Journal of Real-Time Image Processing (2020) 17:1029–1049
https://doi.org/10.1007/s11554-019-00854-1

ORIGINAL RESEARCH PAPER

Fast video encoding based on random forests

Muhammad Tahir[1] · Imtiaz A. Taj[1] · Pedro A. Assuncao[2] · Muhammad Asif[3]

Received: 2 June 2018 / Accepted: 22 January 2019 / Published online: 5 February 2019

Abstract
Machine learning approaches have been increasingly used to reduce the high computational complexity of high-efficiency video coding (HEVC), as this is a major limiting factor for real-time implementations, due to the decision process required to find optimal coding modes and partition sizes for the quad-tree data structures defined by the standard. This paper proposes a systematic approach to reduce the computational complexity of HEVC based on an ensemble of online and offline Random Forests classifiers. A reduced set of features for training the Random Forests classifier is proposed, based on the rankings obtained from information gain and a wrapper-based approach. The best model parameters are also obtained through a consistent and generalizable method. The proposed Random Forests classifier is used to model the coding unit and transform unit-splitting decision and the SKIP-mode prediction, as binary classification problems, taking advantage from the combination of online and offline approaches, which adapts better to the dynamic characteristics of video content. Experimental results show that, on average, the proposed approach reduces the computational complexity of HEVC by 62.64% for the random access (RA) profile and 54.57% for the low-delay (LD) main profile, with an increase in BD-Rate of 2.58% for RA and 2.97% for LD, respectively. These results outperform the previous works also using ensemble classifiers for the same purpose.

Keywords Fast video coding · HEVC · Random forests in HEVC · Machine learning in HEVC

1 Introduction

High-efficiency video coding (HEVC) a.k.a H.265 offers many advanced features. It supports demanding applications and services like Ultra High Definition TV (UHD-TV) with display resolutions 4 K × 2 K, or higher, and multi-view video with higher dynamic range in terms of color content. In HEVC, several new coding structures like Coding Unit (CU), Prediction Unit (PU), and Transform Unit (TU) have been introduced to improve the coding efficiency. Each video frame is first divided into Coding Tree Units (CTU) of size 64 × 64. CTU is further divided into multiple square size CUs at different CTU depth levels, which can vary from 0 to 3 according to the quad-tree structure. The CU size at depth level 0 is 64 × 64, whilst, at depth level 3, this is 8 × 8 [1]. For encoding, at each CU depth level, the rate-distortion cost (RD) is evaluated for different prediction modes and for each PU, TUs of different sizes are also evaluated. The size of TUs can vary from 32 × 32 to 4 × 4. Together, these CUs and TUs at different depth levels form a quad-tree, as shown in Fig. 1. Similarly, for each PU, several intra- and inter-prediction modes are evaluated. These modes include intra N × N, intra 2N × 2N, Merge/SKIP mode (MSM), inter 2N × 2N, inter N × 2N, inter 2N × N, inter N × N, inter 2N × nU, inter 2N × nD, inter nL × 2N, and inter nR × 2N. The MSM is used to merge the current PU with its neighbors when they have the same motion information. The SKIP mode on its own is a special case of MSM when all coded

✉ Muhammad Tahir
mtahir.awan@yahoo.com

Imtiaz A. Taj
imtiaztaj@cust.edu.pk

Pedro A. Assuncao
amado@co.it.pt

Muhammad Asif
astz786@yahoo.com

1 Department of Electrical Engineering, Capital University of Science and Technology, Islamabad, Pakistan

2 Polytechnic Institute of Leiria and Instituto de Telecomunicacoes (IT), Morro do Lena-Alto do Vieiro, 2411-901 Leiria, Portugal

3 Department of Computer Science, Lahore Garrison University, Lahore, Pakistan

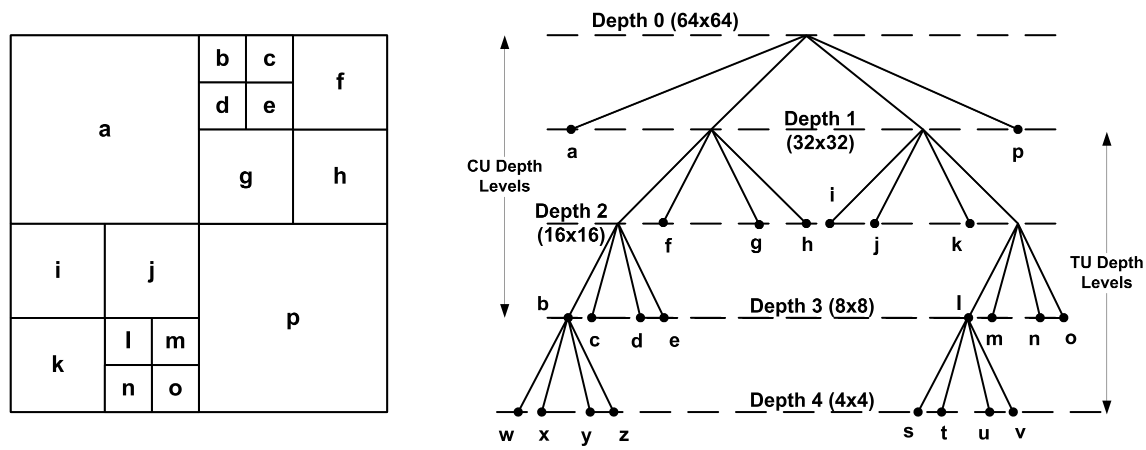


Fig. 1 HEVC CU and TU quad-tree structures

block flags are zero, and thus, it is mostly selected as the optimal prediction mode for coding homogeneous regions in video frames with very little or no motion. These prediction modes, combined with the nested CU and TU structures, have significantly increased the computational complexity of HEVC, and thus, real-time implementation on platforms with limited computational resources is quite challenging.

In HEVC implementation, the quad-tree structure for CU is the main responsible factor for making the encoding decision process highly complex and computationally intensive, because every possible CU size from 64×64 down to 8×8 is explored to find the optimal depth level, based on the RD optimization through exhaustive search. Similarly, within each CU, checking all possible TU sizes at different depth levels and PUs to select the optimal one is also computationally intensive [2]. For real-time implementation of HEVC, these tasks cannot be based on exhaustive search approaches.

In the recent past, various techniques for fast encoding have been proposed for HEVC using different approaches, which can be categorized as (1) methods based on statistical parameters and spatio-temporal correlations and (2) methods using machine learning approaches. Spatio-temporal correlation methods exploit spatial and temporal similarities in video data and often reuse coding information about spatial neighbors in the current frame and temporal neighboring blocks from adjacent frames. These techniques often use soft or hard thresholds to predict block type, prediction modes, and motion vectors of the current PU (e.g., [3, 4]). On the other hand, machine learning approaches are based upon classification models, which use appropriate features extracted from the video data and also from the encoding process itself to train a specific type of classifier (e.g., [5, 6]). Then, such a classifier model is used to predict block types, partition sizes, and prediction modes with much lower computational complexity than the other methods based on exhaustive search.

Random forests (RF) is an ensemble classifier composed of multiple decision trees and a random set of features is used in each decision tree [7]. Random selection of features along with majority voting among multiple decision trees enables the classifier to easily adapt to the changes in input data. This work exploits the RF characteristics to achieve fast-encoding decisions in the process of determining the best CU and TU depths and also whether a PU should be skipped. An efficient feature selection methodology is followed, by considering two distinct approaches: information gain filtering and a wrapper method. Then, an ensemble of online and offline RF classifiers is investigated to model the CU and TU splitting decision and also the SKIP-mode prediction as binary classification problems. In comparison with the other state-of-art works using similar approaches, better results are achieved as demonstrated throughout 20 test sequences.

The paper is organized as follows. Section 2 presents a review of the most recent related work and then Sect. 3 describes the feature selection methodology. Section 4 explains the proposed classifier design and training. Section 5 presents the simulation results, and finally, Sect. 6 concludes the paper.

2 Related work

Fast-coding methods based on spatio-temporal correlations are among those mostly described in the literature. Rhee et al. [8] have surveyed many fast-encoding algorithms used for inter-prediction and coding mode decisions in H.264 and discussed their application to HEVC. Sun et al. [9] have used direction variance for fast intra-mode decision and efficient CU partitioning in HEVC. Liquan et al. [10] proposed an adaptive inter-mode decision scheme for HEVC that exploits correlations between spatio-temporal parameters of adjacent CUs and different CU depth levels to predict the best mode for the current CU, including the SKIP mode. Lin et al. [11] proposed two early termination algorithms for optimal CU depth selection that use PU residual histogram in the current CU and SKIP information of neighboring blocks for the early termination of CU depth search. Similarly, Tai et al. [12] proposed an early CU Split scheme that also skips unnecessary PU predictions to reduce computational complexity of HEVC. Their approach uses depth information of co-located CUs and RD cost correlation of different prediction modes for the early termination of the CU Split procedure. Chen et al. [13] also proposed a fast inter-coding algorithm for HEVC. Their scheme first divides the CU into static or motion block, and then, spatio-temporal correlations are used to determine the CU depth range and prediction mode. Hoyoung et al. [4] proposed an early SKIP-mode decision scheme, with an emphasis on video quality, using thresholds on the RD cost of merge mode for the early decision of SKIP mode. Similarly, [14–17] have used fast-encoding approaches for inter-mode decision in HEVC, based on spatio-temporal neighbors. These techniques that use the behavior of spatial and co-located neighboring blocks for prediction and speed-up, often use fixed thresholds for prediction, and exit strategies. Therefore, their performance is highly susceptible to scene changes and do not easily adapt to the inherent variability of video content.

Machine learning approaches for fast coding of HEVC have also been gaining relevance in recent years. Shen et al. [18] have modeled the CU size decision as a binary classification problem, where the CU Split decision is taken upon a Bayesian decision rule. A set of features is calculated for each CU and the split decision is taken using the conditional probability density functions (pdfs) of respective classes. A mutual information parameter is used to rank and short list the feature set. Liquan et al. [19] have also used Bayesian decision theory for the early termination of TU size decision that uses the variance of residual coefficients for the early termination of TU Split procedure. Xiong et al. [20] proposed a pyramid motion divergence approach for CU size selection in HEVC. Optical flow is estimated for downsampled video frames and the variance of the optical flows of current CU and sub-CUs is used as a motion divergence parameter. A k-nearest neighbor classifier is then used to take the split decision at each CU depth level. However, optical flow and motion divergence computation at pixel level over the entire frame is a computational overhead in this approach.

Zhang et al. [5] proposed a machine learning approach for fast CU depth decision in HEVC. The authors model the CU Split decision as a three-level hierarchical binary decision problem, using support vector machines (SVM) for classification. A three-output joint classifier is designed to manage the false predictions. A large feature set is used for classifier training and appropriate parameter selection is performed with the help of an RD complexity model. Grellert et al. [21] also used SVM for fast CU partition decisions in HEVC that uses offline training to model the classifier, while Kim et al. [22] used online learning based on a Bayesian decision rule, also for fast CU partitioning. Similarly, Zhu et al. [23] have implemented multiple binary and multi-class SVM classifier models to speed-up the CU Split decision and prediction mode selection, respectively. They also use a combination of online- and offline-trained models to achieve better classifier accuracy. Shen et al. [24] also used weighted SVM for early CU split termination in HEVC. In this case, the weights are used to reduce misclassification in the SVM model.

A data mining approach based on decision trees is used in [6] to allow the early termination of decision processes that find the best CUs, PUs, and TUs. A separate decision tree is used for each case and their fixed structure does not allow any form of adaptability to different types of content. Furthermore, using single decision trees may lead to overfitting during the training phase, while this is not a problem in RF. Ruiz et al. [25] have also used decision trees to speed up HEVC intra-coding. Bochuan Du et al. [26] used RF to speed up intra-prediction in HEVC. First, CU depth search is reduced by selecting a range using neighboring CU depth levels. Then, a RF classifier is used for Split vs. Non-Split decision. The authors have used the individual pixel values as feature vectors for the RF classifier, which may contribute to expand the feature set because of the limited information gain of individual pixels.

A common aspect of previous works using machine learning approaches is that most of them use offline training and are based on a single classifier. These offline-trained classifiers do not easily adapt to changing conditions in different types of video sequences. Therefore, feature selection becomes highly critical and this is the reason for the majority of machine learning approaches to compute a large feature set from video data. Thus, some ranking mechanism should be used to shortlist the most appropriate and effective features to improve the classifier accuracy. Although some techniques use online training or cost functions based on misclassification error to improve classifier performance, using a single classifier may result in overfitting of the classifier model. This leads to either poor performance in terms of speed-up for low-motion video or poor performance in terms of quality for high-motion video. Moreover, computation of cost functions for classifier accuracy adds to the overall computational complexity. From the related work discussed above, one can easily infer that very few techniques available in the literature use ensemble classifiers. This may be due to the common understanding that ensemble classifiers are more computationally intensive. However, ensemble classifiers have the unique advantage of diversification that can easily avoid classifier overfitting and result in a better performance [27].

In this work, we propose a machine learning approach based on RF, for reduction of HEVC computational complexity. An ensemble of offline and online classifier models is proposed, taking advantage of all the characteristics of these type of classifiers beside being quite computationally efficient [28].

3 Proposed method

As mentioned in the previous sections, testing all CU sizes at different depth levels of the CU quad-tree structure is one of the most computationally intensive tasks during HEVC encoding. Similarly, within each CU, testing of TU quad-tree structure and evaluation of all inter-prediction modes for different CU sizes at all depth levels also adds to the computational complexity. Hence, the proposed methodology targets these three tasks to reduce the computational complexity of HEVC, based on the RF ensemble classification to intelligently predict the early CU and TU depth termination rather than testing all depth levels to find the optimal one. Similarly, RF-based classifier is also used for the early prediction of SKIP mode during inter-coding. First, it predicts the SKIP mode at different CU depth levels. Second, it is used to predict the optimal TU size within each CU. Finally, the classifier is used for the prediction of early CU depth termination in the inter-frames. The classification problem is formulated as binary classification with two classes in each case. First, a set of features is identified as containing the best ones to be extracted from the video data and to train the classifier. Second, the generated classifier model is integrated in the HEVC encoder. Overall the processing pipeline of the proposed method is shown in Fig. 2. The fast-coding decisions are taken by the classifier based upon the input features extracted from video data during encoding at run time. The feature selection process is described in Sect. 3.2.

3.1 Random forests

RF is an ensemble classifier composed of multiple decision trees, as shown in Fig. 3. An ensemble classifier, by definition,

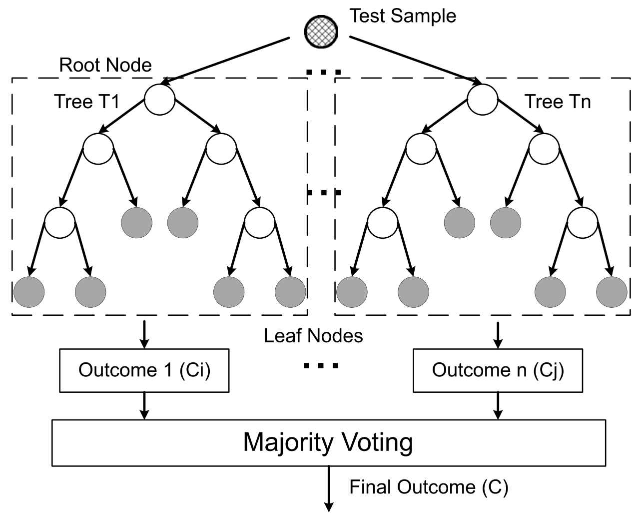


Fig. 3 Random forests (RF) with N decision trees

is a group of weak classifiers that collaborate to take a strong decision. For classification, the input test sample is fed to all decision trees, and then, each decision tree associates the sample to a specific class. Majority voting is performed on the outcome of all decision trees to classify the test sample.

The accuracy of RF depends upon the number of trees in the forest and depth of each decision tree. During the training process, these trees are generated using random samples from training data. To further split each node within the tree, a random number of features are used [7]. This randomness in tree generation, along with majority voting on the outcome of all trees in the forest, adds diversification to the RF classifier and makes it immune to overfitting. During tree generation, an entropy-based impurity measure can be computed for split decision at each node in the decision tree according to Eq. 1, where $i(X)$ is the impurity at node X, $p(w_i)$ is the fraction of samples that belongs to class w_i. The decrease in impurity at node X is measured using Eq. 2, where $i(X_L)$ and $i(X_R)$ are the impurities of left and right children nodes X_L and X_R. P_L is the fraction of training samples that will go to the left child node after split. Eq. 3 is used as a stopping criterion in decision tree generation during the training process, where β is a threshold value [29]:

$$i(X) = -\sum_{i\in C}^{n} \left(\left(p\left(w_i\right)\right) \log_2 \left(p\left(w_i\right)\right)\right) \tag{1}$$

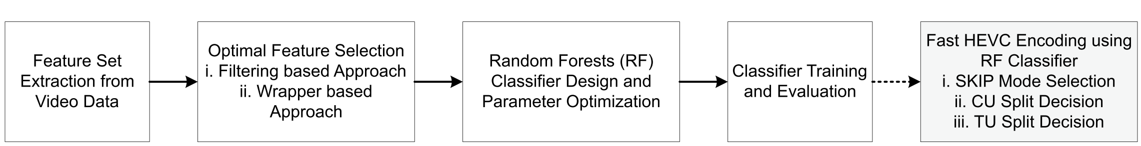


Fig. 2 Overall block diagram of the proposed methodology

$$\Delta i(X) = i(X) - P_{\mathrm{L}} i\left(X_{\mathrm{L}}\right) + \left(1 - P_{\mathrm{L}}\right) i\left(X_{\mathrm{R}}\right) \tag{2}$$

$$\Delta i(X) \leq \beta . \tag{3}$$

For training of RF and tree generation, some samples are randomly selected from a whole data set, while remaining samples are used to validate the classifier. Such process is known as out-of-bag training accuracy and this is often used to cross-validate RF classifier models.

3.2 Feature selection

Feature selection is of utmost importance in machine learning approaches due to its strong influence on accuracy. Dimensionality of the feature set is also important, because a large set may lead to overfitting and increases the computational complexity. In this work, two different approaches were evaluated for efficient feature selection. (1) Information gain filtering-based approach; (2) wrapper-based approach using RF feature importance as elimination criterion. In the filtering-based approach, individual features are ranked based on information gain to reduce the size of the feature set by only including those with higher rank. In the wrapper-based approach, the combined effect of different feature subsets is analyzed using a predefined classifier, and then, the least important ones are excluded [30].

In fast video coding, the relevant features are derived from the inherent characteristics of the video signal, coding process, mode decision, and data structures. For instance, homogeneous regions in a video sequence with smooth texture and small motion content are more often encoded as large CU blocks. The depth level of a CU is also highly correlated with spatial and temporal neighbors. TU Split decision is highly correlated with the number of non-zero coefficients and the absolute residue of a block. Similarly, SKIP inter-prediction mode is usually selected as the optimal mode for homogeneous or static regions in video frames. SKIP-mode selection is also correlated with the prediction modes of neighboring blocks [3].

The proposed methodology to select the best features comprises two steps. First, a large set of relevant features is extracted from video data to predict CU and TU depth levels and SKIP inter-prediction mode. Second, both the filtering-based approach and wrapper-based approach are used to find the most relevant features and to drop those that do not significantly contribute to the decision process. Overall, the large set includes features about texture, motion, and other parameters related to RD optimization and data structures. These include pixel variance and gradient of the current block using the Sobel operator that gives information about the texture in the block. The variance of current CU motion vectors (MV), average MV of neighboring CUs, and sum of absolute differences (SAD) with co-located blocks that give information about motion content and homogeneous regions is also used [24]. Features related to RD optimization and data structures include non-zero (NZ) coefficient count in the current block, RD cost of the current block, quantization parameter (QP), and total bits to encode the current block. These features give information about residual content in the current block. In addition, the depth of the neighboring CU blocks, partition size, average NZ coefficient count of neighbors, and count of SKIP blocks in the neighbors give information about the behavior of the neighboring blocks. To compute the features related to the neighboring CU blocks, three spatio-temporal neighboring CU blocks are considered. These include top and left CU in the current video frame and co-located CU in the adjacent frame [5].

The large feature sets defined in this work are shown in Table 1 for: fast SKIP-mode selection (15 features), early CU depth termination (18 features), and early TU depth termination (7 features). The first approach to selectively reduce the size of these large feature sets was the filtering-based approach using the information gain, which can be described as follows.

If A_i is an attribute and C is the class, a decrease in entropy of C reflects an increase in the information about C because of attribute A_i. This is known as information gain and it is defined by Eq. 4 [31]:

$$\text{Information gain} = H(C) - H\left(C|A_i\right) \tag{4}$$

where

$$H(C) = -\sum_{c \in C} \left((p(c)) \log_2 (p(c))\right) \tag{5}$$

$$H\left(C|A_i\right) = -\sum_{a \in A} p(a) \sum_{c \in C} \left((p(c|a)) \log_2 (p(c|a))\right). \tag{6}$$

Equations 5 and 6 represent the entropy of class C, before and after observing the attribute. Hence, information gain measures the importance of an individual attribute A_i and how effectively it can discriminate between different classes. To filter out the least important features, each feature A_i is assigned a score based upon its information gain and then all features are ranked according to such score.

In the wrapper-based approach, first, an RF model is trained using all features in the data set. To compute the importance of each feature, it is permuted and the classifier accuracy is computed with the resultant modified data set. A decrease in the classifier accuracy because of the permuted feature represents the importance of that feature [7]. This process is repeated to compute the importance of all the features in the data set, and then, they are ranked according to their importance.

The information gain and feature importance of each feature was computed for SKIP mode, CU, and TU Split

Table 1 Large feature set for SKIP-mode selection, CU, and TU SPLIT decision

Features	SKIP mode	CU split	TU split
Average neighbors CU depth	✓	✓	–
Average neighbors partition size	✓	✓	–
Pixel variance partial (1/4) of Current Block	✓	✓	–
Pixel average of current block	✓	✓	–
Pixel variance of current block	✓	✓	✓
MV variance of current block	–	✓	–
Average MV of neighboring blocks	✓	✓	–
Non-zero coefficient count of neighbors	✓	✓	–
NZ coefficients of current block	✓	✓	✓
SAD with co-located blocks	✓	✓	✓
RD cost of SKIP-mode encoding	–	✓	–
Neighboring CU's average RD Cost	✓	✓	–
RD cost of current block	✓	✓	✓
SKIP mode count of neighboring blocks	✓	✓	–
Gradient of current block using sobel	✓	✓	–
Total encoded bits for current block	✓	✓	✓
QP of current block	✓	✓	–
SKIP flag in current block	–	✓	–
Abs sum of coefficients of current block	–	–	✓
Prediction residual of current block	–	–	✓

decision for the following four different video sequences: 'BasketBallDrive', 'BQMall', 'BlowingBubbles', and 'Johnny'. The relevant features are extracted for 20 frames in each video sequence encoded at four different QP values, i.e., 24, 28, 32, and 38. The features are extracted for all CU layers in the CTU, i.e., from layer 0 (64×64) to layer 2 (16×16). Then, the average information gain of each feature is computed for all CU sizes. Ranking of each feature based on average information gain (filter) and average feature importance (wrapper) is shown in Table 2 for CU Split, SKIP mode, and TU Split cases. The rank order obtained for each feature based on the information gain is shown in columns "F", while the rank order obtained from wrapper is shown in columns "W". Those features with the highest information gain/feature importance are ranked as '1' in the table, while higher rank values correspond to lower information gain/feature importance.

To select the optimal reduced feature sets for CU Split, SKIP mode, and TU Split cases, multiple RF classifier models were generated. Starting with the complete feature set in each case, features were gradually eliminated based on their rank (according to information gain/feature importance shown in Table 2), and then, new RF classifier models were generated. The number of trees in these classifiers was kept constant at 20 and the accuracy was calculated for each case. The graphs of the classifier accuracy vs. number of features for filtering and wrapper approaches are shown in Figs. 4, 5 and 6 for CU Split, SKIP mode, and TU Split cases, respectively. In each graph, the number of features used in the *x*-axis corresponds to the subset defined by the same rank order in Table 2. For example, six features in the graphs of Figs. 4, 5, and 6 include those numbered from 1 to 6 in Table 2 for each case. This can also be seen as building the graph from the right to the left, by starting to compute the classifier accuracy for both the filtering and wrapper approaches using the full set of features and then removing those with lowest information gain/feature importance, one by one, till a single feature is left.

Selection of optimal features is a trade-off between the most relevant features and the feature set diversity. According to the graphs of Figs. 4, 5, and 6, as irrelevant features are removed (moving in the graph from the right to the left), classifier accuracy gradually increases up to a certain point. Then, beyond that point, it again starts decreasing because of the lack of diversity in the feature set. Similarly, when features are gradually removed, the impact on accuracy depends on their information gain/feature importance, which may not be same for all the features. For instance in Fig. 4, for filtering-based approach when feature no. 8 is removed, the classifier accuracy significantly decreases, because this feature has a relatively high feature importance, as shown in Table 2. According to the Figs. 4 and 6, the filtering-based approach reaches the maximum accuracy, higher than the wrapper-based approach, at 9 and 4 features for CU Split and TU Split, respectively. For CU SKIP mode, as shown in Fig. 5, variations in accuracy are small, i.e.,

Table 2 Reduced feature set (in Bold) for SKIP mode, CU, and TU Split decision using the filtering-based approach

Features	CU split		SKIP mode		TU split	
	F	W	F	W	F	W
Total encoded bits for current block	**1**	4	**1**	2	**2**	3
NZ coefficients of current block	**2**	2	**2**	1	**1**	7
Average neighbors CU depth	**3**	16	**5**	13	–	–
SKIP flag in current CU block	**4**	1	–	–	–	–
SKIP-mode count of neighbor blocks	**5**	6	**3**	7	–	–
Average neighbors partition size	**6**	7	**6**	8	–	–
NZ coefficient count of neighbors	**7**	14	**4**	10	–	–
MV variance of current block	**8**	3	–	–	–	–
Pixel variance of current block	**9**	5	**10**	3	**4**	4
Average MV of neighboring blocks	10	8	**7**	5	–	–
Gradient of current block using sobel operator	11	11	11	9	–	–
SAD with co-located blocks	12	12	**8**	4	5	2
Pixel variance partial (1/4) of current block	13	10	12	11	–	–
QP of current block,	14	17	**9**	12	–	–
RD cost of SKIP-mode encoding	15	13	–	–	–	–
RD cost of current block	16	9	13	6	7	6
Neighboring CU's average RD cost	17	15	15	14	–	–
Pixel average of current block	18	18	14	15	–	–
Abs sum of coefficients of current block	–	–	–	–	**3**	1
Prediction residual of current block	–	–	–	–	6	5

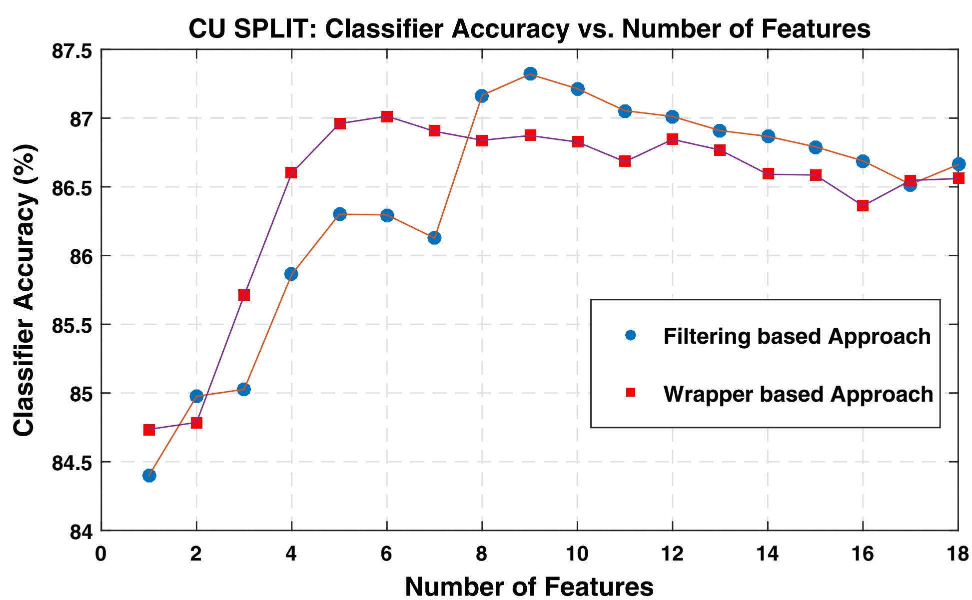


Fig. 4 CU Split decision: RF classifier accuracy vs. number of features for filtering and wrapper-based approaches

91.5%–92.3%, as features are reduced from 15 to 1. Despite being small, this range of variability should not be neglected and approximated to zero (i.e., horizontal lines in Fig. 5), as it would lead to an ambiguous solution where any single feature would be equally good. Moreover, a very small feature set may not guarantee enough diversity for all types of video contents. In Fig. 5, the filtering-based approach shows maximum accuracy at 10 features, while the wrapper-based approach shows maximum accuracy at nine features.

Based on this analysis, any of the two feature selection approaches can be used to select the optimal features in video data. In this work, we have used the filtering-based approach as it gives better accuracy out of the two. From the previous results using the filtering-based approach,

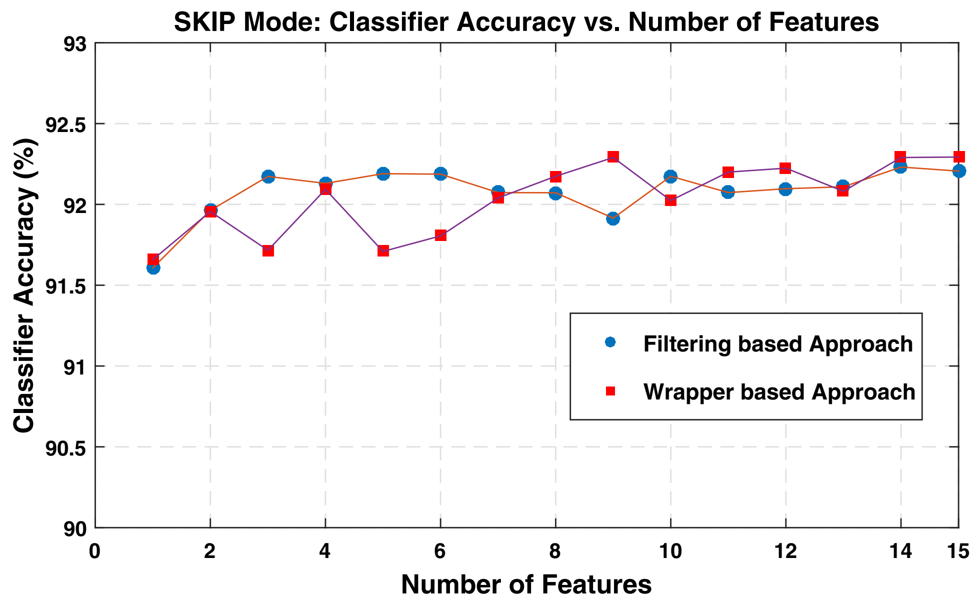


Fig. 5 SKIP-mode selection: RF classifier accuracy vs. number of features for filtering and wrapper-based approaches

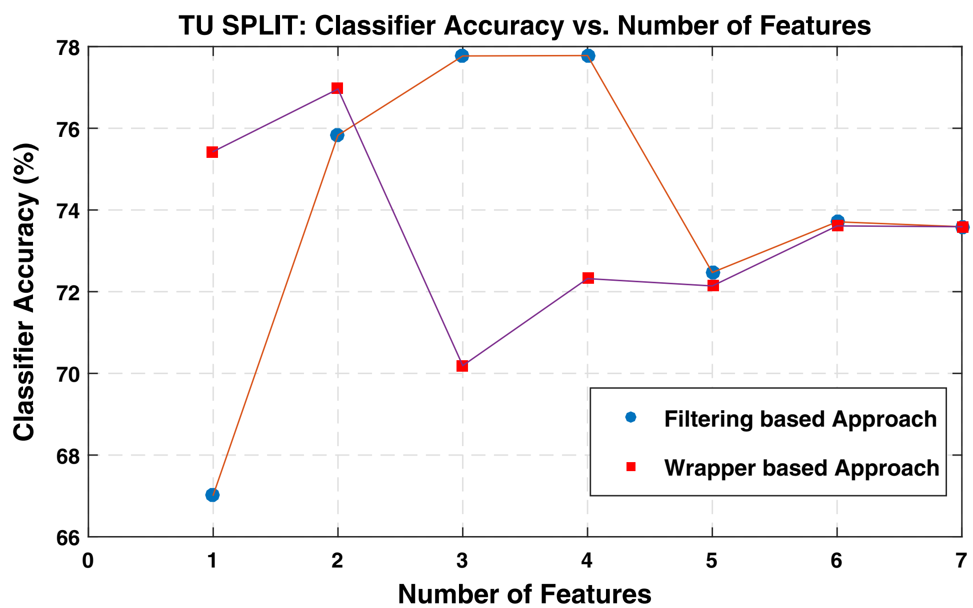


Fig. 6 TU split decision: RF classifier accuracy vs. number of features for filtering and wrapper-based approaches

nine features were found adequate to generate the classifier model for CU Split decision, ten features for SKIP mode selection, and four features for TU Split decision. The reduced feature sets for the three cases are identified in Table 2 in bold figures, which also represent the rank order of the selected features. Examples of other previous works that also followed a similar type of heuristic approach for feature selection are [24] and [32].

4 RF classifier design

The proposed method to achieve fast-coding decisions is based on separate RF classifiers trained for the SKIP-mode prediction and early termination of CU and TU depth search.

4.1 Early SKIP-mode prediction

SKIP mode in HEVC is usually chosen as the optimal prediction mode in smooth image regions with low-motion content. Moreover, SKIP mode is the most likely used mode at low bit-rates, i.e., high QP values. If SKIP mode is selected as the best mode at current CU depth level, then checking other inter-prediction modes can be skipped, avoiding further processing. The ultimate objective of the early SKIP mode prediction is to determine the optimal mode, without exhaustively evaluating all intra- and inter-prediction modes.

The proposed model for the early SKIP-mode prediction uses RF classification to speed up the selection of optimal prediction mode. The ten input features shown in Table 2 are fed to the RF classifier model at run time, where the SKIP mode classification is done before checking all other inter-prediction modes at the current depth level. The outcome of the binary classifier is either SKIP or Non-SKIP. If SKIP mode is predicted as the optimal mode, checking of all the other inter-prediction modes is skipped at current depth level.

4.2 Early CU depth termination

In a full search implementation, such as the reference software HM, the CU depth selection process starts from depth level 0 and then all depth levels (0–3) are recursively evaluated based on RD cost before the selection of the optimal CU depth level. Usually, depth level 0 is selected for smooth regions in video frame and deeper depth levels, i.e., 1, 2, and 3, are selected for regions with more texture detail and motion content. Due to the intrinsic nature of the search algorithm and data structures, large depth levels and small CU sizes result in fewer encoding bits but increased computational complexity.

The proposed method extracts the nine features shown in Table 2, from the video data and uses them as input to the RF classifier model. At each CU depth level, after evaluation of all prediction modes, the RF binary classifier takes a Split vs. Non-Split decision. If the classifier takes a Non-Split decision, all deeper CU depth levels are skipped. This results in major speed boost as all unnecessary CU depth levels are not checked.

4.3 Early TU depth termination

After prediction, a residual quad-tree is formed at the root of each CU, which recursively divides each TU into four sub-TUs starting from TU size 32×32. While an exhaustive search approach checks all possible TU sizes to find the optimal one, the proposed model takes an early TU Split decision at each TU depth level. If a Non-Split decision is taken, then checking of sub-TUs is skipped to speed-up encoding. The four features identified in Table 2 are extracted and fed to the RF binary classifier model to speed-up the selection of best depth level in each TU quad-tree.

4.4 Online vs. offline classifier models

In the proposed approach, three different configurations were investigated for training and testing the RF classifier model. In the first configuration, the RF model is trained offline and integrated in the HM Reference. In the second configuration, the RF model is trained online at run time. In this case, a few initial frames from each video stream are used for training and the remaining ones are encoded using the trained classifier model. In the third configuration, an ensemble of online and offline classifier models is used, to achieve the best possible results. The combined classifier score is computed as a weighted average of both classifier outputs. In this work, a weight of 0.5 was used for each one. For those configurations where only one classifier is used, the weight of the enabled classifier is set to 1, while that of the disabled one is set to zero. The generic diagram of such classifier is shown in Fig. 7, where the combined scores obtained from both the online and offline classifier models are used to take the CU/TU Split and SKIP-mode decisions.

4.5 Overall fast-encoding algorithm

The proposed fast-encoding decision algorithm based on ensemble classifiers for SKIP-mode prediction, TU Split, and CU Split early decisions is shown in the flowcharts of Fig. 8a, b. This algorithm is repeated during HEVC encoding for each CTU in inter-frames starting from depth level 0. The ensemble classifiers for SKIP-mode prediction and early CU depth termination run at CU level, as shown in Fig. 8a. After training the classifier models either offline or online, the main steps of the proposed fast algorithm are summarized as follows:

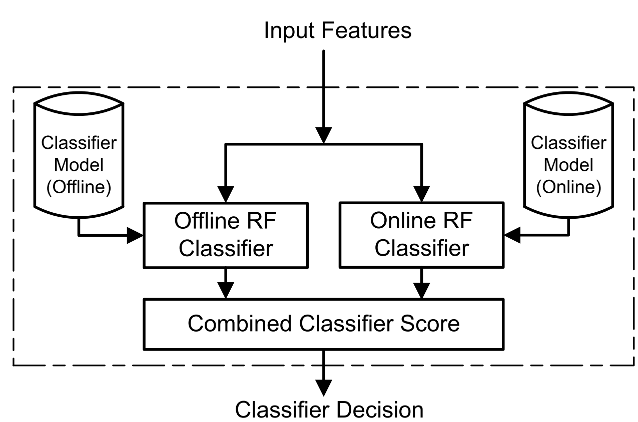


Fig. 7 RF ensemble classifier with online and offline classifier models

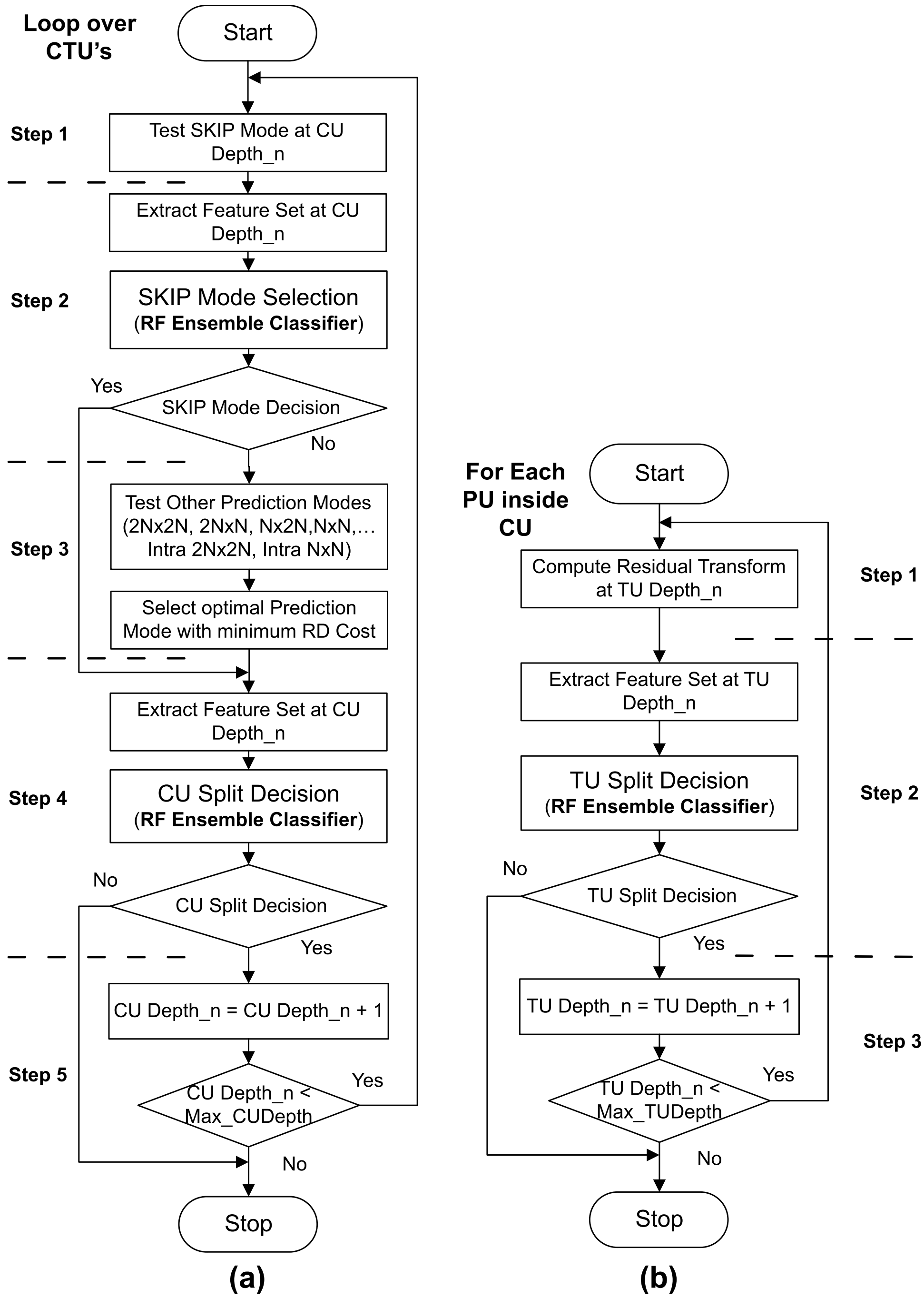


Fig. 8 Flowcharts of the overall fast-encoding algorithm: **a** early SKIP-mode selection and CU Split Decision at different depth levels in a CTU, **b** TU split decision in residual quad-tree inside a PU

Step 1 Test the SKIP prediction mode at current CU depth level and compute its RD cost.

Step 2 Compute the feature set shown in Table 2 for the early SKIP-mode selection and run the RF ensemble classifier. If the outcome is SKIP, then select SKIP as the optimal mode and go to Step 4, otherwise proceed to Step 3.

Step 3 Test all the other inter- and intra-prediction modes and possible PU sizes at current depth level and select the optimal prediction mode with minimum RD cost.

Step 4 Compute the feature set shown in Table 2 for the early CU depth termination and run RF ensemble classifier. If classifier outcome is Non-Split, stop the rate-distortion optimization process at current CU depth level and skip the next step for current CU otherwise proceed to Step 5.

Step 5 Split the current CU into four sub-CUs, increase CU depth level by 1, and repeat Steps 1 to 5 for the next CU depth level.

The ensemble classifier for the early TU depth termination runs at PU level inside every CU, as shown in Fig. 8b.

The main steps of the proposed fast algorithm are summarized as follows:

Step 1 Compute the residual transform at current TU depth level.

Step 2 Extract the feature set shown in Table 2 for early TU depth termination and run RF ensemble classifier. If classifier outcome is Non-Split, stop the rate-distortion optimization process at the current TU depth level, and skip the next step for current TU, otherwise proceed to Step 3.

Step 3 Split the current TU into four sub-TUs, increase the TU depth level by 1 and repeat Steps 1 to 3 for the next TU depth level.

The RF ensemble classifier for fast selection of SKIP mode is run for all CU depth levels from 0 to 3. On the other hand, the RF ensemble classifier for fast CU Split decision is only run at CU depth levels 0, 1, and 2, because CU is not further split at depth level 3. Similarly, the RF ensemble classifier for fast TU Split decision is only run at TU depth levels 1, 2, and 3.

The proposed ensemble classifier can also be run in offline-only mode or online-only mode by disabling the other model. In offline-only mode, the online training is disabled.

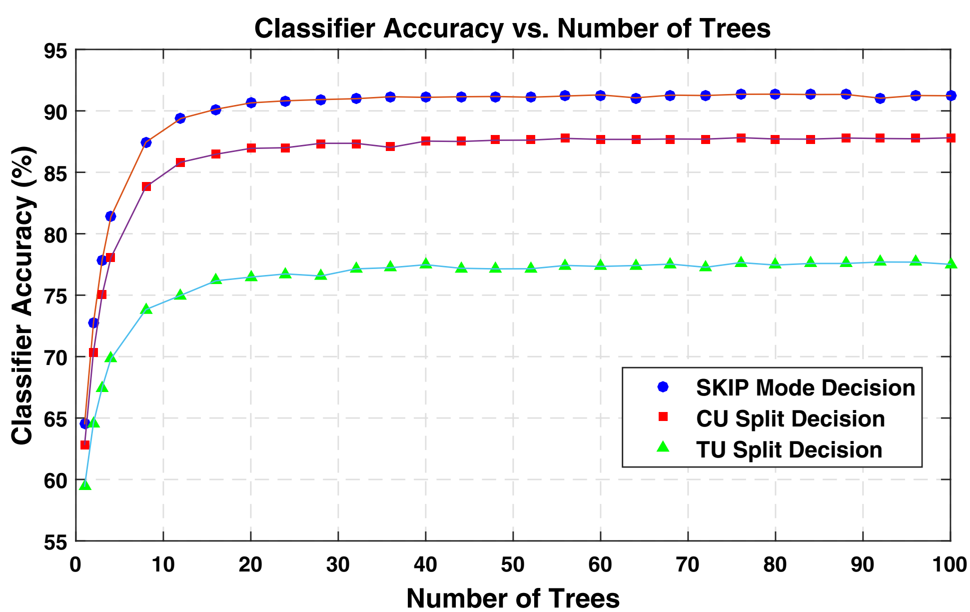
4.6 Classifier training and evaluation

Training of the RF classifier is done in both the offline and online modes to generate classifier models for SKIP-mode selection and TU and CU Split decision. For offline training model, three video sequences 'BasketBallDrill', 'Vidyo1' and 'PartyScene' are used. 'BasketBallDrill' contains high-motion content and large temporal variations. Similarly, 'PartyScene' contains scene change variations. On the other hand, 'Vidyo1' has low-motion content and static regions with homogeneous texture. Using three different sequences enables the classifier to train for different types of video content. Similarly, to avoid overlap between training and testing data and for a better validation of classifier performance, training video sequences are encoded at QP values 24, 28, 32, and 38, which are different from the QP values used during classifier testing.

For all three sequences, 20 frames were encoded and the features previously selected were extracted for training the three RF classifiers, i.e., for SKIP-mode decision, TU Split decision, and CU Split decision. To build the decision trees in the forest, samples were randomly selected from the input data with replacement. This random selection with replacement may result in replication of some samples from the original data set, while some others are left out. Samples that are left out can be used for out-of-bag training accuracy of the classifier. In the design of the RF model, different parameters such as the number of trees in the forest, depth of each decision tree, number of samples at the leaf node, and the threshold β directly affect the classifier accuracy. To get the optimal values for these parameters, multiple simulations were run for each of these parameters. In these simulations, one parameter was varied at a time while keeping the others constant and its impact on the classifier accuracy was observed. The results of these simulations are shown in Figs. 9, 10, 11 and 12. The impact of the number of trees on the classifier accuracy is shown in Fig. 9, where the number of trees is increased from 1 to 100. As observed in the figure, after 20 trees, the classifier accuracy remains constant, and hence, the optimum number of trees was defined as 20.

Similar simulations were run by varying the depth of each tree in the forest from 1 to 50, and results are shown in Fig. 10. From these simulations, it is clear that increasing the

Fig. 9 Relation of RF classifier accuracy with number of trees in the forest for SKIP mode, CU, and TU split decisions



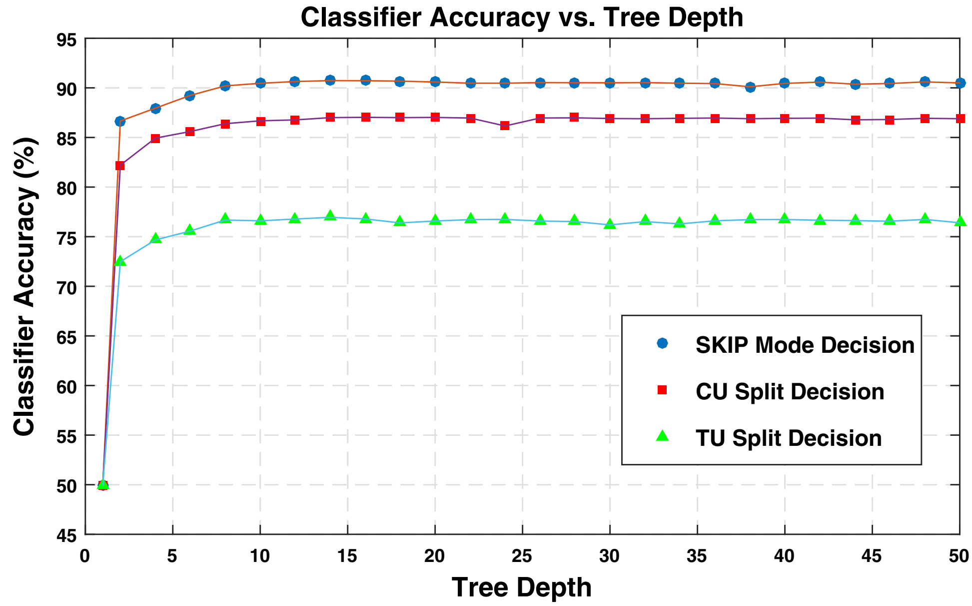


Fig. 10 Relation of RF classifier accuracy with trees depth in the forest for SKIP mode, CU and TU split decisions

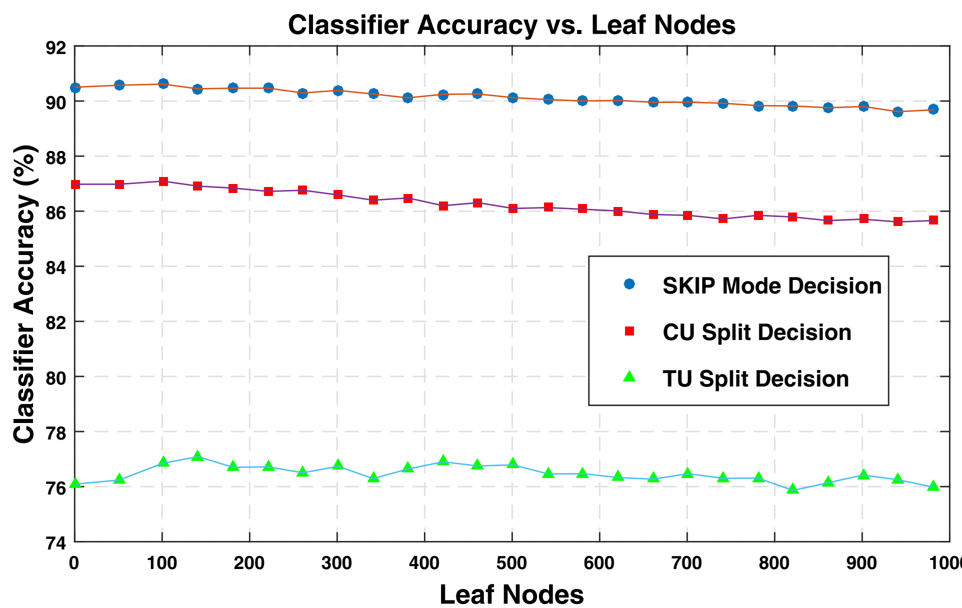


Fig. 11 Relation of RF classifier accuracy with number of leaf nodes in the forest for SKIP mode, CU, and TU split decisions

depth of the decision tree beyond 10 does not significantly affect the classifier accuracy, but increases the tree size. Hence, the depth of the decision tree was selected as 10. Similarly, based on the results shown in Fig. 11, the optimal value for number of samples at the leaf node was defined as 100. To minimize the correlation between different trees in the forest, the number of features (K) for the Split decision at each node is kept less than or equal to the total number features (N) according to Eq. $K \leq \sqrt{N}$. Similarly, to terminate the decision tree, the threshold β is used as stopping criteria at each node (Eq. 3). Since the choice of this threshold influences the tree size and also the accuracy of the classification, a simulation study was carried out to objectively evaluate its influence on the accuracy, for the three types of fast decision. Based on the results shown in Fig. 12, $\beta = 0.3$ was chosen as a good trade-off between speed and accuracy. Moreover, it was found that small variations around this value do not significantly influence the results.

Using the previous optimal values for classifier parameters, the accuracy of the classifier model was evaluated using five video sequences of different resolutions and different video contents in terms of texture and motion. Twenty frames of each sequence were used for accuracy measurement. Based on the confusion matrix, which is often used to evaluate the performance of binary classifier models, the true positive rate TPR $= \frac{\mathrm{TP}}{\mathrm{PP}}$ and the true negative rate

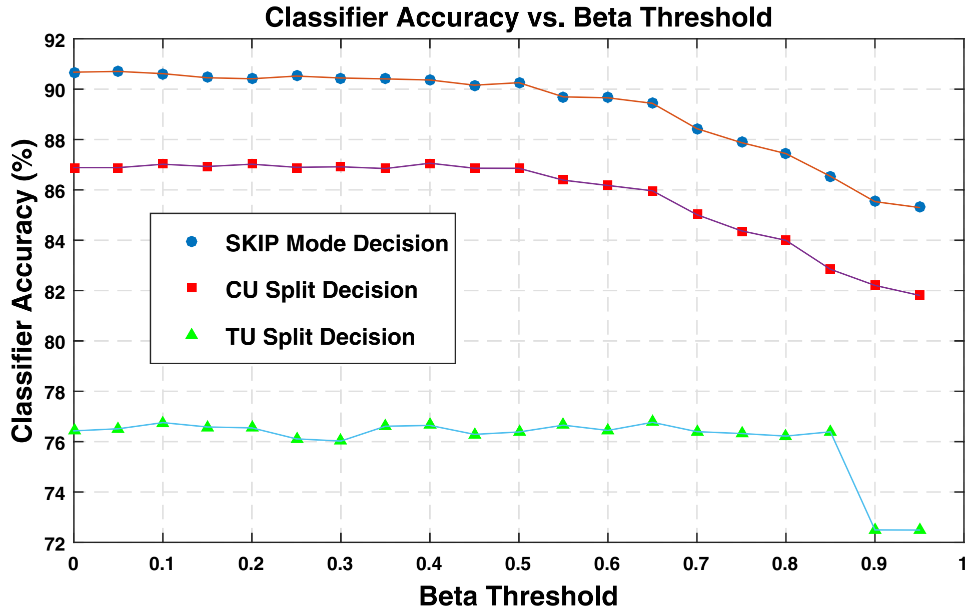


Fig. 12 Relation of RF classifier accuracy with beta threshold in the forest for SKIP Mode, CU, and TU split decisions

$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{NN}}$ were computed. The classifier accuracy was computed using Eq. 7, where (TP) represents true positive (TP), (TN) represents true negative, (FP) represents false positive, and (FN) represents false negative [33]. PP represents total positive cases and NN represents total negative cases in the data set. For SKIP-mode decision, true positive represents SKIP-mode predicted as SKIP and true negative represents Non-SKIP mode predicted as Non-SKIP. Similarly, for CU and TU Split decisions, true positive represents a Split case predicted as Split, and true negative represents Non-Split case predicted as Non-Split by the classifier.

$$\mathrm{Accuracy}(\%) = \frac{(\mathrm{TP} + \mathrm{TN}) \times 100}{\mathrm{PP} + \mathrm{NN}}. \tag{7}$$

For SKIP-mode decision, higher TNR (i.e., large number of cases correctly classified as Non-SKIP) leads to improved video quality, while higher TPR (i.e., a large number of cases correctly classified as SKIP) results in higher speed-up. Similarly, for CU and TU Split decisions, higher TPR (i.e., large number of cases correctly classified as Split) also increases the video quality, while higher TNR (large number of cases correctly classified as Non-Split) increases speed-up.

For CU SKIP-mode decision, the classifier model has an average accuracy of 90.04%, average TPR of 0.88, and average TNR of 0.89, as shown in Table 3. Specially for sequences with high-motion content, like 'PeopleOnStreet' and 'BlowingBubbles', where the percentage of SKIP blocks is small, the proposed classifier achieves small TNR values of '0.96 and 0.90', respectively, as shown in Table 3. Small

Table 3 RF classifier evaluation for SKIP-mode selection, CU, and TU split decision

Stream	Q_p	CU split decision			SKIP-mode selection			TU split decision		
		TPR	TNR	Accuracy (%)	TPR	TNR	Accuracy (%)	TPR	TNR	Accuracy (%)
BlowingBubbles	27	0.93	0.57	70.42	0.84	0.88	86.50	0.85	0.72	74.81
	37	0.84	0.81	81.84	0.76	0.90	80.83	0.73	0.92	90.59
BQMall	27	0.92	0.82	84.38	0.94	0.89	92.42	0.80	0.79	79.00
	37	0.89	0.90	90.39	0.92	0.89	91.19	0.72	0.92	90.95
Johnny	27	0.85	0.93	92.87	0.97	0.77	95.05	0.70	0.88	86.57
	37	0.79	0.98	97.48	0.97	0.72	96.74	0.62	0.98	97.77
BasketBallDrive	27	0.86	0.86	85.59	0.88	0.96	91.44	0.85	0.64	65.54
	37	0.80	0.94	93.66	0.87	0.96	88.96	0.77	0.93	92.23
PeopleOnStreet	27	0.96	0.67	77.39	0.90	0.95	93.09	0.79	0.53	62.49
	37	0.94	0.79	81.43	0.76	0.96	84.13	0.58	0.89	83.52
Average		0.88	0.83	85.55	0.88	0.89	90.04	0.74	0.82	82.35

values of TNR result in good video quality. Similarly, for sequences like 'Johnny', with static regions and low motion, the percentage of SKIP blocks is large, which results in TPR of 0.97 and good speed-up.

For CU Split decision, the classifier has an average accuracy of 85.55%, TPR of 0.88, and TNR of 0.83, as shown in Table 3. Similarly, for TU Split decision, the classifier has an average accuracy of 82.35%, TPR of 0.74, and TNR of 0.82, as shown in Table 3. For sequences with high-motion content, like 'BlowingBubbles' and 'PeopleOnStreet' where Split CUs are often necessary, and TPR of 0.93 and 0.96 results in good video quality. For sequences with low-motion content like 'Johnny', where Split CUs is rarely necessary; high TNR of 0.98 results in good speed-up.

To select the optimal number of training frames for online RF model, several simulations were run for SKIP-mode selection, CU, and TU Split decision, while increasing the number of training frames from 5 to 30. It was found that increasing the number of training frames, results in lower bitrate increase (BDBR) and lower speed-up, with all three cases following an almost linear relationship. A median value of 15 frames was used in the experiments, which is also inline with the other previous works [34].

5 Performance evaluation

To evaluate the performance of the proposed method, the RF models for early SKIP-mode selection, early TU Depth termination, and early CU depth termination were integrated in the HEVC reference software, HM version 16.10 [35]. Twenty video sequences of different resolutions and diverse content have been used. The RF library 'librf' was used for the generation of the proposed RF classifier model [36]. As previously mentioned, offline RF classifier models for SKIP-mode prediction and CU Split decision have been generated using three sequences, 'BasketBallDrill', 'PartyScene', and 'Vidyo1', using the bootstrap method. For online generation of the RF model, 15 frames of each stream were used for training, assuming that the video content characteristics do not significantly change during the limited duration of these test sequences. In longer video sequences without real-time constraints, temporal segments (i.e., video shots) should be identified and retraining should be done at shot boundaries. Simulation results were obtained for Low-Delay Main B profile and Random Access profile in HEVC for QP values of 22, 27, 32, and 37 under the common test conditions [37]. The platform used for evaluation is an Intel Core i5 machine running at 1.70 GHz with 8 GB of RAM.

Coding efficiency is measured using BDPSNR (dB) and BDBR (%) [38]. The speed-up (TS %) is measured in terms of the difference between the coding time of the HM reference encoder and the proposed RF-based fast encoding, as given by Eq. 8, where T_r and T_p are the encoding times of HM reference and proposed RF, respectively. For comparison of different approaches, only analyzing the speed-up, i.e., TS alone, does not provide an absolute and fully meaningful performance measure, because, in general, an increase in TS also increases BDBR. Hence, we have used a fusion measure (FM) that merges both BDBR and TS into a single performance indicator, defined by Eq. 9. Such indicator measures the amount of bit rate overhead per complexity reduction unit. Therefore, any fast-coding method presenting a smaller value of FM means that complexity reduction is achieved at a lower cost in terms of bit rate overhead:

$$\mathrm{TS}(\%) = \frac{(T_\mathrm{r} - T_\mathrm{p}) \times 100}{T_\mathrm{r}} \tag{8}$$

$$\mathrm{FM} = \frac{\mathrm{BDBR} \times 100}{\mathrm{TS}}. \tag{9}$$

5.1 Comparison with other state-of-the-art

To evaluate the contribution of each RF classifier, the speed-up results individually achieved by the SKIP-mode selection, CU Split and TU Split decisions, for Random Access profile, are shown in Table 4. These results were obtained using RF offline models for SKIP mode, CU Split, and TU Split decisions. The maximum contribution to speed-up is due to the early SKIP mode selection algorithm that gives an average speed-up of 47.47%. The speed-up due to early CU depth termination is 47.01%, while the early TU depth termination algorithm yields an average speed-up of 10.48%.

Table 5 shows a comparison of HEVC implementation with the offline RF model, online model, and ensemble of online and offline models for Random Access profile. It is clear from the table that, when implemented individually, both online and offline approaches yield almost the same speed-up, but the online approach produces higher BDBR that results in higher FM score of 6.63 as compared

Table 4 Individual performance achieved by SKIP-mode selection, CU, and TU split decision (RA, Offline)

	SKIP-mode selection				CU split decision				TU split decision			
	BDBR	BDPSNR	TS (%)	FM	BDBR	BDPSNR	TS (%)	FM	BDBR	BDPSNR	TS (%)	FM
Average	0.66	– 0.025	47.47	1.41	0.51	– 0.018	47.01	1.30	0.11	– 0.004	10.48	0.97

Table 5 HEVC RD performance comparison for offline, online, and ensemble classifier models (RA)

Stream	Resolution	Offline				Online				Ensemble of offline and online			
		BDBR	BDPSNR	TS (%)	FM	BDBR	BDPSNR	TS (%)	FM	BDBR	BDPSNR	TS (%)	FM
BasketballPass	416 × 240	3.91	– 0.182	45.58	8.59	3.54	– 0.164	46.45	7.62	2.44	– 0.114	41.79	5.85
BlowingBubbles		3.72	– 0.150	49.95	7.45	4.28	– 0.172	50.25	8.51	2.85	– 0.117	45.36	6.28
BQSquare		1.63	– 0.069	51.61	3.16	2.92	– 0.124	50.54	5.78	1.05	– 0.045	45.16	2.33
Flowervase		2.24	– 0.120	80.20	2.79	1.66	– 0.090	72.24	2.30	1.28	– 0.069	73.70	1.74
BasketballDrill	832 × 480	2.37	– 0.099	54.08	4.38	2.74	– 0.115	54.31	5.04	1.76	– 0.074	50.13	3.51
PartyScene		2.48	– 0.107	41.07	6.03	5.62	– 0.243	52.39	10.72	2.96	– 0.127	42.76	6.93
BQMall		2.29	– 0.085	52.44	4.36	5.25	– 0.193	56.51	9.30	3.00	– 0.113	51.09	5.87
RaceHorses		2.20	– 0.081	33.89	6.49	7.36	– 0.264	47.88	15.37	4.51	– 0.165	37.01	12.20
Johnny	1280 × 720	2.55	– 0.072	84.40	3.03	1.18	– 0.033	74.37	1.59	0.93	– 0.026	77.18	1.20
Vidyo1		1.93	– 0.085	84.73	2.28	0.86	– 0.037	73.52	1.17	0.68	– 0.030	76.40	0.89
Vidyo3		3.92	– 0.166	83.23	4.71	2.43	– 0.102	73.92	3.29	1.52	– 0.065	76.61	1.98
Vidyo4		2.03	– 0.080	82.77	2.45	0.80	– 0.032	69.77	1.14	0.60	– 0.024	73.73	0.82
FourPeople		1.71	– 0.069	83.79	2.04	1.11	– 0.046	69.83	1.59	0.77	– 0.031	70.57	1.09
KristenAndSara		2.12	– 0.079	83.25	2.55	1.03	– 0.040	73.73	1.40	0.72	– 0.027	76.63	0.94
BasketballDrive	1920 × 1080	2.41	– 0.053	56.92	4.24	4.90	– 0.105	63.32	7.74	2.81	– 0.061	56.42	4.98
BQTerrace		2.53	– 0.041	61.48	4.11	3.07	– 0.049	56.92	5.40	1.79	– 0.029	54.90	3.26
Cactus		3.26	– 0.071	61.50	5.29	4.80	– 0.101	66.86	7.18	3.22	– 0.074	61.44	5.23
Kimono1		2.48	– 0.075	59.74	4.16	5.59	– 0.167	68.26	8.19	4.38	– 0.131	62.67	6.99
PeopleOnStreet	2560 × 1600	2.57	– 0.114	33.88	7.59	11.49	– 0.497	51.75	22.20	5.55	– 0.243	40.30	13.78
Traffic		3.20	– 0.111	68.33	4.69	5.17	– 0.168	73.80	7.00	3.00	– 0.102	68.77	4.37
Average		2.58	– 0.095	62.64	4.52	3.79	– 0.137	62.33	6.63	2.29	– 0.08	59.13	4.51

to the FM score of 4.52 for offline implementation. Since the offline RF model has been trained using samples from multiple video streams encoded at different QP values, it adapts better to video sequences with diverse characteristics, such as the high-motion content of 'RaceHorses' and 'PeopleOnStreet'. This is the main reason that justifies the lower FM score obtained for these sequences when encoded with the offline model and compared with the online model. On the other hand, the online model performs better than offline, i.e., lower FM score, for video sequences with low motion and homogeneous regions like 'Johnny' and 'KristanAndSara'.

When the ensemble of both online and offline approaches is used, this results in lower BDBR. Hence, the FM score of the ensemble approach shows improved results as compared to either online or offline individual implementations. Tables 6, 7 show further results, including comparison with the other state-of-the-art implementations for Low-Delay and Random Access (Main profile), respectively. Since the works of Huang [14] and Chen [13] do not fully cover both coding configurations, their methods were implemented for the purpose of this comparison. In the case of Tai [12], the comparison is made with the results obtained by the authors.

For the case of Low-Delay Main profile, the proposed approach gives a maximum speed-up of 79.71% with an average of 54.57%, as shown in Table 6. For Random Access profile, the results are slightly better. The maximum speed-up is 84.73% with an average of 62.64%, as shown in Table 7. For video sequences with low-motion content and homogeneous regions, such as 'FourPeople', where the number of SKIP CU blocks is large and the percentage of Split cases is lower, the proposed approach achieves higher speed-up. For Low-Delay Main profile, this is 78.98% with an increase in BDBR of 1.83%, as shown in Table 6 for sequence 'FourPeople'. In comparison with the recent works, for the same video sequence, Huang [14] and Chen [13] achieve lower speed-up (35.45%) with an increase in BDBR of 0.72% and 71.59% with an increase in BDBR of 7.03%, respectively. Similarly, for the same sequence in Random Access profile, the speed-up of the proposed approach is 83.79% with an increase in BDBR of 1.71%, as shown in Table 7. Huang [14] reaches a speed-up of 36.47% with 0.36% of BDBR increase and Chen [13] gives speed-up of 62.72% with 0.69% BDBR increase for the same stream. Hence, for sequences with low-motion content, the proposed approach performs better as compared to other state-of-the-art methods, in terms of speed-up in both HEVC profiles. Similar results can be seen for the other sequences like 'Johnny' and 'Vidyo1'. This clearly shows that the proposed approach adapts better to low-motion video content and

Table 6 HEVC RD performance comparison for low-delay profile

Stream	Resolution	Proposed approach			Huang [14]			Tai [12]			Chen [13]		
		BDBR	BDPSNR	TS (%)	BDBR	BDPSNR	TS (%)	BDBR	BDPSNR	TS (%)	BDBR	BDPSNR	TS (%)
BasketballPass	416 × 240	3.12	– 0.147	38.24	2.58	– 0.120	32.93	0.60	– 0.029	32.00	2.86	– 0.133	36.21
BlowingBubbles		3.99	– 0.154	39.87	1.85	– 0.072	34.18	0.74	– 0.030	33.00	5.39	– 0.206	41.45
BQSquare		3.02	– 0.117	34.30	2.71	– 0.109	28.70	0.77	– 0.028	34.00	0.93	– 0.038	23.61
Flowervase		2.99	– 0.140	67.83	1.68	– 0.079	43.36	–	–	–	1.88	– 0.086	45.59
BasketballDrill	832 × 480	2.24	– 0.089	46.55	1.15	– 0.046	36.25	0.53	– 0.021	37.00	4.36	– 0.170	58.19
PartyScene		2.58	– 0.108	30.33	1.70	– 0.074	27.10	0.64	– 0.027	32.00	2.86	– 0.124	33.74
BQMall		1.86	– 0.071	43.32	2.08	– 0.085	33.73	0.85	– 0.033	41.00	4.55	– 0.183	45.94
RaceHorses		1.25	– 0.048	27.08	1.72	– 0.071	28.19	0.62	– 0.024	30.00	2.02	– 0.082	31.06
Johnny	1280 × 720	5.45	– 0.139	79.31	1.31	– 0.036	38.77	0.88	– 0.024	67.00	3.94	– 0.101	60.40
Vidyo1		3.53	– 0.140	79.71	0.93	– 0.040	37.98	0.56	– 0.017	61.00	6.59	– 0.270	72.51
Vidyo3		6.96	– 0.279	78.49	1.32	– 0.059	36.73	1.30	– 0.036	62.00	4.17	– 0.177	66.09
Vidyo4		3.70	– 0.130	78.29	0.99	– 0.036	35.81	0.88	– 0.021	63.00	3.96	– 0.145	65.92
FourPeople		1.83	– 0.063	78.98	0.72	– 0.027	35.45	1.00	– 0.034	63.00	7.03	– 0.238	71.59
KristenAndSara		3.30	– 0.108	77.58	1.30	– 0.042	38.11	0.75	– 0.024	63.00	3.78	– 0.121	62.12
BasketballDrive	1920 × 1080	1.93	– 0.046	49.16	0.63	– 0.014	39.87	0.75	– 0.017	41.00	3.41	– 0.077	61.65
BQTerrace		1.74	– 0.032	51.89	0.71	– 0.014	39.48	0.95	– 0.016	46.00	1.76	– 0.036	47.89
Cactus		2.46	– 0.060	53.06	1.14	– 0.028	37.99	1.07	– 0.025	43.00	3.07	– 0.073	59.94
Kimono1		2.43	– 0.079	50.49	0.73	– 0.025	39.77	0.68	– 0.022	40.00	2.23	– 0.076	47.66
PeopleOnStreet	2560 × 1600	1.85	– 0.085	27.22	1.49	– 0.069	29.38	0.96	– 0.030	50.00	3.01	– 0.139	38.71
Traffic		3.20	– 0.102	59.66	1.33	– 0.044	39.52	0.39	– 0.018	33.00	4.72	– 0.157	52.23
Average		2.97	– 0.107	54.57	1.40	– 0.05	35.66	0.79	– 0.025	45.84	3.63	– 0.13	51.13

Table 7 HEVC RD performance comparison for random access profile

Stream	Resolution	Proposed approach			Huang [14]			Tai [12]			Chen [13]		
		BDBR	BDPSNR	TS (%)	BDBR	BDPSNR	TS (%)	BDBR	BDPSNR	TS (%)	BDBR	BDPSNR	TS (%)
BasketballPass	416 × 240	3.91	− 0.182	45.58	1.78	− 0.083	32.03	0.97	− 0.047	34.00	1.03	− 0.049	38.52
BlowingBubbles		3.72	− 0.150	49.95	1.52	− 0.062	32.20	1.42	− 0.060	47.00	0.60	− 0.024	27.95
BQSquare		1.63	− 0.069	51.61	1.42	− 0.062	34.73	1.11	− 0.043	52.00	0.40	− 0.017	26.30
Flowervase		2.24	− 0.120	80.20	1.08	− 0.058	46.93	–	–	–	0.66	− 0.035	47.55
BasketballDrill	832 × 480	2.37	− 0.099	54.08	0.90	− 0.038	39.33	0.95	− 0.039	42.00	1.70	− 0.072	52.73
PartyScene		2.48	− 0.107	41.07	1.31	− 0.061	29.71	1.35	− 0.058	46.00	0.46	− 0.022	25.51
BQMall		2.29	− 0.085	52.44	2.03	− 0.083	33.62	1.59	− 0.061	47.00	1.85	− 0.076	40.93
RaceHorses		2.20	− 0.081	33.89	1.94	− 0.076	27.69	1.18	− 0.044	32.00	1.94	− 0.076	25.24
Johnny	1280 × 720	2.55	− 0.072	84.40	0.75	− 0.023	40.13	0.88	− 0.024	67.00	1.67	− 0.051	58.67
Vidyo1		1.93	− 0.085	84.73	0.43	− 0.019	38.28	1.44	− 0.046	69.00	1.18	− 0.054	66.66
Vidyo3		3.92	− 0.166	83.23	1.11	− 0.050	37.96	1.41	− 0.046	69.00	1.03	− 0.046	60.68
Vidyo4		2.03	− 0.080	82.77	0.67	− 0.026	36.83	1.26	− 0.039	68.00	1.55	− 0.061	61.51
FourPeople		1.71	− 0.069	83.79	0.36	− 0.015	36.47	1.00	− 0.034	63.00	0.69	− 0.026	62.72
KristenAndSara		2.12	− 0.079	83.25	0.52	− 0.021	39.78	0.75	− 0.024	63.00	1.18	− 0.046	59.03
BasketballDrive	1920 × 1080	2.41	− 0.053	56.92	0.94	− 0.021	39.93	1.57	− 0.034	45.00	3.07	− 0.066	55.43
BQTerrace		2.53	− 0.041	61.48	0.78	− 0.016	40.42	2.02	− 0.032	56.00	0.95	− 0.020	47.08
Cactus		3.26	− 0.071	61.50	1.33	− 0.030	38.87	1.84	− 0.039	52.00	1.42	− 0.031	55.18
Kimono1		2.48	− 0.075	59.74	0.96	− 0.031	39.72	1.25	− 0.037	47.00	1.99	− 0.064	42.89
PeopleOnStreet	2560 × 1600	2.57	− 0.114	33.88	1.44	− 0.064	28.54	1.11	− 0.052	38.00	1.40	− 0.063	33.54
Traffic		3.20	− 0.111	68.33	0.84	− 0.030	40.58	1.89	− 0.062	60.00	0.73	− 0.026	44.23
Average		2.58	− 0.095	62.64	1.11	− 0.04	36.69	1.32	− 0.043	52.47	1.27	− 0.05	46.62

gives higher speed-up with negligible increase in BDBR as compared to the other state-of-the-art, which underperform both in terms of speed-up and BDBR.

For sequences like 'PeopleOnStreet' and 'RaceHorses' that have high texture and camera movement, the speed-up of the proposed approach is slightly lower than the average. This is because these sequences are mostly encoded in small CU blocks due to very high motion, which also results in a very low percentage of SKIP blocks. Therefore, despite accurate prediction of SPLIT and Non-SKIP cases by the classifier, the overall speed-up is relatively low. For video sequences with high-motion content like 'BasketBallDrive', the proposed approach achieves a speed-up of 49.16% for Low-Delay profile at BDBR rate of 1.93% and 56.92% at BDBR rate of 2.41% for Random Access profile. For the same sequence, Huang [14] achieves a lower speed-up of 39.87% at BDBR of 0.63% for Low-Delay Profile and Tai [12] achieves 45.0% at BDBR of 1.57% for Random Access profile. Hence, the proposed approach also outperforms these recent works in terms of speed-up and BDBR for video sequences with high-motion content. Furthermore, the proposed approach based on an ensemble of offline and online classifier models reveals a better adaptation to changes in video content as compared to the other approaches that are only based on offline classifier models or use fixed thresholds for prediction.

Overall, the average speed-up of the proposed approach is higher than all the other state-of-the-art implementations under comparison in this work for both the Random Access profile and Low-Delay Main profile. This speed-up is achieved while maintaining comparative RD performance. Taking into account that 20 video sequences with quite different content characteristics and formats were used in the above experiments, generalization to other video sequences is expected to yield very similar performance. Thus, the proposed fast-encoding method is an efficient solution to ease real-time implementation of highly efficient video encoders.

5.2 Reduction in HEVC computation time and RF classifier overhead

Table 8 shows the reduction in HEVC encoding times and the computational overhead of RF classifier during fast HEVC encoding for different video streams. 'Ref.Enc Time' represents encoding time of HM reference without RF classifier. 'RF.Enc Time' represents the encoding time with RF Classifier integrated in the HM reference software. RTime1, RTime2, and RTime3 represent the reduction in HM Reference encoding time, when each of the three RF classifier for SKIP-mode selection, CU Split Decision and TU Split Decision are run one by one. OTime1, OTime2, and OTime3 represent the computational overhead time for feature extraction and classification for each of the SKIP mode selection,

Table 8 Reduction in HEVC computation time (s) and RF classifier overhead (%)

Stream	Qp	Ref.Enc Time	RTime1 SKIP	RTime2 C-Split	Rtime3 T-Split	RF.Enc Time	OTime1 SKIP	OTime2 C-Split	OTime3 T-Split	Overhead time	Overhead (%)
BlowingBubbles	27	690.41	− 188.33	− 185.08	− 55.96	402.10	1.51	0.84	8.20	10.55	2.62
	37	521.06	− 229.10	− 257.43	− 49.58	195.62	0.89	0.44	3.70	5.03	2.57
BQMall	27	2786.88	− 711.56	− 783.94	− 215.59	1467.07	4.83	3.52	32.02	46.14	2.75
	37	2246.71	− 896.03	− 1018.83	− 234.39	815.92	3.69	2.26	16.34	25.47	2.73
Johnny	27	4365.71	− 3145.84	− 2822.22	− 479.68	698.87	5.31	3.97	14.00	23.28	3.33
	37	4031.16	− 3297.01	− 2972.77	− 437.75	331.71	4.09	3.39	6.22	13.69	4.13
BasketBallDrive	27	14540.68	− 4158.18	− 6637.82	− 1295.22	6692.79	21.05	15.34	136.99	173.38	2.59
	37	11829.66	− 5741.54	− 7059.54	− 1112.38	3512.49	14.70	10.79	63.30	88.79	2.53
PeopleOnStreet	27	24190.60	− 4084.42	− 4230.83	− 1208.80	17432.04	58.56	34.30	401.96	494.82	2.84
	37	19392.21	− 5006.33	− 6449.83	− 1619.07	10817.29	41.53	29.13	224.36	295.02	2.73

CU Split Decision, and TU Split Decision, respectively. 'Total Overhead' represents the time for total computations involved in feature extraction and RF classification during fast HEVC encoding. According to the Table, the computational overhead of TU Split Classifier is higher than the other two classifiers. This is because TU Split Classifier is invoked multiple times for different prediction modes in each CU at a specific depth level. These results show that the maximum overhead of proposed classifier model is 4.13% for 'Johnny'. This sequence has homogeneous regions and low motion, which results in higher speed-up. As a result, the classifier overhead time increases as compared to the total encoding time. Overall, the percentage of classifier overhead for different sequences is far less than the reduction in computational complexity of HEVC. Hence, the proposed classifier design can be integrated in real-time video encoders with quite small computational overhead.

5.3 Final remarks

The relative gain in encoding time, given as the percentage of computational time reduction, is the most commonly used metric to evaluate and compare the results of fast video coding algorithms. Although a direct link to real-time video encoding cannot be established with such relative measure, it allows the validation and comparison of results obtained from different methods published in the literature. For instance, the other works previously published within the scope of real-time processing [9, 11, 25] also use percentage of computational time reduction to evaluate the performance of different fast video coding methods. Nevertheless, a relationship with real-time video processing can be established by measuring the real-time performance of the reference video encoder, or equivalent implementation running the same coding decision functions (source code is available online). Once the run time of the reference is known for any specific hardware platform, it is straightforward to infer the processing time expected for an implementation using the proposed fast methods. Therefore, an indirect link can always be established between the results presented in the paper and real-time performance on specific platforms.

6 Conclusion

This paper proposed a fast method for reduction of the HEVC computational complexity, based on RF ensemble of online and offline classifiers. A consistent and reproducible methodology was presented to select the best feature sets and classifier design parameters. The proposed approach models CU and TU splitting decision and SKIP-mode selection as a binary classification problem, using Random Forests for the early splitting decision and SKIP-mode prediction. The RF ensemble classifier shows good performance both for sequences with high- and low-motion contents. Particularly, the experimental results demonstrate that the proposed approach achieves increased speed-up for low- and high-motion video streams while ensuring comparative video quality as compared to the other state-of-the-art. Maximum and average speed-up achieved by the proposed approach is 84.40% and 62.64%, for Random Access, while, for Low-Delay Main, it is 78.98% and 54.57%, respectively. Taking into consideration that these gains are obtained with an average BDBR of 2.58% (Random Access) and 2.97% (Low-Delay Main), the overall results demonstrate that RF provide an efficient solution for fast video encoding that can be used in real-time implementations.

Acknowledgements The authors would like to thank the anonymous reviewers for their comments and suggestions to improve this work. Pedro A. Assuncao would like to acknowledge the support of Fundacao para a Ciencia e Tecnologia (FCT) by Instituto de Telecomunicacoes (IT), grant UID/EEA/50008/2013, and Project ARoundVision SAICT-45-2017-POCI-01-0145-FEDER-030652, PTDC/EEI-COM/30652/2017, Portugal.

References

1. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circ. Syst. Video Technol. **22**(12), 1649 (2012)
2. Correa, G., Assuncao, P., Agostini, L., Da Silva Cruz, L.A.: Performance and computational complexity assessment of high-efficiency video encoders. IEEE Trans. Circ. Syst. Video Technol. **22**(12), 1899 (2012)
3. Shen, L., Liu, Z., Zhang, X., Zhao, W., Zhang, Z.: An effective CU size decision method for HEVC encoders. IEEE Trans. Multimedia **15**(2), 465 (2013)
4. Lee, Hoyoung, Shim, Huik Jae, Park, Younghyeon, Jeon, B.: Early skip mode decision for HEVC encoder with emphasis on coding quality. IEEE Trans. Broadcast. **61**(3), 388 (2015)
5. Zhang, Y., Kwong, S., Wang, X., Yuan, H., Pan, Z., Xu, L.: Machine learning based coding unit depth decisions for flexible complexity allocation in high efficiency video coding. IEEE Trans. Image Process. **24**(7), 2225 (2015)
6. Correa, G., Assuncao, P., Agostini, L., Da Silva Cruz, L.A.: Fast HEVC encoding decisions using data mining. IEEE Trans. Circ. Syst. Video Technol. **25**(4), 660 (2015)
7. Breiman, L.: Random forests. J. Mach. Learn. **45**(1), 5 (2001)
8. Rhee, C.E., Lee, K., Kim, T., Lee, H.J.: A survey of fast mode decision algorithms for inter-prediction and their applications to high efficiency video coding. IEEE Trans. Consum. Electron. **58**(4), 1375 (2012)
9. Sun, X., Chen, X., Xu, Y., Xiao, Y., Wang, Y., Yu, D.: Fast CU size and prediction mode decision algorithm for HEVC based on direction variance. J. Real-Time Image Proc. (2017). https://doi.org/10.1007/s11554-017-0682-7
10. Shen, L., Zhang, Z., Liu, Z.: Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations. IEEE Trans. Circuits Syst. Video Technol. **24**(10), 1709 (2014)
11. Lin, T.L., Chou, C.C., Liu, Z., Tung, K.H.: HEVC early termination methods for optimal CU decision utilizing encoding

residual information. J. Real-Time Image Proc. (2016). https://doi.org/10.1007/s11554-016-0608-9
12. Tai, Kh, Hsieh, My, Chen, Mj, Chen, Cy, Yeh, C.H.: A fast HEVC encoding method using depth information of collocated CUs and RD cost characteristics of PU modes. IEEE Trans. Broadcast. **63**(4), 680 (2017)
13. Chen, F., Li, P., Peng, Z., Jiang, G., Yu, M., Shao, F.: A fast inter coding algorithm for hevc based on texture and motion quad-tree models. Signal Process. Image Commun. **47**, 271 (2016)
14. Huang, X., Zhang, Q., Zhao, X., Zhang, W., Zhang, Y., Gan, Y.: Fast inter-prediction mode decision algorithm for HEVC. Signal Image Video Process. **11**(1), 33 (2017)
15. Jaja, E.T., Omar, Z., Ab Rahman, A.AH., et al.: Enhanced inter-mode decision algorithm for HEVC/H. 265 video coding. J. Real-Time Image Proc. (2015). https://doi.org/10.1007/s11554-015-0542-2
16. Ahn, S., Lee, B., Kim, M.: A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding. IEEE Trans. Circ. Syst. Video Technol. **25**(3), 422 (2015)
17. Lee, J.H., Goswami, K., Kim, B.G., Jeong, S., Choi, J.S.: Fast encoding algorithm for high-efficiency video coding (HEVC) system based on spatio-temporal correlation. J. Real-Time Image Process. **12**(2), 407 (2016)
18. Shen, X., Yu, L., Chen, J.: Fast coding unit size selection for HEVC based on Bayesian decision rule. In: *Picture Coding Symposium (PCS), 2012* (IEEE, 2012), pp. 453–456
19. Shen, L., Zhang, Z., Zhang, X., An, P., Liu, Z.: Fast TU size decision algorithm for HEVC encoders using Bayesian theorem detection. Signal Process. Image Commun. **32**, 1–8 (2015)
20. Xiong, J., Li, H., Wu, Q., Meng, F.: A fast HEVC inter CU selection method based on pyramid motion divergence. IEEE Trans. Multimedia **16**(2), 559 (2014)
21. Grellert, M., Zatt, B., Bampi, S., da Silva Cruz, L.A.: Fast coding unit partition decision for HEVC using support vector machines. IEEE Trans. Circ. Syst. Video Technol. (2018). https://doi.org/10.1109/TCSVT.2018.2849941
22. Kim, H.S., Park, R.H.: Fast CU partitioning algorithm for HEVC using an online-learning-based bayesian decision rule. IEEE Trans. Circ. Syst. Video Technol. **26**(1), 130 (2016)
23. Zhu, L., Zhang, Y., Pan, Z., Wang, R., Kwong, S., Peng, Z.: Binary and multi-class learning based low complexity optimization for HEVC encoding. IEEE Trans. Broadcast. **63**(3), 547 (2017)
24. Shen, X., Yu, L.: CU splitting early termination based on weighted SVM. EURASIP J. Image Video Process. **2013**(4), 1 (2013)
25. Ruiz, D., Fernández-Escribano, G., Martínez, J.L., Cuenca, P.: A unified architecture for fast HEVC intra-prediction coding. J. Real-Time Image Proc. (2017). https://doi.org/10.1007/s11554-017-0685-4
26. Du, B., Siu, W.C., Yang, X.: Fast CU partition strategy for HEVC intra-frame coding using learning approach via random forests. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, APSIPA ASC 2015 (0), 1085 (2016)
27. Woźniak, M., Graña, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. Inf. Fusion **16**, 3 (2014)
28. Fern, M., Cernadas, E.: Do we need hundreds of classifiers to solve real world classification problems ? J. Mach. Learn. Res. **15**(1), 3133 (2014)
29. Duda, R.O., Hart, P.E.P.E., Stork, D.G.: Pattern Classification. Wiley, Oxford (2001)
30. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res. **5**, 1205 (2004)
31. Hall, M.A., Holmes, G.: Benchmarking attribute selection techniques for discrete class data mining. IEEE Trans. Knowl. Data Eng. **15**(6), 1437 (2003)
32. Zhu, L., Zhang, Y., Li, N., Jiang, G., Kwong, S.: Machine learning based fast h.264/avc to hevc transcoding exploiting block partition similarity. J. Vis. Commun. Image Rep. **38**, 824 (2016)
33. Fawcett, T.: Tom: an introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861 (2006)
34. Mallikarachchi, T., Talagala, D.S., Arachchi, H.K., Fernando, A.: Content-adaptive feature-based CU size prediction for fast low-delay video encoding in HEVC. IEEE Trans. Circ. Syst. Video Technol. **8215**(c), 1 (2016)
35. High Efficiency Video Coding (HEVC) | JCT-VC. https://hevc.hhi.fraunhofer.de/. Accessed 2 Feb 2019
36. Lee, B.N.: librf: C++ random forests library. http://mtv.ece.ucsb.edu/benlee/librf.html. Accessed 2 Feb 2019
37. Bossen, F.: Common test conditions and software reference configurations. in *12th Meeting of JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11* (Geneva)
38. Bjontegaard, G.: Calculation of average PSNR differences between RD-curves. In *ITU - T SG16 Q. 6 VCEG-M33* (Austin, Texas)

Muhammad Tahir received the M.S. degree in computer engineering in 2005 from Lahore University of Management Sciences (LUMS), and the B.S. degree in electrical engineering in 2000 from University of Engineering and Technology (UET), Lahore, Pakistan. He is now working towards the Ph.D. degree from the Department of Electrical Engineering at Capital University of Science and Technology (CUST), Islamabad, Pakistan. His research interests include video coding, multimedia systems, and algorithm development for parallel and multi-core processor architectures.

Imtiaz A. Taj received the M.Sc. and Ph.D. degrees in electronics and information engineering from Hokkaido University, Japan, in March 2001. Currently, he is Professor in the Department of Electrical Engineering, Capital University of Science and Technology (CUST), Islamabad, Pakistan. He is also heading Vision and Pattern Recognition research group at CUST. His research interests include computer vision, image processing, video processing, pattern recognition, biometrics, and optics.

Pedro A. Assuncao received the Licenciado and M.Sc. degrees in electrical engineering from the University of Coimbra, Portugal, in 1988 and 1993, respectively, and the Ph.D. in electronic systems engineering from the University of Essex, UK, in 1998. He is currently a Professor of Electrical Engineering and Multimedia Communication Systems at Polytechnic Institute of Leiria and senior researcher at Instituto de Telecommunicacoes, Portugal. He is author/co-author of about an hundred fifty papers in conferences and journals, two books, twelve book chapters, and four US patents. His research interests include 2D/3D, light field and omnidirectional video coding, complexity control, multimedia error concealment, and quality evaluation.

Muhammad Asif received the Ph.D degree in electrical engineering from Capital University of Science and Technology (CUST), Islamabad, Pakistan, in 2016. His research interests include image and video processing, computer vision, pattern recognition, biometrics, computer networks and security, parallel processing, and embedded system optimization. He has published more than ten research papers in international reputed journals/conferences.