.
7. B. Min and R. C. C. Cheung, "A fast CU size decision algorithm for the HEVC intra encoder," *IEEE Transactions on Circuits Systems for Video Technology*, Vol. 25, 2015, pp. 892-896.
8. L. Shen, Z. Zhang, and Z. Liu, "Effective CU size decision for HEVC intracoding," *IEEE Transactions on Image Processing*, Vol. 23, 2014, pp. 4232-4241.
9. H. Kalva, P. Kunzelmann, R. Jillani, and A. Pandya, "Low complexity H.264 intra MB coding," in *Proceedings of IEEE International Conference on Consumer Electronics*, 2008, pp. 1-2.
10. J. Kim, M. C. Kim, S. J. Hahm, I. J. Cho, and C. S. Park, "Block-mode classification using SVMs for early termination of block mode decision in H.264|MPEG-4 Part 10 AVC," in *Proceeding of International Conference on Advances in Pattern Recognition*, 2009, pp. 83-86.
11. J. Xiong and H. L. Li, "Fast and efficient prediction unit size selection for HEVC intra prediction," in *Proceedings of IEEE International Symposium on Intelligent Signal Processing and Communication Systems*, 2012, pp. 366-369.
12. X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP Journal on Image and Video Processing*, Vol. 2013, pp. 1-11.
13. C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, Vol. 20, 1995, pp. 273-297.
14. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, Vol. 313, 2006, pp. 504-507.
15. Matlab Central, Deep Learning Toolbox, http://www.mathworks.com/matlabcentral/fileexchange/38310-deep-learning-toolbox.
16. Y. C. Liu, Z. Y. Chen, J. T. Fang, and P. C. Chang, "SVM-based fast intra CU depth decision for HEVC," in *Proceedings of IEEE Data Compression Conference*, 2015, p. 458.
17. C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, Apr. 2011, http://www.csie.ntu.edu.tw/~cjlin/libsvm.
18. G. Bjøntegaard, "Calculation of average PSNR difference between RD-curves," ITU-T Q.6/SG16 VCEG 13th Meeting, Document VCEG-M33, 2001.

Zong-Yi Chen (陳宗毅) received the B.S. degree in Electrical Engineering and the M.S. degree in Communication Engineering from National Central University, Taiwan, in 2005 and 2007, respectively. He is currently pursuing the Ph.D. degree at the Video-Audio Processing Laboratory (VAPLab) in the Department of Communication Engineering at National Central University, Taiwan. His research interests include video/image processing and video compression.

Jiunn-Tsair Fang (方俊才) received the B.S. degree in Physics from National Taiwan University in 1987, and Ph.D. degree in Electrical Engineering from National Chung-Cheng University, Taiwan in 2004. Currently, he is an Assistant Professor in the Department of Electronic Engineering at Ming Chuan University, Taiwan. His researching interest includes video/image coding, channel coding, and joint source and channel coding.

Yen-Chun Liu (劉宴均) received the B.S. degree and the M.S. degree in Communication Engineering from National Central University, Taiwan, in 2012 and 2014, respectively. Her research interest is video coding and machine learning. Now her work centered on solid state disk (SSD) firmware development.

Pao-Chi Chang (張寶基) received the B.S. and M.S. degrees from National Chiao Tung University, Taiwan, in 1977 and 1979, respectively, and the Ph.D. degree from Stanford University, California, 1986, all in Electrical Engineering. From 1986 to 1993, he was a research staff member of the Department of Communications at IBM T. J. Watson Research Center, Hawthorne, New York. At Watson, his work centered on high speed switching systems, efficient network design algorithms, and multimedia conferencing. In 1993, he joined the faculty of National Central University, Taiwan, where he is presently a Professor in the Department of Communication Engineering. In 1994, Dr. Chang established and has headed the Video-Audio Processing Laboratory (VAPLab) in the Electrical Engineering Department and Communication Department of National Central University since. Dr. Chang is the principle investigator for many joint projects with National Science Council (NSC), Institute of Information Industry (III), Chung Hwa Telecommunication Laboratories (TL), and many other companies. His research interests include speech/audio coding, video/image compression, digital watermarking and data hiding, multimedia communication, deep learning, and multimedia retrieval.