Machine Learning-Based Coding Unit Depth Decisions for Flexible Complexity Allocation in High Efficiency Video Coding

Yun Zhang, *Member, IEEE*, Sam Kwong, *Fellow, IEEE*, Xu Wang, *Member, IEEE*, Hui Yuan, *Member, IEEE*, Zhaoqing Pan, and Long Xu

***Abstract*— In this paper, we propose a machine learning-based fast coding unit (CU) depth decision method for High Efficiency Video Coding (HEVC), which optimizes the complexity allocation at CU level with given rate-distortion (RD) cost constraints. First, we analyze quad-tree CU depth decision process in HEVC and model it as a three-level of hierarchical binary decision problem. Second, a flexible CU depth decision structure is presented, which allows the performances of each CU depth decision be smoothly transferred between the coding complexity and RD performance. Then, a three-output joint classifier consists of multiple binary classifiers with different parameters is designed to control the risk of false prediction. Finally, a sophisticated RD-complexity model is derived to determine the optimal parameters for the joint classifier, which is capable of minimizing the complexity in each CU depth at given RD degradation constraints. Comparative experiments over various sequences show that the proposed CU depth decision algorithm can reduce the computational complexity from 28.82% to 70.93%, and 51.45% on average when compared with the original HEVC test model. The Bjøntegaard delta peak signal-to-noise ratio and Bjøntegaard delta bit rate are −0.061 dB and 1.98% on average, which is negligible. The overall performance of the proposed algorithm outperforms those of the state-of-the-art schemes.**

***Index Terms*— High efficiency video coding, coding unit, machine learning, support vector machine.**

I. Introduction

HIGH Definition (HD) and Ultra-High Definition (UHD) videos are becoming more and more popular since they can provide higher perceptual quality and create a more realistic viewing experience, which has a promising video market, such as TV broadcasting, IMAX movie, immersive video communication, network video streaming and HD video surveillance, etc. However, the data volume of these HD and UHD videos increase dramatically as the resolution and frame rate increase. For example, an 8K × 4K@120fps video has 11.5 Giga Bytes per second raw data, which requires highly efficient compression. To address this problem, the Joint Collaborative Team on Video Coding (JCT-VC) has developed the High Efficiency Video Coding (HEVC) standard [1] with 50% bit rate reduction beyond the H.264/AVC high profile. The HEVC adopts advanced coding techniques, including highly flexible quad-tree coding block partition, INTRA prediction with 35 modes, discrete sine transform, sophisticated interpolation and filtering etc. These techniques significantly improve the compression efficiency. However, they cost intensive computational complexity and increase hardware costs, including computing power, memory access, storage space and energy consumption, which hinder HD/UHD videos in real-time applications, such as live TV broadcasting, mobile video communication and surveillance.

The core of the coding layer in HEVC is the Coding Tree Unit (CTU), conceptually similar to the macroblock in H.264/AVC, consisting of a Luma Coding Tree Block (CTB), the corresponding Chroma CTBs, and syntax elements. A CTB may contain only one Coding Unit (CU) or be split to multiple CUs, with the size of 8 × 8, 16 × 16, 32 × 32 or 64 × 64 [1]. Each CU and associated Coding Blocks (CB) can be further divided into smaller Prediction Units (PUs) with various modes, including SKIP or MERGE mode, 8 INTER modes and 2 INTRA modes. Finally, the residues of the PU will be processed by a tree of Transform Units (TUs). The optimal CU, PU, and TU are then determined based on recursive Rate-Distortion (RD) cost comparisons,

Manuscript received August 26, 2014; revised January 9, 2015; accepted March 12, 2015. Date of publication March 27, 2015; date of current version April 14, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61471348, Grant 61102088, Grant 61272289, and Grant 61202242, in part by the Shenzhen Overseas High-Caliber Personnel Innovation and Entrepreneurship Project under Grant KQCX20140520154115027, in part by the Shenzhen Emerging Industries of the Strategic Basic Research Project under Grant JCYJ20120617151719115, and in part by the 100-Talents Program of Chinese Academy of Sciences under Grant Y434061V01. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chang-Su Kim.

Y. Zhang is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: yun.zhang@siat.ac.cn).

S. Kwong is with the Department of Computer Science, The City University of Hong Kong, Hong Kong, and also with the Shenzhen Research Institute, The City University of Hong Kong, Hong Kong (e-mail: cssamk@cityu.edu.hk).

X. Wang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: wangxu.cise@gmail.com).

H. Yuan is with the School of Information Science and Engineering, Shandong University, Jinan 250100, China (e-mail: yuanhui0325@gmail.com).

Z. Pan is with the Jiangsu Engineering Center of Network Monitoring, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: zhaoqingpan@nuist.edu.cn).

L. Xu is with the Key Laboratory of Solar Activity, National Astronomical Observatories, Chinese Academy of Sciences, Beijing 130117, China (e-mail: lxu@nao.cas.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2417498

where the "try all and find the best" strategy is of extremely high complexity.

Many researchers have devoted their efforts on optimizing the coding complexity of HEVC encoder [2]–[10]. Li *et al.* [2] proposed an algorithm that adaptively determine CU depth range based on the depth information of previously coded slices and CUs. The CU depth range is adapted at slice level to avoid potential encoding complexity. In [3], Xiong *et al.* proposed CU selection algorithm based on Pyramid Motion Divergence (PMD) formed by optical flow of down-sampled frames. Then, whether split the CU or not was determined by K-nearest neighboring like algorithm. Basically, it exploits motion properties and spatial neighboring correlations of the video content while doing the CU depth decision optimization. However, optical flow estimation and PMD comparison among hundreds of samples in First-In-First-Out (FIFO) stack causes large complexity overhead. In [4], Shen *et al*. exploited the RD cost and mode correlations among different depth levels and/or spatially neighboring CUs so as to skip some rarely used CU and PU modes for INTRA coding. It is a joint optimization of the CU depth decision and PU mode decision. In terms of the INTER frame optimization, Shen *et al.* [5] introduced early termination based on motion homogeneity and SKIP mode checking to skip the complex Motion Estimation (ME) process. Similarly, Goswami *et al.* [6] proposed a CU quad-tree early termination algorithm based on the RD cost difference between the root and children CUs. In [7], the CU depth is determined based on RD cost comparisons among 2N $\times$ 2N PU modes in different CU depth levels. Also, MERGE or SKIP mode is early decided based on the contextual mode information of neighboring CUs. These algorithms focus on the CU depth or size decision. Besides, optimizations on PU mode prediction were investigated in [8]–[10]. In [8], Lee *et al.* proposed a PU decision method which computed the priority of all INTER prediction modes and performed the ME only on the selected PU mode. Zhao *et al.* [9] computed a 2D distance for each PU mode and applied the optimal stopping theory to early terminate the PU searching process. Pan *et al.* [10] proposed an early MERGE mode decision method for both root and children CUs based on hierarchical depth correlation and ME information. Basically, these algorithms are based on the statistics on the RD cost properties, temporal and spatial correlation, and then the PUs are early decided based on some hard or soft thresholds, which limit their applicability and may be difficult to handle the situations with various contents, parameters and/or complex coding structures.

Machine learning is a hotspot and widely applied in artificial intelligence, pattern recognition and signal processing, since it learns from big data of complex situations and gives the optimal solution. With this excellent property, researchers attempted to apply learning based algorithms in video coding for better performances. In [11], Martinez-Enriquez *et al*. proposed a two-level classification based approach for the INTER mode decision in H.264/AVC, where a normalized RD cost by Sum of Absolute Difference (SAD) was adopted as a new feature for Support Vector Machine (SVM) classifier. In addition, RD cost increment caused by misclassification was considered in the SVM training. Zhang *et al.* [12], [13] proposed a statistical model for SKIP/DIRECT mode decision in H.264/AVC based multiview video coding, where the early termination decision was made based on the Laplace RD cost distribution and inter-view correlation. Sung and Wang [14] adopted clustering to predict the INTER modes for H.264/AVC based on a 3D feature vector consists of spatial, temporal and SKIP/DIRECT mode RD cost. Additionally, multi-phase classifications were presented. However, this approach highly depends on the global statistical information of previous coded blocks. In [15], Chiang *et al*. proposed two levels of classifications by using the Back Propagation Neural Network (BPNN), where the first level was for the mode decision and the second level was for the reference frame selection. A subset of the mode candidates was finally selected. The RD cost increase and complexity reduction can be adjusted by controlling the number of the mode candidates. In addition, decision tree was applied to tackle the mode prediction in H.264/AVC [16] and separate out SKIP mode in stereo video coding [17]. In [18], Kim and Kuo handled the feature space by considering the risks of wrong mode decision, and meanwhile, this scheme could have a trade-off between RD performance and complexity by using different mode decision parameters. However, these schemes were proposed for H.264/AVC, which may not be efficient as they are directly applied to HEVC, since H.264 and HEVC have different coding block structures and modes.

In machine learning based HEVC optimization, Shen and Yu [19] proposed a CU splitting early termination algorithm based on weighted SVM for the latest HEVC standard, in which the RD loss caused by misclassifications was modeled as weights in offline SVM training. Meanwhile, feature selection optimization was also presented to reduce the size of feature space. Shanableh *et al.* [20] and Peixoto *et al.* [21] proposed a fast CU size decision algorithm for transcoding bit stream from MPEG-2 and H.264/AVC to HEVC, in which the information of the MPEG-2 and H.264 bit stream was extracted and adopted as input features of the SVM classifier. Additionally, statistical thresholds were added to tackle the inaccurate prediction. The performances of these algorithm highly depend on the feature selection and the prediction accuracy. Once the prediction is not so accurate for an un-expected data set, they may cause large RD degradation. The existing works generally used a single or ensemble classifier to do the classification, in which the coding performances, such as the RD cost and complexity, cannot be modified or transformed. These encoders thereby can hardly adapt to different system requirements.

In this paper, we propose a machine learning based fast CU depth decision for HEVC. The major contributions are: 1) The HEVC CU depth decision is modeled as a hierarchical binary classification problem. 2) A flexible CU depth decision structure is presented for each decision level, which can take advantages of three existing CU depth decision structures. 3) A joint classifier consisting of multiple SVM classifiers is proposed by considering the risk of false prediction. 4) A novel optimal parameter determination algorithm of training the

TABLE I
CU DEPTH DISTRIBUTION AND POTENTIAL TIME SAVING RATIO [%]

Sequences	QP	CU Depth Distribution				Theoretical Time Saving Ratio		
		D0	D1	D2	D3	DL1	DL1,2	DL1,2,3
Basketballpass (416×240)	24	31.2	34.8	30.1	3.9	39.9	66.4	74.8
	28	30.2	36.9	30.1	2.8	39.4	66.7	74.7
	32	31.2	38.8	28.2	1.8	39.9	67.4	74.7
	36	33.2	41.4	24.4	1.0	41.0	68.6	74.8
BQMall (832×480)	24	29.7	38.6	27.8	3.9	39.1	66.9	74.7
	28	38.9	36.9	21.5	2.7	44.0	69.1	75.0
	32	45.9	34.7	17.7	1.7	47.8	70.4	75.2
	36	52.9	32.6	13.5	1.0	51.5	71.9	75.4
FourPeople (1280×720)	24	64.6	22.0	12.8	0.6	57.7	72.4	75.7
	28	71.9	17.8	9.9	0.4	61.6	73.4	75.9
	32	76.4	15.3	8.1	1.2	64.0	74.0	75.8
	36	79.8	12.9	7.1	0.2	65.9	74.3	76.1
Tennis (1920×1080)	24	18.3	50.7	28.9	2.1	33.0	66.9	74.4
	28	27.4	49.6	21.7	1.3	37.9	69.1	74.7
	32	38.2	45.3	15.6	0.9	43.7	71.0	75.0
	36	48.2	40.3	10.9	0.6	49.0	72.5	75.3
Traffic (2560×1600)	24	37.5	34.3	25.8	2.4	43.3	68.0	74.9
	28	50.7	30.7	17.3	1.3	50.3	70.8	75.3
	32	60.4	27.6	11.4	0.6	55.5	72.7	75.6
	36	69.3	23.2	7.2	0.3	60.3	74.0	75.8
Average		**46.80**	**33.22**	**18.50**	**1.53**	**48.24**	**70.32**	**75.19**

joint classifier is proposed for CU level complexity allocation, which has a good trade-off between RD cost and complexity. This paper is organized as follow, Section II presents the motivation and analyses of CU depth decision. Section III presents the proposed machine learning based CU depth decision algorithm, including the coding structure, joint classifier, feature selection and optimal parameter determination. Then, experimental results and analyses are presented in Section IV. Finally, conclusions are drawn in Section V.

II. MOTIVATION AND STATISTICAL ANALYSES

In the latest HEVC standard, the luma CTB supports quad-tree CU partitions with four level of CU depth from 0 to 3, which corresponds to CU size from 64 × 64 to 8 × 8. The HEVC encoder checks 4^i, $i \in [0, 3]$, CU partitions for depth level i, which is $1 + 4 + 16 + 64 = 85$ CU partitions in total. However, only parts of them will be finally selected as the optimal CU partitions, which are from 1 to 64. Therefeore, if we can precisely predict the best CU mode, 84 to 21 CU modes are not necessary to be checked. TABLE I shows the CU depth distribution and theoretical time saving ratio. D0 to D3 indicates the four depth levels. Five diverse sequences with various resolutions and properties were encoded for statistical analyses. We find that there are 46.80%, 33.22%, 18.50% and 1.53% CUs selected the depth 0, 1, 2 and 3 as its best CU depth, respectively. The average probability increases as the depth level decreases and as the Quantization Parameter (QP) increases. On the other hand, the amount of CUs with depth 3 is very small. The right three columns are theoretical maximun time saving ratio for different Decision Levels (DLs) when the CU partition is 100% acurately predicted and selected. DL1 to DL3 are

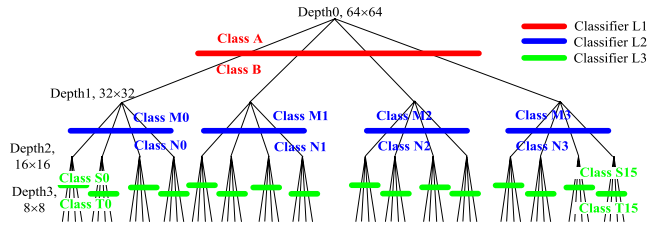


Fig. 1. Classifier distribution for a CTB.

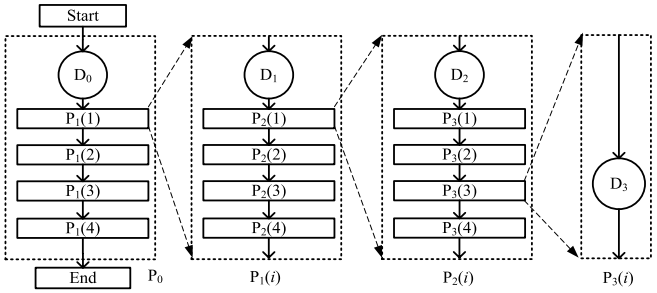


Fig. 2. Flowchart of compressing a CTB.

three DLs of non-spliting or futher spliting prediction. We can find there would be 75.19% complexity reduction on average if all CU depth levels can be accurately predicted.

III. THE PROPOSED MACHINE LEARNING BASED CU DEPTH DECISION ALGORITHM

A. Classification Problem Formulation for CU Depth Decision in HEVC

For each CTB, there are $(2^4+1)^4 + 1 = 83522$ different combinations in CU size for a CTB partition when the allowed CU depth is from 0 to 3. There are too many types of CU partitions (classes) and it is hard to be solved by a single multi-class classification. Fortunately, we handle this complicated multi-class problem with three-level of hierarchical binary classifications. Fig. 1 shows the classifiers assigned for the depth decision in a CTB. The quad-tree is corresponding to the CU partition for a CTB and each horizontal line is a classifier. Above and below the line are two classes, i.e. split or non-split. There are 3 kinds of classifiers from level 1 to 3, also called DLs, which are labeled with red, blue and green color, respectively. Meanwhile, there could be $1+4+16=21$ classifiers or 3 classifiers that repeated 1, 4 and 16 times respectively to solve the whole classification problem.

B. The Proposed Structure for Fast CU Depth Decision

The CU depth decision in HEVC is a recursive process, as shown in Fig. 2, where D_n represents a process of checking the CU at depth level n. For example, D_0 checks CU depth at level 0 where the CU size is 64 × 64. $P_n(i)$ is the basic component for the recursive CU depth decision in HEVC, where $n \in \{0,1,2,3\}$ is the depth level, $i \in \{1,2,3,4\}$ is the index of sub-CUs,. Each $P_n(i)$ $n \in \{0,1,2\}$ includes one D_n and four $P_{n+1}(i)$, while the $P_3(i)$ is D_3. The CTB is firstly checked with CU depth 0, i.e. D_0. Then it is checked with four $P_1(i)$. Each $P_1(i)$ needs recursively checking D_1 and

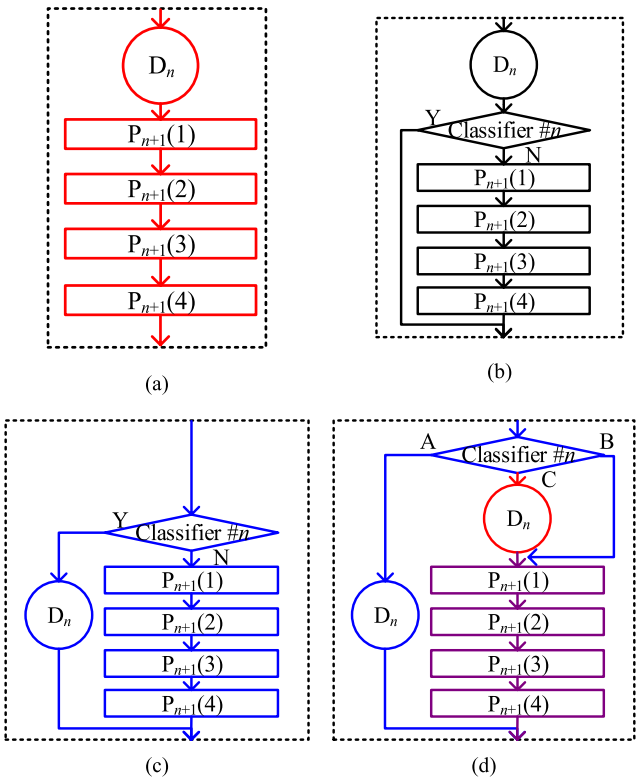


Fig. 3. $P_n(i)$ component, $n = 0,1,2$, $i = 1,2,3,4$. (a) $P_n(i)$ in HM. (b) $P_n(i)$ in early termination scheme (Shen's scheme) [19]. (c) $P_n(i)$ in a fast scheme. (d) Proposed $P_n(i)$ component.

four $P_2(i)$. The $P_n(i)$ is recursively checked until reaches D_3. After all CU depths/sizes have been checked, the optimal CU size or their combinations will be selected as the one has the minimum RD cost. Since all CU sizes from 64 × 64 to 8 × 8 shall be checked, it is very time consuming as analyzed in Section II. Since $P_n(i)$ is the basic component for the recursive CU decision process in Fig. 2, we can analyze the $P_n(i)$ instead of the whole CU decision process. Fig. 3(a) shows the $P_n(i)$ component in the original HEVC test model (HM), which checked one D_n and four $P_{n+1}(i)$. This "try all and select the best" strategy has the best RD performance but with the highest complexity.

To reduce the complexity of the CU depth decision process, many fast algorithms or early termination algorithms [3]–[5], [19] have been investigated. In [19], Shen *et al.* presented an early termination in which D_n is firstly checked and then a classification is applied to predict whether the CU coding process shall be early terminated or not, as shown in Fig. 3(b). In this structure, on matter which CU depth is the optimal, D_n is always checked. This D_n checking is unnecessary when $P_{n+1}(i)$ is the best, which causes computational complexity overhead. Additionally, false acceptance of "N" leads to unnecessary $P_{n+1}(i)$ checking. False acceptance of "Y" will reduce the complexity and meanwhile miss the optimal CU size, which cause RD degradation. This structure is efficient in the case that D_n is with low complexity and poses large probability of being the optimal. However, according to the statistical results in TABLE I, we found the probabilities for each CU depth distribute evenly, which may make the

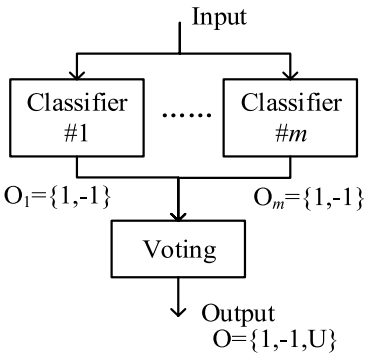


Fig. 4. Flowchart of the joint classifier, where 1, −1, and, U corresponds to A, B and C in Fig. 3(d).

Structure (b) not so efficient. For example, for the case that Depth 3 is the best CU depth level, even though the Depth 3 can be precisely predicted, this framework has to check Depth 0 to Depth 2, which are unnecessary.

To tackle these unnecessary checking and minimize the coding complexity, we propose a new structure where the classification is performed to predict the CU depth before checking D_n, as shown in Fig. 3(c). If D_n is predicted to be the optimal, only D_n will be checked. Otherwise, only $P_{n+1}(i)$ will be checked. In this structure, the advantage is no additional complexity overhead and the complexity could be minimized if the classification is accurate. However, the disadvantage is the byproduct information of the coded D_n is not available for the CU depth prediction. Additionally, the RD performance highly depends on the prediction accuracy of the classification, since misclassifications (either false acceptance or false rejection) will change the optimal CU partition and consequently cause the RD degradation. Unfortunately, due to the variety of the video contents, CU modes, and limited available information, the prediction accuracy of the classifiers can hardly be guaranteed in sufficient high level, e.g. 95%, which may fail to maintain the compression efficiency.

To control the potential RD degradation and make the structure more flexible to different prediction accuracies, we proposed a more advanced structure which integrates the Structure (a) and Structure (c) together, as shown in Fig. 3(d). In the figure, the components with red color and purple color are the $P_n(i)$ in HM, i.e. Structure (a). The components with blue and purple color are the $P_n(i)$ in (c). The classifier in Fig. 3(d) has three outputs, i.e. A, B and C. If the choice of C is 100%, i.e. the output is fixed to C, this structure is transformed to be Structure (a), which has the best RD performance and highest complexity since it tries all and selects the best. If the prediction of C is 0%, i.e. no output to C, this Structure (d) is transformed to be Structure (c), which has the lowest complexity without any unnecessary checking. Actually, the proposed flexible structure can also be transformed to the Structure (b) in Fig. 4(b) when the probability of B is zero.

In summary, the proposed structure is an integration of Structure (a), (b) and (c), and it can thereby be transformed to either one of them or in between them as changes the probabilities of A, B and C. This property makes the structure

more flexible and is capable of having a better trade-off among the RD performance and complexity according to user's preferences or system objectives. That is the RD degradation can varies from 0 to ω_{RD} and complexity reduction varies from 0 to ω_{C}, where ω_{RD} and ω_{C} denote the maximum RD degradation and complexity reduction achieved by Structure (c).

Based on the above discussions, the challenging issues are to design the three-output classifier and to determine the probabilities of A, B and C, which are investigated in the next sub-section.

C. Design of the Three-Output Classifier at CU DL i

There are many multi-class classifiers available for the structure in Fig. 3(d). However, in the CU size depth decision, the misclassifications of split or non-split mode have different impacts on the coding performance. For example, in Fig. 3(d), the false acceptance of non-split mode (A) degrades the RD while the false acceptance of split mode (B) causes complexity overhead as well as RD degradation. Besides, choosing C in Fig. 3(d) is actually choosing both A and B. To take advantage of these properties and minimize the complexity in CU depth decision, we design a joint three-output classifier which is flexible to tackle the classification problem in Fig. 3(d).

Fig. 4 shows a flowchart of the joint classifier, which consists of multiple binary classifiers from 1 to m and a voting module. Given an input, each classifier will give a prediction from O_1 to O_m with the binary results, 1 or -1. After that, the voting module gives the finally output as 1, -1 or U, where U indicates uncertain or high risk of false prediction. The joint classifier has the following five properties:

1) Classifier #1 to #m can be either one learning machine with different parameters or different learning machines;
2) Classifier #1 to #m can be a single classifier or classifier group that optimized with boosting or ensemble algorithms [22];
3) It includes more than two classifiers, i.e. $m \geq 2$;
4) It can be employed for binary classification or multi-class problem which depends on the properties of the classifiers and the voting algorithm;
5) In the CU depth decision, if the prediction is the U, conservative CU decision method, i.e. checks both split and non-split mode, is adopted to determine the CU depth.

Since SVM is robust and popular in solving the binary classification problem, we employed the SVM with different parameters and m is 2 for simplicity in this paper.

For a set of N_A and N_B training instances in Class A and B, the i^{th} training sample $\{x_i, y_i\}_{i=1}^{N_A+N_B}$, $x_i \in R^n$, $y_i \in \{-1, 1\}$, x_i is a n-dimensional input feature vector and y_i is the output class label. Based on the SVM, there is a trained hyperplane $f(x) = \mathbf{w}^T \phi(x) + b$ to discriminate the samples, where $\phi()$ is a nonlinear operator. The hyperplane is constructed by minimizing a cost function. In the paper, we adopt the weighted SVM classifier, where the objective function is increased by penalizes non-zero ξ_i and ξ_j and the optimization becomes a trade-off between a large margin and a small error penalty. The optimization cost function is thus become minimizing

$$J(\mathbf{w}, w_0, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C_P\left(W_A \sum_{i=1}^{N_A} \xi_i + W_B \sum_{j=N_A+1}^{N_A+N_B} \xi_i\right), \tag{1}$$

subject to

$$\begin{aligned} y_i\left[\mathbf{w}^T \mathbf{x}_i + w_0\right] &\geq 1 - \xi_i, \quad i = 1, 2, \ldots, N_A + N_B \\ \xi_i &\geq 0, \qquad i = 1, 2, \ldots N_A + N_B, \end{aligned} \tag{2}$$

where ξ is a vector of parameters ξ_i and ξ_j. Parameter C_P is a positive constant that controls the relative influence of the two competing terms. W_A and W_B are weighted factor for Class A and Class B, respectively. The Lagrange multipliers can be solved by

$$\begin{aligned} \alpha^* = \arg\max_{\alpha} & \sum_{i=1}^{N_A+N_B} \alpha_i \\ & - \frac{1}{2} \sum_{i=1}^{N_A+N_B} \sum_{j=1}^{N_A+N_B} \alpha_i \alpha_j y_i y_j K\left(x_i, x_j\right), \end{aligned} \tag{3}$$

subject to

$$\begin{aligned} &\sum_{i=1}^{N_S} \alpha_i y_i = 0, \quad C_P W_A \geq \alpha_i \geq 0, \quad i = 1, 2, \ldots N_A \\ &\sum_{i=1}^{N_B} \alpha_{i+N_A} y_{i+N_A} = 0, \quad C_P W_B \geq \alpha_{N_A+i} \geq 0, \quad i = 1, 2, \ldots N_B, \end{aligned} \tag{4}$$

where α_i is the Lagrange multipliers, $K(x_i, x_j)$ is the kernel operator. Equation (3) can be solved based on the Karush-Kuhn-Tucker conditions. The point x_i with $\alpha_i > 0$ is called support vector. In this paper, Gaussian Radial Basis Function (RBF) is employed as the kernel function since it handles the non-linear case well and has fewer numerical difficulties.

In the voting module, m output of classifiers O_i, $i \in \{1, 2, \ldots m\}$, are combined to generate the output of joint classier O_{ALL} as

$$O_{ALL} = \begin{cases} 1 & \sum_{i=1}^{m} O_i \geq T_A \\ U & others \\ -1 & \sum_{i=1}^{m} O_i < T_B, \end{cases} \tag{5}$$

where U is the output of uncertain prediction, which indicates either positive $(+1)$ or negative (-1) prediction has a high risk of false prediction. T_{A} and T_{B} are thresholds for $+1$ and -1 prediction, respectively. A special case is when T_{A} equals to T_{B} there is no uncertain output. When T_{A} is larger than the maximum value m and T_{B} is smaller than the minimum value $-m$, the output is all uncertain.

In a general classification, the misclassification of Class A and B are treated equally and the hyperplane is determined by maximizing the average prediction accuracy

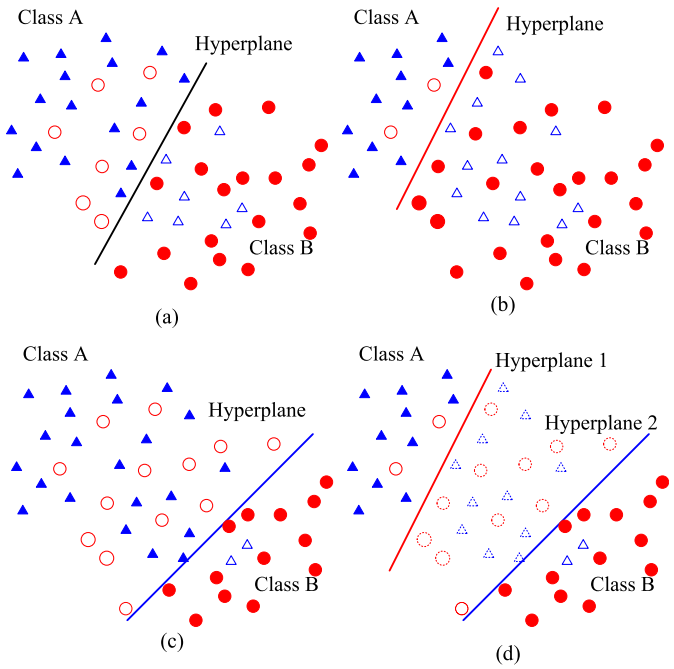


Fig. 5. Examples of Classification. (a) Traditional SVM. (b) Weighted SVM Classifier #1 ($W_B > W_A$). (c) Weighted SVM Classifier #2 ($W_B < W_A$). (d) Joint Classifier.

and margin. An example is shown in Fig. 5(a). However, in the CU depth decision, the misclassification of Class A (split) and B (non-split) leads to different costs. Therefore, the coefficients W_A and W_B are employed to indicate different penalty weights for the misclassification costs of Class A and B, respectively. To reduce the False Acceptance Rate (FAR) of Class A/B, we can enlarge W_B or W_A to give higher penalty to the false acceptance of Class A/B. As shown in Fig. 5(b), the hyperplane moves upward as W_B increases. Thus, the FAR of Class A increases and meanwhile the FAR of Class B reduces. On the contrary, the hyperplane moves downward as W_A increases, and thereby the FAR of Class A can be reduced, as shown in Fig. 5(c). As combined the two classifiers together, the up-left instances of hyperplane #1 are classified as Class A and the bottom-right instances of the hyperplane #2 are classified as Class B. The instances between the two hyperplanes are determined as uncertain instances.

In the CU depth decision, the joint classifier can be applied to predict the split and non-split mode with sufficient high prediction accuracy by given a high value W_A and W_B. Meanwhile, for the instances with uncertain prediction, they are determined by full RDO comparison, which has 100% CU depth decision accuracy since it "tries all and selects the best" and is identical to the original HM model. Thus, the overall prediction accuracy of the CU depth decision can be guaranteed to a required high level. In the example in Fig. 5(d), only 4 samples (2 red circles and 2 blue triangles) are misclassified.

D. Features Selection for CU Depth Decision

The aim of this paper is to reduce the computational complexity of the CU depth decision while maintaining the RD performance. Thus, the feature extraction process must be low complexity to avoid the computation overheads. On the other hand, the CU size mainly depends on the texture of the video content, motion, context of temporal and spatial neighboring information, etc. Based on these two principles, we consider the nine features for INTER CU depth decision as follow:

1) Information of the SKIP or Merge Mode in current CU. Since the SKIP/MERGE mode of the current CU size is with low complexity compared to other INTER modes, the SKIP mode can be checked first and its output information is helpful to the following CU depth prediction. Thereby, the Coded Block Flag (CBF), RD cost, total distortion, and total coding bits of the SKIP/Merge mode are used as the features since they indicate the INTER prediction error, video texture and coding efficiency by using SKIP/Merge mode. They are denoted as $x_{CBF\_Meg}(i)$, $x_{RD\_Meg}(i)$, $x_{D\_Meg}(i)$, $x_{Bit\_Meg}(i)$, respectively, where i indicates the CU depth. In addition, SKIP flag after the merge mode is also used as a feature, denoted $x_{SKIP}(i)$.
2) Motion information. Since the CU partition is highly correlated with the motion in INTER frames, the motion vectors of the MERGE mode $x_{MV\_Meg}(i)$ can be used as the feature. In this paper, it is defined as $x_{MV\_Meg}(i) = |MVx|+|MVy|$, where MVx and MVy are the horizontal and vertical motion.
3) Context information. Since the video content has spatial correlation, the neighboring CU information, including neighboring CU RD cost $x_{NB\_RD}(i)$ and CU depth $x_{CU\_depth}(i)$, can be employed as features. In this paper, $x_{NB\_RD}(i)$ is the average RD cost of the left and above CUs. $x_{CU\_depth}(i)$ is an average value of the CU depth, which is $x_{CU\_depth}(i) = \frac{1}{N_{LFT}(i)+N_{ABV}(i)} \sum_{j=1}^{N_{LFT}(i)+N_{ABV}(i)} d_j$, where d_j is the CU depth labels of the j^{th} 4×4 unit. For example, for a CU size with 16×16, there are 16 4×4 units labeled with 2. $N_{LFT}(i)$ and $N_{ABV}(i)$ are the number of 4×4 unit in left and above CUs with depth i.
4) Quantization parameter x_{QP} is also employed as a feature since it is of higher probability to select large CU size when QP becomes larger.

E. Training and Testing Mode

1) ONline Training Mode (ON-TM): In the ON-TM, some frames of a sequence are encoded with the original encoder and it outputs class labels and feature for model training. Then, the successive frames are encoded and their CU depths are predicted based on the trained model. The training frames and models can be refreshed as demand. An example of ON-TM is shown in Fig. 6(a), where the red dash lines are training frames and the black solid lines are predicting frames. The advantage of ON-TM is the properties of the video (sequence) for the training and testing are quite close. It is good for improving the prediction accuracy. However, 1) the online training set is usually small; 2) the training frames are encoded without low complexity optimization; 3) the training process is time consuming,

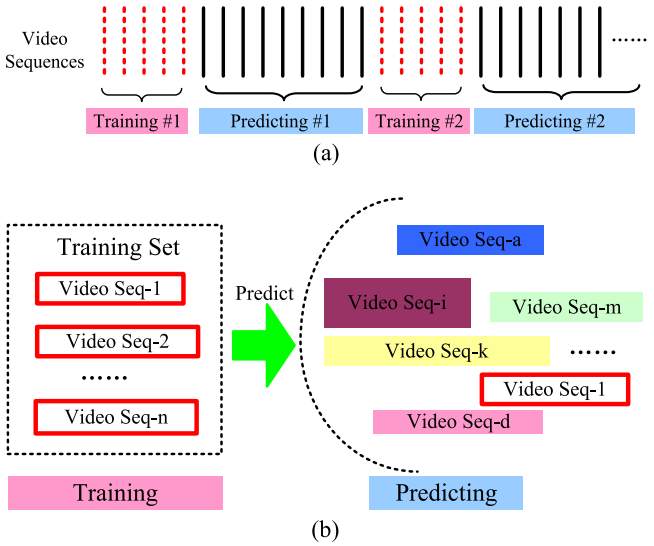


Fig. 6. Training Mode. (a) Online. (b) Offline.

which has complexity overhead, especially for the case of using large training set and complicated learning algorithm; 4) the encoder shall integrate the learning algorithm as well as the training process which is not easy for hardware implementation; 5) complicated learning algorithm and multi-pass training will increase the complexity overhead. This ON-TM mode is suitable for the case that features and learning model are highly sensitive to the content.

2) OFFline Training Mode (OFF-TM): For the OFF-TM, multiple video sequences with different properties are encoded with the original encoder to generate the training data. Then, we use an offline learning algorithm to train a model. Finally, the learned model is loaded into the learning based encoder to encode the video sequences, as shown in Fig. 6(b). The OFF-TM can well solve the above mentioned five shortcomings in ON-TM. In addition, optimal model parameters can be obtained via sophisticated learning or multiple-pass training. Since the correlation and statistical properties of the CU is not highly content dependent and there are many advantages of this OFF-TM mode. We thus adopt the OFF-TM mode for model training in this paper. Besides, we could include diverse video content, motions, properties, textures, resolutions, and coding parameters (e.g. different QPs) in this mode to train a robust and better learning model.

IV. Optimal Parameter Determination for the Complexity Allocation

In this section, we analyze the prediction accuracy of the proposed SVM predictor in CU depth decision. Then, we present a weighted factor determination algorithm for the model training so as to have a good trade-off between the RD cost increase and complexity reduction in CU depth prediction.

A. Prediction Accuracy Analysis for the SVM Predictor

Let N_A and N_B be the number of instances labeled as Class A and Class B in ground truth, respectively. $N_{AA,k}$ is the number of Class A instances that have been correctly classified as Class A by the SVM classifier #k. $N_{AB,k}$ is the number of Class A instances that have been falsely classified as Class B by the classifier #k. Similarly, $N_{BB,k}$ and $N_{BA,k}$ are the number of Class B instances that have been correctly or falsely classified. $N_{\mathrm{A}} = N_{AA,k}+N_{AB,k}$, $N_B = N_{BB,k}+N_{BA,k}$. The prediction accuracy of classifier #k for Class A, B and overall samples can be calculated as

$$\begin{cases} P_{A,k} = \dfrac{N_{AA,k}}{N_{AA,k}+N_{BA,k}} \\ P_{B,k} = \dfrac{N_{BB,k}}{N_{BB,k}+N_{AB,k}} \\ P_k = \dfrac{N_{AA,k}+N_{BB,k}}{N_A+N_B}, \end{cases} \tag{6}$$

To analyze the prediction accuracy of the SVM classifier, we statistically analyzed the $P_{A,k}$, $P_{B,k}$ and P_k with the samples collected from the video coding process. The training and testing samples were collected by the HEVC coding process with the original HM encoder. Two sequences, *BQMall* and *FourPeople*, 20 frames for each sequence, were encoded with four QPs, which are 24, 28, 32, and 36. There are 20090, 22725 and 36345 total samples respectively from DL 1 to 3. Sixty percent of them were used in model training and the rest of them were used for testing. The ratio of positive samples to negative samples of each DL are 1:2.45, 1:1.48 and 1:2.95, respectively, which means the number of positive (split mode) and negative (non-split mode) samples are generally in balance. C-SVM [23] with different weighted factors (W_{A} and W_{B}) are employed. For each DL, we trained and tested the model multiple times with different weighted factors (W_A, W_B), and finally collected the $P_{A,k}$, $P_{B,k}$, and P_k. The weighted factors W_{A} and W_{B} are set as $\{(W_A, W_B)|1 \le W_A \le 10, W_A \in \mathbf{N}, W_B = 1\} \cup \{(W_A, W_B)|1 \le \mathrm{W_B} \le 10, W_B \in \mathbf{N}, W_A = 1\}$. Thus, there are 20 pairs of (W_A, W_B) in total in each DL.

Fig. 7 shows the prediction accuracy ($P_{A,k}$, $P_{B,k}$ and P_k) for the SVM classifier with different weighted factors. We can find that from DL1 to DL3 the $P_{A,k}$ and $P_{B,k}$ are monotonic and can reach up to 100% as the ratio W_A/W_B properly changes. As shown in Fig. 5(c), as W_A/W_B increases, more samples of Class B are falsely classified as Class A and less Class A samples are falsely classified as Class B. Thus, $N_{BA,k}$ increases and $P_{A,k}$ decreases. The prediction accuracy for Class B ($P_{B,k}$) can be deduced in a similar way. The P_{k} value achieves its peak when W_B/W_A is 1, which equally treats the positive and negative samples. The peak P_k from DL1 to DL3 are 90.21%, 82.62%, 79.42%, respectively, which are the maximum prediction accuracy of using a single SVM predictor in the CU depth prediction. These values indicate that using a single SVM with given features and training way is not accurate enough and may cause large RD degradation in CU depth prediction.

We thus combined these two SVM classifiers with different $W_{\mathrm{B}}/W_{\mathrm{A}}$ together to formulate the joint classifier shown in Fig. 4. The uncertain CUs are encoded by the original RDO process and its prediction can be regarded as 100% correct. The overall prediction accuracy of the

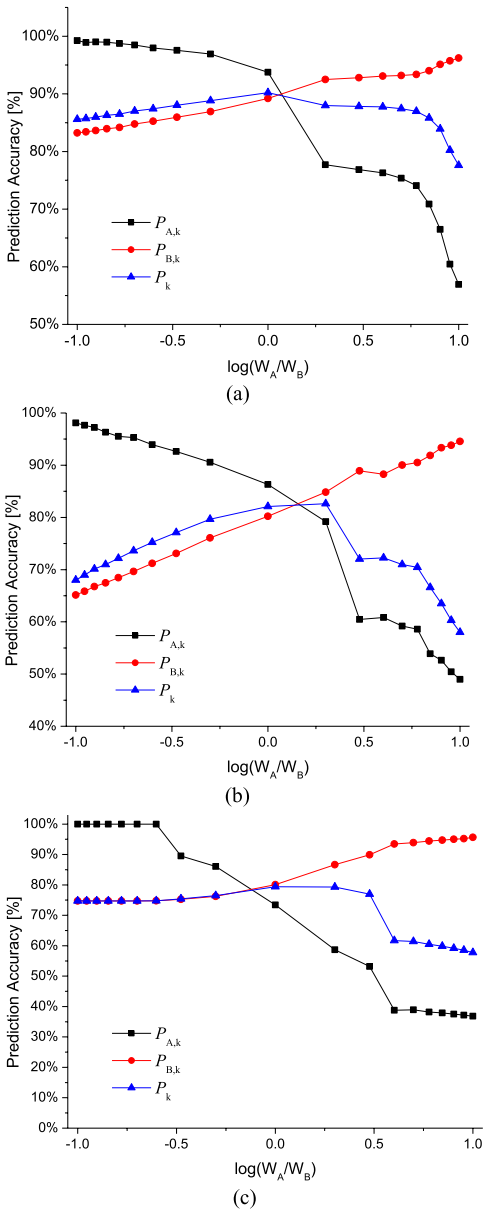


Fig. 7. Prediction accuracy for SVM classifier with different weighted factors. (a) DL 1. (b) DL 2. (c) DL 3.

joint classifier can be presented as

$$Q_{ALL} = \frac{N_{AA,1} + N_{BB,2} + N_{RD}}{N_A + N_B} = 1 - \frac{N_{BA,1} + N_{AB,2}}{N_A + N_B}. \tag{7}$$

where $N_{RD} = N_{AB,1} + N_{BB,1} - N_{BB,2} - N_{AB,2}$ is the number of uncertain predicted instances (dashed samples in Fig. 5(d)), which will be determined by the RD comparison in the HM. As the number N_{RD} increases, the prediction accuracy of Q_{ALL} will increase. However, the complexity reduction will decrease since the number of un-optimized CUs (N_{RD}) increases.

Fig. 8 shows the prediction accuracy of the joint classifier consist of two SVM classifiers with different weighted factors at DL 1. We can find that the curved surfaces are symmetrical with the $x = y$ since the two classifiers are the same

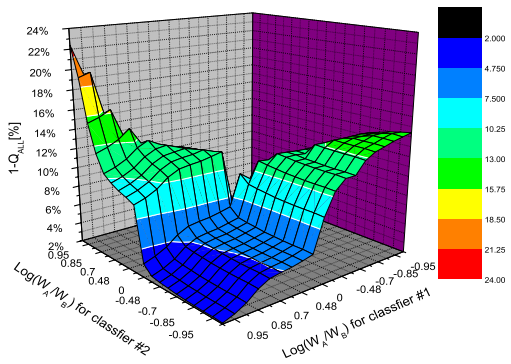


Fig. 8. 1-Q_{ALL} of the joint classifier (DL 1).

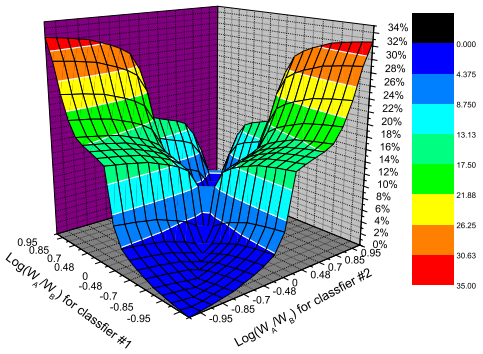


Fig. 9. The ratio of N_{RD} to N_{ALL} (DL 1).

learning machine. When $W_{\mathrm{A}}/W_{\mathrm{B}}$ increases for classifier #1 and decreases for classifier #2, the 1-Q_{ALL} decreases from 22% to 0%. That means that the prediction accuracy of the joint classifier (Q_{ALL}) is able to vary from 78% to 100%, which guarantees the prediction accuracy and RD performance in CU depth decision. Fig. 9 shows the ratio of un-optimized CUs, i.e. N_{RD}. Similarly, curved surfaces are also symmetrical with $x = y$. Meanwhile, the N_{RD} ratio increases as $\mathrm{W_A/W_B}$ increases in classifier #1 and decreases in classifier #2. In other word, the prediction accuracy Q_{ALL} increases; however, the N_{RD} ratio also increases. This direct proportion between Q_{ALL} and N_{RD} ratio indicates the complexity reduction is in direct proportional to the RD degradation. Similar results can be found for DL 2 and 3. Therefore, there is a trade-off between N_{RD} (complexity) and prediction accuracy Q_{ALL} (RD degradation) with respect to the optimal parameters W_A/W_B, which shall be determined.

B. Optimal Weighted Factor Determination

In this sub-section, we firstly analyze RD cost increase and complexity reduction, then model the trade-off between the two performances as an optimization problem. Finally, the optimal solution (W_A/W_B) to the optimization problem is given.

To analyze the RD cost caused by misclassification, we let $\Delta\eta_{S\to nS}(i)$ be the RD cost increase when we use non-split mode to encode the CU of split mode. It is defined as $\Delta\eta_{S\to nS}(i) = (J_{nS}(i)/J_{Best}(i) - 1)\times 100\%$, where i is the CU DL i, $i \in \{1,2,3\}$, $J_{nS}(i)$ and $J_{Best}(i)$ are the RD cost

of using non-split mode and the best mode in encoding the current CU, respectively. Similarly, let $\Delta\eta_{nS\to S}(i)$ be the RD cost increase when we use split mode to encode the non-split CUs and $\Delta\eta_{nS\to S}(i) = (J_S(i)/J_{Best}(i) - 1) \times 100\%$, where $J_S(i)$ is the RD cost of using split mode to encode the current CU. The RD cost increase at DL i is

$$\Delta\eta(i) = \Delta\eta_{nS\to S}(i) \times p_{BA}(i) + \Delta\eta_{S\to nS}(i) \times p_{AB}(i), \tag{8}$$

where $p_{BA}(i) = N_{BA,1}(i)/N_{ALL}(i)$, $p_{AB}(i) = N_{AB,2}(i)/N_{ALL}(i)$ are the ratio of false prediction of split or non-split modes at DL i, $N_{ALL}(i)$ is the total number of CUs that needs prediction in DL i, and N_{ALL} (1) is the number of CUs in a frame.

To analyze the complexity reduction of CU split/non-split prediction, let $q_{S,1}(i) = N_{S,1}(i)/N_{ALL}(i)$ and $q_{nS,2}(i) = N_{nS,2}(i)/N_{ALL}(i)$ be the percentages of split mode and non-split mode predictions by the joint classifier, respectively. $N_{S,1}(i)$ and $N_{nS,2}(i)$ are the number of CU classified as split and non-split mode by the classifier #1 and #2 in the joint classifier. The uncertain prediction is neither split nor non-split prediction. The time reduction of DL i can be calculated as

$$\Delta T(i) = \Delta T_S(i) \times q_{S,1}(i) + \Delta T_{nS}(i) \times q_{nS,2}(i), \tag{9}$$

where $\Delta T_S(i)$ and $\Delta T_{nS}(i)$ are the time saving ratio by using the split and non-split prediction at DL i, $\Delta T_S(i) = 1 - T_S(i)/T_{ALL}(i)$, $\Delta T_{nS}(i) = 1 - T_{nS}(i)/T_{ALL}(i)$, where $T_S(i)$, $T_{nS}(i)$ and $T_{\mathrm{ALL}}(i)$ are complexity of using split, non-split and all modes, respectively.

The optimization problem of each joint classifier can be modeled as minimizing the computational complexity of the encoder $1-\Delta T(i)$ subject to negligible RD cost increase, which can be mathematically presented as

$$\min_{x_i, y_i} 1 - \Delta T(i), \quad s.t. \quad \Delta\eta(i) \le \Delta\eta_{T,i}, \tag{10}$$

where x_{i} and y_{i} are the model parameters for the joint classifier at DL i, i.e. $x_i = \log\left(\frac{W_A(1,i)}{W_B(1,i)}\right)$, $y_i = \log\left(\frac{W_A(2,i)}{W_B(2,i)}\right)$, where i is the DL, 1 and 2 indicate SVM classifier #1 and #2 of the joint classifier. $\Delta\eta_{T,i}$ is upper bound of RD cost increase.

Based on the training data from video coding mentioned in Section IV.A, we collected $p_{BA}(i)$ and $p_{AB}(i)$ when different training factors W_A and W_B were used for the DLs, as shown in Fig. 10. In the figures, the x-axis is the weighted parameter, $\log(W_A/W_B)$, i.e. x_{i} or y_{i}, y-axis is $p_{BA}(i)$ or $p_{AB}(i)$. The dots are real collected data and curves are the fitting results. We fitted $p_{BA}(i)$ and $p_{AB}(i)$ with exponential growth and linear functions, respectively, and the fitting model can be presented as

$$\begin{cases} p_{BA}(i) = b_i + a_i e^{\frac{x_i}{t_i}} \\ p_{AB}(i) = B_i + A_i y_i, \end{cases} \tag{11}$$

where b_i, a_i, t_i, B_i and A_i are model parameters, x_i and y_i are log functions of weighted coefficients (W_A/W_B). The bottom two rows of TABLE II shows the fitting parameters and fitting accuracy (R^2) for Fig. 10. The R^2 values are all higher than 0.954 which means the fitting accuracies are good. Then, applying Eq.11 to Eq.8, the RD cost increase can be presented as

$$\Delta\eta(i) = C_1(i) + k_1(i) e^{\frac{x_i}{t_i}} + k_2(i) y_i \tag{12}$$

where $C_1(i) = \Delta\eta_{S\to nS}(i)B_i + \Delta\eta_{nS\to S}(i)b_i$. $k_1(i) = \Delta\eta_{nS\to S}(i)a_i$, $k_2(i) = \Delta\eta_{S\to nS}(i)A_i$,

Fig. 11 show the percentage of split mode $q_{S,k}(i)$ for a SVM classifier that collected from the off-line training. In the figure, the points with different symbols are the real data and the curves are the exponential fitting of these points. $q_{s,k}(i)$ can be modeled as

$$q_{S,k}(i) = u_i + v_i e^{\frac{x_i}{T_i}}, \tag{13}$$

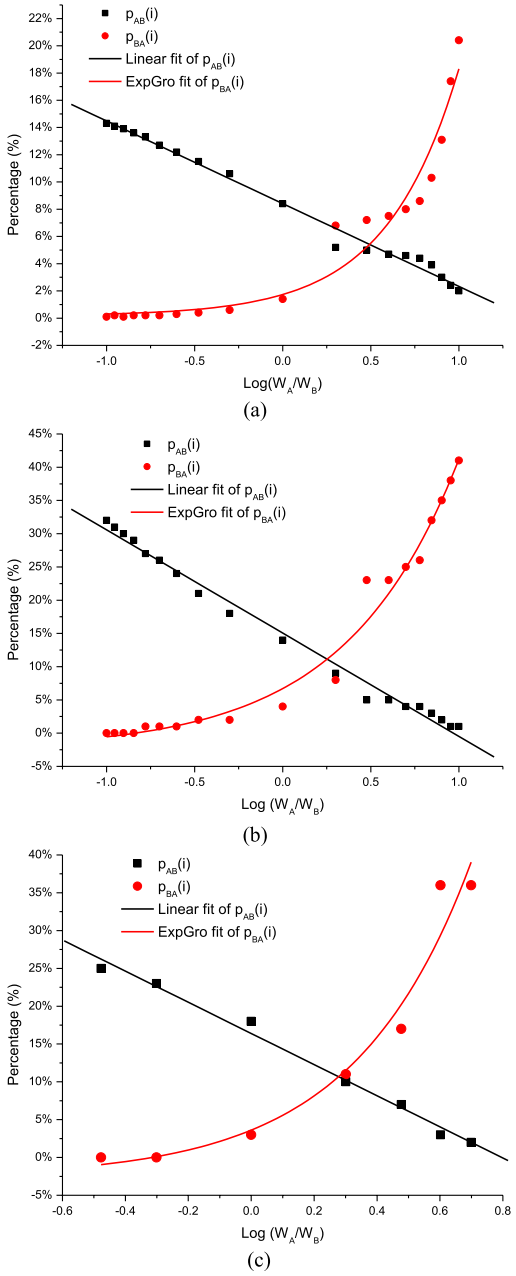


Fig. 10. False Prediction Rate from DL 1 to DL 3 with Different Weighted Factors. (a) DL1. (b) DL 2. (c) DL 3.

TABLE II
FITTING PARAMETERS AND FITTING ACCURACY

Functions and Parameters		DL i		
		1	2	3
$\Delta T_{nS}(i)$		0.784	0.669	0.523
$\Delta T_{S}(i)$		0.216	0.331	0.477
$\Delta\eta_{S\to nS}(i)$		1.185	1.089	1.026
$\Delta\eta_{nS\to S}(i)$		1.014	1.038	1.057
$q_{S,k}(i)=u_i+v_i e^{\frac{x_i}{T_i}}$	u_i	0.118	-0.161	-0.195
	v_i	0.103	0.493	0.375
	T_i	0.885	1.522	1.126
	R^2	**0.968**	**0.991**	**0.963**
$p_{BA}(i)=b_i+a_i e^{\frac{x_i}{t_i}}$	b_i	0.002	-0.025	-0.027
	a_i	0.016	0.092	0.063
	t_i	0.408	0.642	0.370
	R^2	**0.954**	**0.982**	**0.957**
$p_{AB}(i)=B_i+A_i y_i$	B_i	0.084	0.151	0.164
	A_i	-0.061	-0.155	-0.206
	R^2	**0.990**	**0.989**	**0.988**

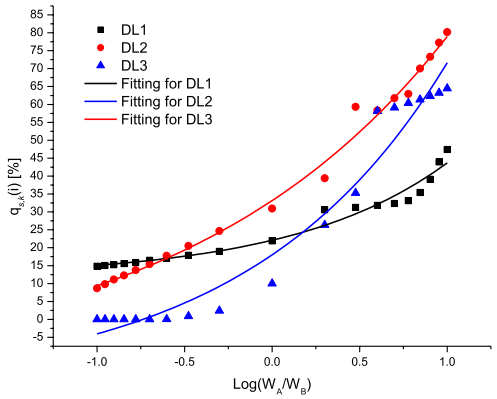


Fig. 11. Percentage of Split Mode $q_{S,k}(i)$ in DL 1 to DL 3.

where u_i, v_i and T_i are parameters for DL i. In terms of the same SVM classifier #k, $q_{nS,k}(i) = 1 - q_{S,k}(i)$. Therefore, apply Eq.13 to Eq.9, we can obtain the complexity reduction of each DL as

$$\Delta T(i) = C_2(i) + h_1(i)\, e^{\frac{x_i}{T_i}} - h_2(i)\, e^{\frac{y_i}{T_i}}, \tag{14}$$

where $C_2(i) = \Delta T_{nS}(i)u_i + \Delta T_S(i)(1-u_i)$, $h_1(i) = \Delta T_S(i)v_i$, $h_2(i) = \Delta T_{nS}(i)v_i$. The middle rows of TABLE II shows the fitting parameters and fitting accuracy for the $q_{s,k}(i)$ in Fig. 11, where the R^2 value are higher than 0.963. In addition, TABLE II also shows the average $\Delta T_{nS}(i)$, $\Delta T_S(i)$, $\Delta\eta_{S\to nS}(i)$ and $\Delta\eta_{nS\to S}(i)$ that are statistically collected from the training videos.

To solve the problem in Eq.10, we introduce the Lagrange multiplier λ_i, and the optimization becomes

$$\begin{cases} \{x_i, y_i\} = \arg\min\limits_{x_i, y_i} J(i) \\ J(i) = 1 - \Delta T(i) + \lambda_i \left(\Delta\eta(i) - \Delta\eta_{T,i}\right). \end{cases} \tag{15}$$

We apply Eqs.12 and 14 to Eq.15 and it is easy to prove that Eq.15 is convex. To get the optimal x_i and y_i, we take the partial derivation with respect to x_i, y_i and λ_i, then set them to 0 and get

$$\begin{cases} \dfrac{\partial J(i)}{\partial x_i} = \dfrac{\partial\left(-h_1(i)\, e^{\frac{x_i}{T_i}} + \lambda_i k_1 e^{\frac{x_i}{t_i}}\right)}{\partial x_i} \\ \qquad = -\frac{h_1(i)}{T_i} e^{\frac{x_i}{T_i}} + \frac{\lambda_i k_1(i)}{t_i} e^{\frac{x_i}{t_i}} = 0 \\ \dfrac{\partial J(i)}{\partial y_i} = \dfrac{\partial\left(\lambda_i k_2(i)\, y_i + h_2(i)\, e^{\frac{y_i}{T_i}}\right)}{\partial y_i} \\ \qquad = \lambda_i k_2(i) + \dfrac{h_2(i)}{T_i} e^{\frac{y_i}{T_i}} = 0 \\ \dfrac{\partial J(i)}{\partial \lambda_i} = C_1(i) + k_1(i)\, e^{\frac{x_i}{t_i}} + k_2(i)\, y_i - \Delta\eta_{T,i} = 0. \end{cases} \tag{16}$$

Solve these equations, we obtain

$$\begin{cases} x_i = \dfrac{t_i T_i}{T_i - t_i} \ln \dfrac{t_i h_1(i)}{\lambda_i T_i k_1(i)} \\ y_i = T_i \ln \dfrac{-\lambda_i k_2(i) T_i}{h_2(i)}, \end{cases} \tag{17}$$

$$C_1(i) + k_1(i)\left(\frac{t_i h_1(i)}{\lambda_i T_i k_1(i)}\right)^{\frac{T_i}{T_i - t_i}} + k_2(i) T_i \ln \frac{-\lambda_i k_2(i) T_i}{h_2(i)} - \Delta\eta_{T,i} = 0 \tag{18}$$

In Eq.18, λ_i is a function of $\Delta\eta_{T,i}$. Only $\Delta\eta_{T,i}$ is a parameter and other variables are constants. Though the mapping function cannot be explicitly presented, it can be calculated by least square method with a given $\Delta\eta_{T,i}$. Then, apply this λ_i to Eq.17 and we can get the optimal x_i, y_i (i.e. W_A/W_B) for the joint classifier. $\Delta\eta_{T,i}$ is the RD increase ratio for DL i. Basically, $\Delta\eta_{T,i}$ indicates the allowable percentage of RD cost increase for each DL and it can be given according to the complexity and coding efficiency requirements of the video encoding system.

V. EXPERIMENTAL RESULTS AND ANALYSES

To evaluate the performance of the proposed algorithm, we implemented the proposed algorithm and benchmarks on the HEVC reference software HM12.0 [24]. Low delay B main profile was used in the coding experiments, the first frame was encoded with INTRA and the rest of the frames were encoded with low delay B frames. The sizes of LCU and SCU are 64 × 64 and 8 × 8, respectively, which means the maximum CU depth is 4. GOP size is 4. The minimum and maximum RQT transform size is 4 and 32, respectively. Motion search range is 64. Other parameters were set as default. The proposed algorithm was implemented on HM12.0 and tested with the common test conditions [25]. C-SVM [23] was adopted in model training and CU depth predicting in the proposed encoder. C_P is set as 100 and the rest parameters of the C-SVM classifiers were set as default. All the video coding experiments were performed on computer with CPU AMD Athlon IIX2 B24, 2.99GHz, 2 GB memory, Window XP operating system. Bjonteggard Delta Bit Rate (BDBR) and Peak-Signal-to-Noise Ratio (BDPSNR) [26] are applied to evaluate the RD performance of different schemes as compared with the original HM. Additionally, time saving (ΔT) is

employed to measure the time reduction of the tested schemes, which is defined as

$$\Delta T = \frac{1}{4}\sum_{i=1}^{4}\frac{T_{HM}(QP_i) - T_{\psi}(QP_i)}{T_{HM}(QP_i)} \cdot 100\%, \quad (19)$$

where $T_{\mathrm{HM}}(QP_i)$ and $T_{\Psi}(QP_i)$ are the encoding time of using the original HM [24] and scheme Ψ with QP_i.

There are two phases of the coding performance evaluation. One is testing the coding performance of the proposed algorithm while using different parameters, and the other is make comparison between the proposed algorithm and the state-of-the-art benchmarks.

A. Coding Performance of Using Different Parameters ($\Delta\eta_{T,i}$)

We tested the coding performance of using different parameters for the proposed algorithm. The lower DL usually has higher potential of complexity reduction. Thus, we shall give the lower DL with higher $\Delta\eta_{T,i}$ to maximize the total ΔT. In order to verify this phenomena, we tested the proposed algorithm with different $\Delta\eta_{T,i}$ at the three DLs. Five test sequences with different resolutions, including *Basketballpass* (416 × 240), *Partyscene* (832 × 480), *Johnny* (1280 × 720), *Kimono* (1920 × 1080), *Traffic* (2560 × 1600), were encoded with different $\Delta\eta_{T,i}$. 100 frames were encoded for each sequence. The first set of $\Delta\eta_{T,i}$ for the three DLs $\{\Delta\eta_{T,1},\Delta\eta_{T,2},\Delta\eta_{T,3}\}$ are set equally, which are {0.1%,0.1%, 0.1%}, {0.3%,0.3%,0.3%}, {0.5%, 0.5%,0.5%}, {0.7%,0.7%, 0.7%}, denoted as Para_111, Para_333, Para_555 and Para_777. In addition, {0.3%, 0.2%, 0.1%}, {0.6%, 0.4%, 0.2%}, {0.9%, 0.6%, 0.3%}, denoted as Para_321, Para_642, Para_963, are also tested. With a given $\{\Delta\eta_{T,1},\Delta\eta_{T,2},\Delta\eta_{T,3}\}$, the first 20 frames of *BQMall* and *FourPeople* were encoded with four QPs {24,28,32,36} by the original HM, and then the features and class labels are input in SVM training module. The trained SVM models with different W_{A} and W_{B} that calculated from Eq.17 with given $\Delta\eta_{T,i}$ are then input in the proposed encoder to encode the five test sequences.

Fig. 12 show the relationship between average complexity reduction and average BDBR/BDPSNR for different parameter sets. The average BDBR/BDPSNR and complexity reduction are average values over the five test sequences. In the figure, we can observe that 1) the BDBR, BDPSNR and complexity reduction increases as we choose larger $\Delta\eta_{T,i}$. 2) The BDBR is generally in direct proportional to the complexity and BDPSNR is in inverse proportional to the complexity. 3) The red and black lines are formulated by connecting solid and hollow dots respectively. The red solid line is above the black dash line which means the in-equally parameter set has a better performance than the equal parameter setting, since it reduces more computational complexity at the same BDPSNR/BDBR. In this paper, the optimization target is to minimize the computational complexity subject negligible RD degradation, we therefore select Para_642 set since it has a better trade-off between the two indices and BDBR is within 1.68%. The optimal parameters $\Delta\eta_{T,i}$ shall be selected based on the coding system's requirements. Also, they may changes along tested video content. There might be other parameter sets that can allocate the complexity better. However, they are not fully

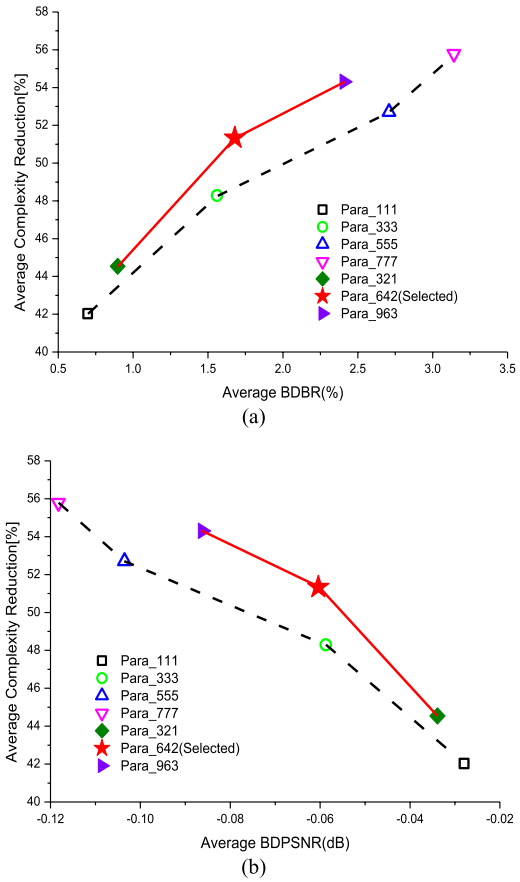


Fig. 12. Relation Between Average Complexity Reduction and Average BDBR/BDPSNR over Different $\Delta\eta_{T,i}$ Sets. (a) Complexity reduction vs BDBR. (b) Complexity reduction vs BDPSNR.

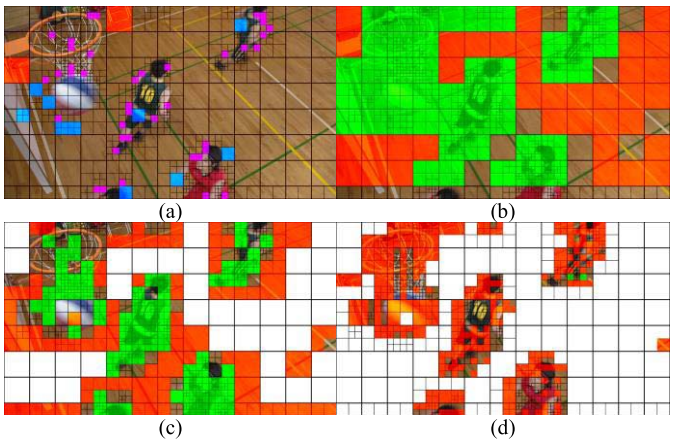


Fig. 13. CU Prediction by the Proposed Algorithm (BasketballDrill, 832×480). (a) False predicted CUs. (b) CU prediction in DL1. (c) CU prediction in DL2. (d) CU prediction in DL3.

enumerated in this paper. Actually, there is "Pareto front" and "Pareto optimal" of selecting $\Delta\eta_{T,i}$ set, which can be solved and presented by the multi-objective optimization.

Figs. 13 and 14 show the CU prediction by the proposed algorithm when *QP* equals 28 for *BasketballDrill* and *ParkScene* sequence, where the grids in the pictures are the ground truth CU partition produced by the original HM. The Para_642 is selected for the proposed encoder.

TABLE III

RD AND COMPLEXITY REDUCTION COMPARISONS BETWEEN THE PROPOSED ALGORITHM AND THE BENCHMARKS (UNIT: %/dB/%)

Test Sequences	Resolution	ShenEVIP[3] vs HM			ShenTMM[5] vs HM			XiongTMM[19] vs HM			The Proposed vs HM		
		BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT
Basketballpass	416×240	5.21	-0.251	-28.37	2.29	-0.111	-27.92	2.61	-0.128	-40.85	1.19	-0.059	-36.63
BlowingBubbles		5.02	-0.195	-34.49	1.20	-0.048	-29.56	4.57	-0.179	-26.73	0.89	-0.036	-28.82
BQSquare		4.62	-0.161	-40.39	1.63	-0.057	-31.95	3.16	-0.108	-29.50	0.84	-0.030	-30.67
RaceHorses		5.49	-0.247	-22.46	1.53	-0.070	-27.82	3.95	-0.183	-30.39	1.21	-0.053	-33.37
BasketballDrill	832×480	4.63	-0.179	-39.49	2.48	-0.097	-27.98	7.10	-0.271	-38.93	1.70	-0.068	-47.58
BQMall		5.26	-0.201	-41.55	2.58	-0.100	-30.56	5.98	-0.229	-37.12	1.58	-0.063	-45.96
Mobisode2		4.78	-0.120	-62.40	5.77	-0.150	-43.38	9.13	-0.231	-53.77	3.70	-0.090	-62.85
PartyScene		5.41	-0.213	-35.94	0.93	-0.038	-28.71	5.48	-0.222	-32.57	1.30	-0.053	-36.63
FourPeople	1280×720	5.39	-0.175	-64.80	3.93	-0.135	-51.78	11.72	-0.372	-58.36	2.83	-0.097	-66.35
Johnny		4.37	-0.098	-70.32	3.64	-0.091	-55.00	8.86	-0.213	-61.13	2.48	-0.057	-70.93
KristenAndSara		4.12	-0.114	-67.41	5.20	-0.144	-48.61	7.14	-0.197	-57.57	2.29	-0.067	-68.23
Vidyo1		4.28	-0.129	-65.53	2.00	-0.064	-48.43	8.57	-0.263	-60.16	2.66	-0.079	-68.17
Vidyo3		6.02	-0.167	-65.62	2.79	-0.073	-48.33	7.63	-0.228	-58.92	3.22	-0.095	-67.51
Vidyo4		4.55	-0.124	-66.63	4.17	-0.110	-50.33	6.42	-0.173	-59.17	2.90	-0.077	-67.77
BasketballDrive	1920×1080	3.01	-0.071	-48.35	1.19	-0.026	-26.64	10.11	-0.230	-52.08	2.21	-0.054	-50.96
BQTerrace		2.76	-0.046	-50.88	1.39	-0.026	-36.57	5.99	-0.100	-40.16	2.10	-0.037	-45.07
Cactus		5.20	-0.113	-49.00	2.25	-0.052	-36.49	11.94	-0.271	-52.01	1.77	-0.043	-48.44
Kimono		4.83	-0.147	-52.37	1.36	-0.044	-37.04	6.35	-0.198	-46.63	2.04	-0.068	-56.13
ParkScene		4.49	-0.139	-49.51	0.90	-0.028	-34.47	5.29	-0.163	-42.30	1.46	-0.045	-47.82
PeopleOnStreet	2560×1600	4.93	-0.221	-26.46	1.21	-0.055	-24.55	10.95	-0.478	-32.03	1.24	-0.062	-44.44
Traffic		4.88	-0.135	-52.37	1.35	-0.032	-37.04	6.36	-0.170	-46.63	2.03	-0.051	-56.13
Average		**4.73**	**-0.155**	**-49.25**	**2.37**	**-0.074**	**-37.29**	**7.11**	**-0.219**	**-45.57**	**1.98**	**-0.061**	**-51.45**

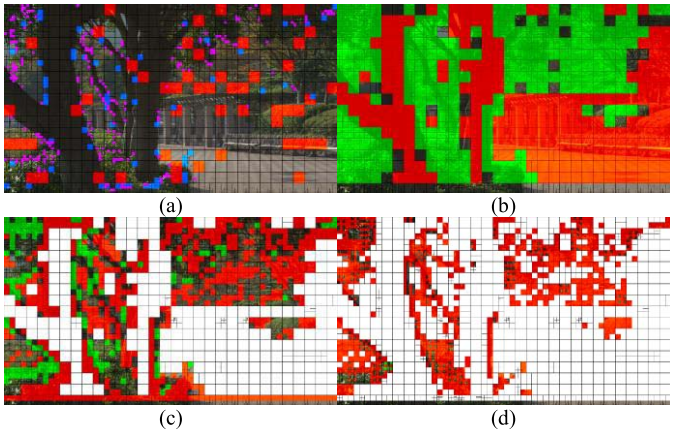


Fig. 14. CU Prediction by the Proposed Algorithm (ParkScene, 1920×1080). (a) False predicted CUs. (b) CU prediction in DL1. (c) CU prediction in DL2. (d) CU prediction in DL3.

Figs. 13(a) and 14(a) show the false predicted CUs that are labeled with red, blue and purple color for DL1 to DL3. Figs. 13(b) to 13(d) and Figs 14(b) to 14(d) show the CU prediction for DL1 to DL3, where red color blocks are non-split prediction, green are split prediction, the original blocks are uncertain prediction which are checked by the RDO process. The white blocks are the CUs early terminated by the upper DL and are not necessary to be checked in the current DL. We can have the following three observations; 1) the number of false predicted CUs is relative small compared the whole image and the prediction accuracy in each DL is generally higher than 90%. 2) Most CUs are effectively predicted by the proposed algorithm with split or non-split mode, and only a few CUs are predicted with uncertain. 3) The red color blocks possess a large proportion which early terminate the recursive CU checking process that is not necessary to be further checked. For the green blocks, they skip the checking of the current CU depth and will be checked in next DLs. Actually, when we select model trained from stricter $\Delta\eta_{T,i}$, e.g. Para_321 and Para_111, the number of false predicted CUs reduces. However, the number of blocks with uncertain prediction in DL1 to DL3 (the original blocks) increases, which indicates the proposed encoder can adapt to applications with different complexity and RD requirements.

B. Coding Performance Comparison Among the Proposed Algorithm and the State-of-the-Art Schemes

In this phase, we evaluated the coding performance of the proposed algorithm in a more comprehensive way and make a comparison with three benchmarks, including Xiong's scheme [3] (denoted by XiongTMM) X. Shen's scheme [19] (denoted by ShenJIVP) and L. Shen's scheme [5] (denoted by ShenTMM). Twenty-one diverse test sequences from Class A to Class E were employed in the coding experiment, and all frames of the test sequences were encoded with four QPs, which are 22, 27, 32 and 37. Low delay B main configuration was used and encoding settings followed the common test conditions [25]. The SVM models used the proposed encoder were trained with Para_642 and training dataset from the first 20 frames of *BQMall* and *FourPeople* encoded by the original HM with four QPs {24,28,32,36}.

The BDBR, BDPSNR and ΔT comparisons among the proposed algorithm and the benchmarks are shown in TABLE III. In ShenEVIP scheme, the training sequences, number of

frames and encoding *QP*s are the same as the proposed algorithm. We can observe that ShenEVIP can reduce the complexity from 26.46% to 70.32%, 49.25% on average compared with the original HM. The BDBR and BDPSNR between ShenEVIP and HM are 4.73% and −0.155dB on average, respectively. Though ShenEVIP achieves a significant complexity reduction, the RD degradation is also significant. The main reason is that it is difficult to maintain sufficient high prediction accuracy of the classifier for different QPs and various contents, and misclassification causes RD degradation. The complexity reduction of ShenTMM is from 24.55% to 55.00%, and 37.29% on average. The BDBR and BDPSNR between ShenTMM and HM are 2.37% and −0.074dB. XiongTMM scheme achieves complexity reduction from 26.73% to 61.13%, and 45.57% on average. However, the BDPSNR degradation is −0.219dB and BDBR increase is 7.11%. For this scheme, the FIFO structure of CU mode prediction may lead to error prorogation as well as RD degradation when using different QPs along the frames. In addition, the FIFO structure can hardly fully exploit the spatial and temporal correlation of the video content, since the upper CU and temporal collocated CU may be out of the FIFOs, especially for HD videos.

As for the proposed algorithm, it achieves complexity reduction from 28.82% to 70.93%, and 51.45% on average, which is better than those of ShenEVIP, ShenTMM and XiongTMM. Meanwhile, the BDPSNR degradation between the proposed algorithm and HM is within 0.030dB to 0.097dB, 0.061dB on average. The BDBR increase is from 0.84% to 3.70%, 1.98% on average. In addition, we also tested the proposed algorithm with another training model that trained from "BasketballDrill" and "KristenAndSara" sequences with four QPs {24,28,32,36}. Similar coding results can be found. From the overall performance evaluation we can find that the proposed fast CU depth decision algorithm can achieve more complexity reduction and cost less RD degradation when compared with the benchmarks.

VI. Conclusion

In this paper, we proposed an efficient machine learning based CU depth decision method. Firstly, we analyze CU depth decision process in HEVC and model the CU depth decision process as a three-level of hierarchical decision problem. Secondly, we present an improved CU depth decision structure which allows the performances of the CU depth decision can be transferred between the complexity and RD cost. Then, a three-output classifier is designed to control the risk of false prediction. Finally, a sophisticated RD-complexity model is derived for the optimal training parameter determination, which is capable of allocating the computation complexity of each CU DL at given allowable RD cost increase. The comparative experimental results demonstrate that the proposed algorithm is effective.

References

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

[2] X. Li, J. An, X. Guo, and S. Lei, *Adaptive CU Depth Range*, Joint Collaborative Team on Video Coding, document Rec. JCTVC-E090, Geneva, Switzerland, Apr. 2011, pp. 16–23.

[3] J. Xiong, H. Li, Q. Wu, and F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 559–564, Feb. 2014.

[4] L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 207–213, Feb. 2013.

[5] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An effective CU size decision method for HEVC encoders," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 465–470, Feb. 2013.

[6] K. Goswami, B.-G. Kim, D.-S. Jun, S.-H. Jung, and J. S. Choi, "Early coding unit–splitting termination algorithm for high efficiency video coding (HEVC)," *ETRI J.*, vol. 36, no. 3, pp. 407–417, Jun. 2014.

[7] C.-S. Park, B.-G. Kim, G.-S. Hong, and S.-K. Kim, "Fast coding unit (CU) depth decision algorithm for high efficiency video coding (HEVC)," in *Advances in Computer Science and Its Applications* (Lecture Notes in Electrical Engineering), vol. 279. Berlin, Germany: Springer-Verlag, 2014, pp. 293–299.

[8] A. Lee, D. Jun, J. Kim, J. S. Choi, and J. Kim, "Efficient inter prediction mode decision method for fast motion estimation in high efficiency video coding," *ETRI J.*, vol. 36, no. 4, pp. 528–536, Aug. 2014.

[9] T. Zhao, Z. Wang, and S. Kwong, "Flexible mode selection and complexity allocation in high efficiency video coding," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1135–1144, Dec. 2013.

[10] Z. Pan, S. Kwong, M.-T. Sun, and J. Lei, "Early MERGE mode decision based on motion estimation and hierarchical depth correlation for HEVC," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 405–412, Jun. 2014.

[11] E. Martinez-Enriquez, A. Jimenez-Moreno, M. Angel-Pellon, and F. Diaz-de-Maria, "A two-level classification-based approach to inter mode decision in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1719–1732, Nov. 2011.

[12] Y. Zhang, S. Kwong, G. Jiang, X. Wang, and M. Yu, "Statistical early termination model for fast mode decision and reference frame selection in multiview video coding," *IEEE Trans. Broadcast.*, vol. 58, no. 1, pp. 10–23, Mar. 2012.

[13] Y. Zhang, S. Kwong, L. Xu, and G. Jiang, "DIRECT mode early decision optimization based on rate distortion cost property and inter-view correlation," *IEEE Trans. Broadcast.*, vol. 59, no. 2, pp. 390–398, Jun. 2013.

[14] Y.-H. Sung and J.-C. Wang, "Fast mode decision for H.264/AVC based on rate-distortion clustering," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 693–702, Jun. 2012.

[15] J.-C. Chiang, W.-C. Chen, L.-M. Liu, K.-F. Hsu, and W.-N. Lie, "A fast H.264/AVC-based stereo video encoding algorithm based on hierarchical two-stage neural classification," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 2, pp. 309–320, Apr. 2011.

[16] P. Carrillo, T. Pin, and H. Kalva, "Low complexity H.264 video encoder design using machine learning techniques," in *Dig. Tech. Papers ICCE*, Jan. 2010, pp. 461–462.

[17] T. Yu, Y. Zhang, and P. C. Cosman, "Classification based fast mode decision for stereo video coding," in *Proc. 20th IEEE ICIP*, Melbourne, VIC, Australia, Sep. 2013, pp. 1724–1728.

[18] C. Kim and C.-C. J. Kuo, "Feature-based intra-/InterCoding mode selection for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 441–453, Apr. 2007.

[19] X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP J. Image Video Process.*, vol. 2013, Jan. 2013, Art. ID 4.

[20] T. Shanableh, E. Peixoto, and E. Izquierdo, "MPEG-2 to HEVC video transcoding with content-based modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1191–1196, Jul. 2013.

[21] E. Peixoto, T. Shanableh, and E. Izquierdo, "H.264/AVC to HEVC video transcoder based on dynamic thresholding and content modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 99–112, Jan. 2014.

[22] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. ID 27. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[24] (2013). *HM Software*. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-12.0

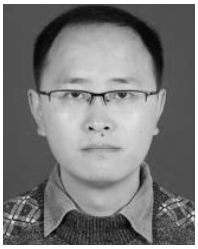
[25] F. Bossen, *Common Test Conditions and Software Reference Configurations*, Joint Collaborative Team on Video Coding, document Rec. JCTVC-L1100, Jan. 2013.
[26] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document Rec. ITU-T VCEG-M33, Austin, TX, USA, Apr. 2001.

Yun Zhang (M'12) received the B.S. and M.S. degrees in electrical engineering from Ningbo University, Ningbo, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, China, in 2010. From 2009 to 2014, he was a Post-Doctoral Researcher with the Department of Computer Science, The City University of Hong Kong, Hong Kong. In 2010, he became an Assistant Professor with the Shenzhen Institutes of Advanced Technology, CAS, where he has been an Associate Professor since 2012. His research interests are video compression, 3D video processing, and visual perception.

Sam Kwong (M'93–SM'04–F'13) received the B.S. degree from the State University of New York at Buffalo, in 1983, the M.S. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1985, and the Ph.D. degree from the University of Hagen, Germany, in 1996. From 1985 to 1987, he was a Diagnostic Engineer with Control Data Canada. He joined Bell Northern Research Canada as a member of Scientific Staff. In 1990, he became a Lecturer with the Department of Electronic Engineering, The City University of Hong Kong, where he is currently a Professor with the Department of Computer Science. His research interests are video and image coding, and evolutionary algorithms.

Xu Wang (M'15) received the B.S. degree from South China Normal University, Guangzhou, China, in 2007, the M.S. degree from Ningbo University, Ningbo, China, in 2010, and the Ph.D. degree from the Department of Computer Science, The City University of Hong Kong, Hong Kong, in 2014. He joined the College of Computer Science and Software Engineering, Shenzhen University, in 2015, as an Assistant Professor. His research interests are video coding and stereoscopic image/video quality assessment.

Hui Yuan (S'08–M'12) received the B.E. and Ph.D. degrees in telecommunication engineering from Xidian University, Xi'an, China, in 2006 and 2011, respectively. Since 2011, he has been a Lecturer and an Associate Professor with the School of Information Science and Engineering, Shandong University, Jinan, China. Since 2013, he has also been a Post-Doctoral Fellow with the Department of Computer Science, The City University of Hong Kong, Hong Kong. His current research interests include video coding and multimedia communication.



Zhaoqing Pan received the B.S. degree in computer science and technology from Yancheng Normal University, Yancheng, China, in 2009, and the Ph.D. degree in computer science from The City University of Hong Kong, Hong Kong, in 2014. In 2013, he was a Visiting Scholar with the Department of Electrical Engineering, University of Washington, Seattle, WA, USA. He is currently a Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. His research interest focuses on video compression/coding.

Long Xu received the M.S. degree in applied mathematics from Xidian University, China, in 2002, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2009. He was a Post-Doctoral Researcher with the Department of Computer Science, The City University of Hong Kong, and the Department of Electronic Engineering, The Chinese University of Hong Kong, from 2009 to 2012. From 2013 to 2014, he was a Research Fellow with the School of Computer and Engineering, Nanyang Technological University. He is currently with the Key Laboratory of Solar Activity, National Astronomical Observatories, Chinese Academy of Sciences. His research interests include image/video coding, solar radio astronomy, image quality assessment, compressed sensing, image/video processing, and machine learning. He was selected into the 100-Talents Program, Chinese Academy of Sciences, in 2014.