Journal of Real-Time Image Processing (2020) 17:185–196
https://doi.org/10.1007/s11554-019-00936-0

SPECIAL ISSUE PAPER

Fast CU partition-based machine learning approach for reducing HEVC complexity

Soulef Bouaafia[1] · Randa Khemiri[1] · Fatma Ezahra Sayadi[1] · Mohamed Atri[1,2]

Received: 31 August 2019 / Accepted: 2 December 2019 / Published online: 9 December 2019

Abstract
With the development of video coding technology, the high efficiency video coding (HEVC) provides better coding efficiency compared to its predecessors H.264/AVC. HEVC improves rate distortion (RD) performance significantly with increased encoding complexity. Due to the adoption of a large variety of coding unit (CU) sizes, at RD optimization level, the quadtree partition of the CU consumes a large proportion of the encoding complexity. Hence, the computational complexity cost remains a critical issue that must be properly considered in the optimization task. In this paper, two machine learning-based fast CU partition method for inter-mode HEVC are proposed, to optimize the complexity allocation at CU level. First, we propose an online support vector machine (SVM)-based fast CU algorithm for reducing HEVC complexity. The later was trained in an online way. Second, a deep convolutional neural network (CNN) is designed to predict the CU partition, in which large-scale training database including substantial CU partition data is considered. Experimental results demonstrate that the proposed online SVM can achieve a time saving of 52.28% with a degradation of 1.928% in the bitrate (BR). However, the proposed deep CNN can reduce the encoding time by 53.99% with 0.195% BR degradation. Compared to the state-of-the art, the two proposed approaches outperform the related works in terms of both RD performance and complexity reduction at inter-mode.

Keywords HEVC · Deep CNN · Online SVM · Complexity reduction

1 Introduction

Over the last decade, with the significant deployment of new technology, such as the Internet of Things (IoT) in smart city and smart industry, video data has become the largest source of data consumed globally. Due to the rapid growth of video applications (video surveillance, 3D videos, mobile video, smart city and industry video traffic) and the increasing demand for superior quality video services, video data volume has become a challenge for transmission and multimedia storage. According to the high-quality requirements of video applications, the ultra high definition (UHD) video (4 K and 8 K) increases the traffic load explosively. In this context, the new generation of video coding standard 'High Efficiency Video Coding' (HEVC) standardized by the Joint Collaborative Team on Video Coding (JCT-VC) in 2013, was created to address these challenges [1]. HEVC saves approximately 50% of bitrate (BR) for the same subjective video quality, with respect to its predecessor H.264/advanced video coding (AVC) standard. However, this unmatched performance is achieved by increasing the encoder computational complexity mainly due to its block partition structure [2]. The quadtree structure of the CU is the most critical part in terms of HEVC coding complexity, since it consists of an exhaustive search for the best rate distortion optimization (RDO) partition. Consequently, the HEVC complexity becomes one of the most important tasks requiring more advanced techniques to provide the optimal performance in terms of rate distortion (RD) performance and complexity reduction. As shown in Fig. 1, the greatest complexity lies in the selection of the optimal prediction mode, especially in the inter-mode [3].

To adequately address the CU mode decision issue in video coding, the existing works on fast mode decision

✉ Soulef Bouaafia
soulefbouaafia@gmail.com

[1] Electronics and Microelectronics Laboratory, Faculty of Sciences of Monastir, University of Monastir, Environment Street, 5019 Monastir, Tunisia

[2] College of Computer Science, King Khalid University, Abha, Saudi Arabia

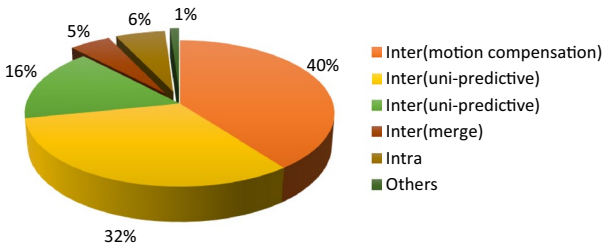


Fig. 1 HEVC profile

algorithms can be divided into two categories, including statistical approaches and machine learning-based schemes. Several statistical approaches have contributed in different ways to reduce the HEVC complexity in terms of both inter and intra coding [3–6]. For example, authors in Ref. [3] proposed a mode decision algorithm based on a look-ahead stage to simplify the CU partition process at inter-mode. To alleviate the computational complexity of HEVC, in Ref. [5], the authors proposed a fast algorithm to split CU based on pyramid motion divergence at inter prediction. In addition, a fast early CU-splitting and pruning method using Bayesian decision rule with low complexity and full RD cost was developed in Ref. [6]. Although the above statistical methods have considerably enhanced the coding efficiency, they give an insufficient performance in some special cases because of statistical thresholds.

On the other hand, machine learning category has witnessed great success in many disciplines, especially in image and video compression. However, mode decision problem can be transformed into a classification problem, and learning algorithms were then explored when classifying modes in video coding. In this context, machine learning approaches have been adopted to predict the CU partition toward HEVC complexity reduction. A CU depth algorithm composed of multiple binary classifiers based on SVM with different parameters was proposed in Ref. [15] to predict the splitting of the CU partition. To reduce the encoding complexity, a fuzzy SVM-based fast CU decision method was proposed to reduce the HEVC complexity [16]. For the HEVC intra coding, Liu et al. [17] applied a hardware CNN to reduce the maximum intra coding complexity. In view of this complexity at HEVC inter-mode, a neural network-based inter prediction algorithm was proposed in Ref. [23]. However, these approaches are shallow, with limited learning capacity, which makes them insufficient to accurately model the complicated CU partition process. Based on machine learning, this paper proposes a fast CU partition algorithm to reduce both HEVC complexity and RD performance. We propose an online SVM-based fast CU partition, which reduces the HEVC complexity at inter-mode. Next, this paper develops a deep convolutional neural network (CNN) structure to predict the CU partition at HEVC inter-mode. To learn our deep CNN model, a large-scale database is built on the inter-mode CU partition. The proposed method can achieve a good trade-off between complexity reduction and RD performance. Our main contributions are summarized in:

1. We propose an online SVM-based fast CU algorithm for reducing HEVC complexity.
2. We design a deep CNN architecture to predict the CU partition of HEVC at inter-mode.
3. We construct a large-scale database for CU partition of the inter-mode HEVC, to accurately train the deep CNN that aims to reduce the HEVC complexity.

The remaining of this paper is organized as follows. Section 2 presents the review of related works. Section 3 explains the overview of the CU partition in HEVC. In Sect. 4, we propose a machine learning approach to reduce HEVC complexity at inter-mode. The experimental results are shown in Sect. 5. We conclude the paper in Sect. 6.

2 Related works

The HEVC complexity reduction has always been a popular challenge in the video coding field. According to this complexity, many features have been adopted to simplify the RDO search of the CU partition, which classified into statistical and machine learning methods. In statistical methods, several fast decision algorithms have been introduced in Refs. [3–11]. To reduce the HEVC computational complexity, Gabriel et al. [3] introduced a look-ahead stage-based fast partitioning and mode decision algorithm. Wang et al. [4] proposed a threshold-based splitting decision scheme with respect to the RD cost of each CU. It reduces the number of available intra candidates, adaptive reference frame selection and early termination of coding unit splitting. In Ref. [5], authors proposed a fast algorithm to split CU based on pyramid motion divergence at inter prediction. In addition, a fast early CU-splitting and pruning method with low complexity and full RD cost was developed by Cho et al. [6]. In a similar way, a fast coding unit based on Bayesian rules to minimize the RD cost was proposed, as Shen et al. [7]. Furthermore, authors in Ref. [8] proposed an adaptive CU depth decision approach, which exploits both the existence of non-zero coefficients after the transform, and the maximum depth of temporally co-located CTUs. Also based on the spatial and temporal homogeneity of the images, some authors perform an analysis of the input pictures, such as Fernandez et al. [9] and Lee et al. [10], who proposed an early termination algorithm. With regard to inter

prediction, the square-type-first mode decision algorithm was proposed to decrease the encoding time [11]. These methods are based on the statistics on the RD cost properties, temporal and spatial correlation, which limit their applicability and may be difficult to handle the situations with various contents, complex coding structures.

The past few years have exhibited great success in applying machine learning tools to enhance the video coding. In this vein, great efforts have been carried out to integrate machine learning tools to predict the CU partition to reduce HEVC complexity [12–23]. The search for the optimal partitioning has also been considered as a classification problem. For example, Corrêa et al. [12] proposed data mining techniques-based three early termination schemes to simplify the decision on the optimal CTU structures. In a similar way, an SVM-based fast CU partition decision is proposed in Ref. [13]. To reduce the encoding complexity, Zhang et al. [14] propose a CU early termination algorithm. In this work, the authors designed a CU depth decision process in HEVC and model it as a three-level of hierarchical classification decision. In this regard, an SVM-based fast HEVC encoding algorithm was proposed by Zhu et al. [15] to predict both the CU partition and PU mode. The CU early termination is modeled as hierarchical binary classifications, whereas the PU selection is decided as a multi-class classification. To reduce the HEVC encoding complexity, Zhu et al. [16] proposed a CU decision method based on fuzzy SVM that achieve a good trade-off between computational complexity reduction and RD performance. In recent studies, learning-based techniques have also been applied to fast CU partitioning of intra-mode HEVC, such as [17], which implements a CNN algorithm along with its VLSI design, and [18], which uses logistic regression classification-based fast HEVC intra mode decision. To improve the HEVC complexity, Amer et al. [20] proposed a fully connected neural networks and Laplacian transparent composite models. Most recently, deep learning techniques have also been employed to speed up the encoding process and to predict the CU partition [21–23].

The analysis of these earlier works shows that it is possible to achieve a significant HEVC complexity reduction by using learning-based solutions. More sophisticated techniques such as CNNs should then be able to yield competitive results, particularly with regard to CU prediction.

While studying the methodology of the machine learning-based approaches [12–23], we noticed that some aspects could still be improved, for example in the training process. In contrast, the main motivation of this paper is to focus our efforts between SVM and CNN, in which the large-scale database for the CU partition of HEVC at inter-mode is considered.

3 Overview CU partition

The main contributions of the HEVC standard are the block partition structure that significantly improves compression performance [24]. First, the picture is partitioned into several coding tree unit (CTU) of size 64×64. This CTU replaces the macroblock in the previous standard. The hierarchical coding structure of the HEVC varies between the largest coding unit (LCU), having a size of 64×64, and the smallest coding unit (SCU) of size 8×8. A CU can be 64×64, 32×32, 16×16 or 8×8, corresponding to four CU depths, 0, 1, 2 and 3. A quadtree partition can be used to represent the hierarchical partition of CTU into CU [25]. Moreover, CUs are split into prediction unit (PU) and transform unit (TU). With our knowledge, the depth choice in each CTU goes through a decision process the RD cost calculation-based of each CU partition inside the CTU. An example of the HEVC quadtree concept is shown in Fig. 2.

In the CTU with 64×64 size, the split flag is set to 0, the RD cost RD1 is calculated. Then the sub-CUs of 32×32 size are obtained when the split flag changes to 1. The first one, $CU_1 0$ has an RD cost equal to RD2. The next depth is reached where the CU is partitioned into four CUs of size 16×16. The first CU ($CU_2 0$) of size 16×16 has an RD cost equal to RD3. When its split flag is 1, the last depth (depth = 3) is reached and it is therefore partitioned into four SCU of size 8×8. The RD cost for each SCU will be noted RD4, RD5, RD6 and RD7, respectively. The first decision will be taken from the bottom to the top by determining if the first CU of size 16×16 is checked or not. We need a comparison of the sum of the four RD cost of the SCU 8×8 with the RD3 of the CU 16×16 to make a decision. If the RD3 is greater than the sum of RD4, RD5, RD6 and RD7, the partitioning decision of $CU_2 0$ will be taken, otherwise $CU_2 0$ will not be split. Alike for the other CUs, the decision is always based on the Eq. (1). Generally, RDO is a method to decide the optimal mode in video coding. The determination of optimal CU modes is obtained via the minimum RD cost. In a 64×64 CTU, 85 possible CUs are selected:

$$\mathrm{RD}_{\mathrm{cost}_{\mathrm{CU}}} < \sum_{k=0}^{3} \mathrm{RD}_{\mathrm{cost}_{\mathrm{subCU}}}(k). \tag{1}$$

4 Proposed work

In this section, we introduce the proposed learning approach-based fast CU partition. Firstly, an online SVM-based fast CU algorithm for reducing HEVC complexity is introduced. Secondly, we design the deep CNN-based network architecture to predict the CU partition structure

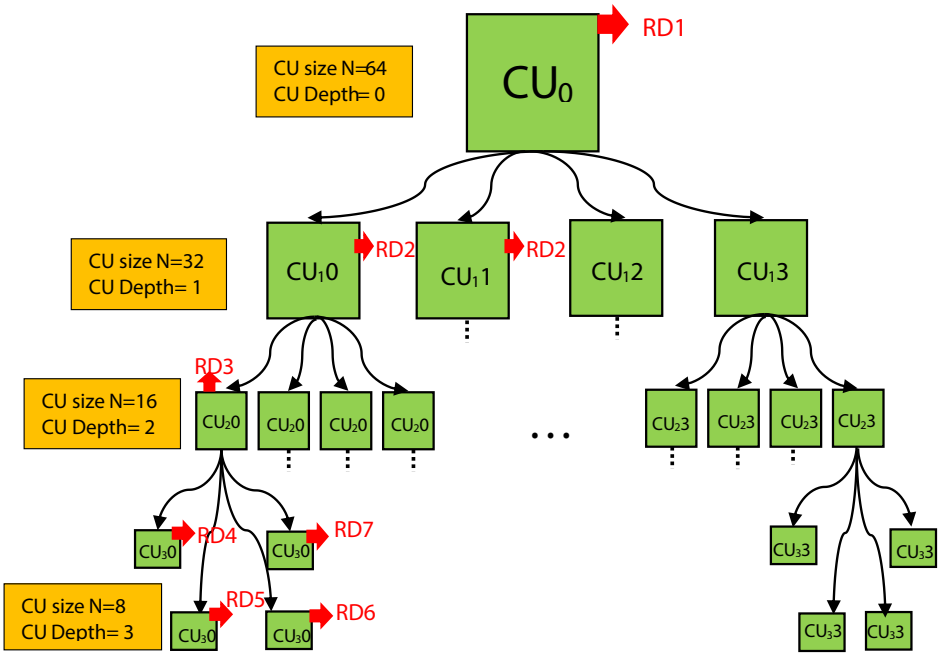


Fig. 2 CU partition structure in HEVC

at each depth from 0 (64 × 64) to 3 (16 × 16). Finally, before results analysis, we introduce the training phase of our deep CNN, in which a large-scale database modeled on the encoding information obtained from HEVC standard.

4.1 Three-level CU classifier

In the HEVC standard, the CTU supports quadtree CU partitions with four levels of CU depth from 0 to 3, which corresponds to CU size from 64 × 64 to 8 × 8. The CU partition can be considered as a combination of binary classifiers $\{F_i\}_1^3$ at three levels of decisions $l \in \{1, 2, 3\}$ on whether to split a parent CU into sub-CUs. For example, at level 1, the CU with size 64 × 64 is split into 32 × 32 CUs. Next, $l = 2$ means the level of decision for 32 × 32 into 16 × 16, and $l = 3$ stands for 16 × 16 into 8 × 8 as shown in Fig. 2. According to the CTU, we assume that the CUs are denoted as CU, CU_i, $\text{CU}_{i,j}$ corresponding to depth 0, 1, 2, 3, where $i, j \in \{0, 1, 2, 3\}$ are the index of sub-CUs. In each CU depth, we need to determine whether to split the current CU or not. The total number of splitting patterns for CU is 83,522. There are too many types of CU partitions and it is hard to be solved by a single multi-class classification in one step. However, due to the large number of patterns combinations, the prediction is adopted at each decision level to yield $\tilde{F}_1(\text{CU})$, $\{\tilde{F}_2(\text{CU}_i)\}_{i=0}^3$ and $\{\tilde{F}_3(\text{CU}_{i,j})\}_{i,j=0}^3$, which denotes the predicted $F_1(\text{CU})$, $\{F_2(\text{CU}_i)\}_{i=0}^3$ and $\{F_3(\text{CU}_{i,j})\}_{i,j=0}^3$, respectively.

4.2 Online support vector machine (SVM)

In machine learning theory, SVM is a supervised learning tool that performs classification and regression analysis [26]. A hyperplane technique is used in SVM to separate the data from one dimension to high dimensional space.

The SVM can transform the data to the high dimensional space through nonlinear transformation, if the data points are clearly not linearly separable in the input space. To separate the two classes of data points, SVM maps the sample data into a hyperspace. In addition, the main goal of SVM is to solve linear and nonlinear problems to find an optimal hyperplane. SVM classifier creates a hyperplane to maximize the margin between the hyperplanes and the support vectors [27]. The support vector classifier principal used in this work is shown in Fig. 3.

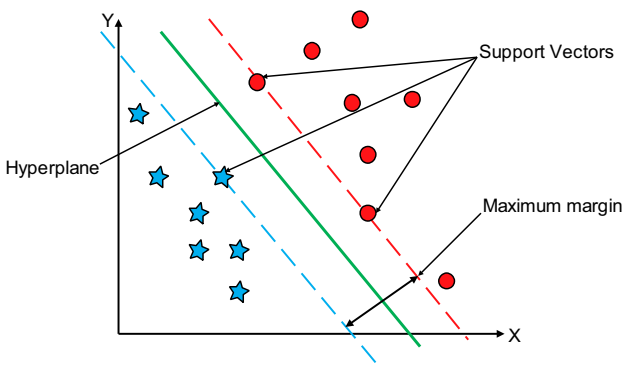


Fig. 3 Example of support vector classifier

The CU split decision can be modeled as a binary classification problem, with classes split and non-split. Here, we propose an online SVM as a machine learning technique, since it is robust and popular in solving the binary classification problem with significant computational advantages. The main idea is to find a hyperplane that can separate the training samples of different classes while maximizing the margin between these classes to determine the CU splitting level. According to Eq. (2), the ideal weight vector w is a linear combination of support vectors. Therefore, the support vectors are the training points that minimize the misclassification. Given training set with N samples, $\{x_i, y_i\}_{i=1}^{N}$, $x_i \in R^n$ while $y_i \in \{-1, 1\}$, the hyperplane parameterized by the normal vector w that maximizes margins can be found by solving the optimization problem:

$$\min_{w} \frac{\gamma}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y(w \cdot x)), \tag{2}$$

where $\gamma \geq 0$ is the smoothing parameter and is defined by: $\gamma = \frac{1}{nC}$, where C is the parameter which need to be tuned during SVM training.

Mathematically, support vector machines (SVMs) handle such situations by using a kernel function which maps the data to a different space where a linear hyperplane can be used to separate classes. In this work, Gaussian radial basis function (RBF) is applied as the kernel function, which is defined as:

$$K(x_i, x_j) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right). \tag{3}$$

To reduce the HEVC complexity, we online train our SVM model to early terminate the CU splitting process. In the online training mode, 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 on demand. Figure 4 illustrates an example of an online training mode, where training frames are in yellow color and the predicting frames are in blue color. The advantage of online training is the properties of the video sequence of the training and testing are quite close. It is better for improving the prediction accuracy.

4.3 Deep CNN architecture

Convolutional neural network is the most widely used deep learning model for video processing applications. According to the mechanism of the CU partition at inter-mode HEVC, a deep CNN structure is shown in Fig. 5.

The residual CTU is fed into CNN architecture. Here, the residue is obtained by pre-coding the frame in HEVC. Our proposed architecture is composed of pre-convolution, convolution layers, concatenated vector and fully connected layers. The pre-convolution layers are residual CUs of CU, CU_i or $CU_{i,j}$, corresponding to the three levels. Therefore, the residual block is subtracted by the mean intensity values to reduce the variation of the input CTU samples. Specifically, at the first level of CU partition, the mean value of CU is removed in accordance with the output of $\tilde{F}_1(\text{CU})$. At the second level, four CUs $\{CU_i\}_0^3$ are subtracted by their corresponding mean values, matching the 2×2 output of $\{\tilde{F}_2(CU_i)\}_{i=0}^3$. At the third level, $\{CU_{i,j}\}_{i,j=0}^3$ remove the mean values in each CU for the 4×4 output $\{\tilde{F}_3(CU_{i,j})\}_{i,j=0}^3$.

After pre-convolution task, the three convolutional layers are used to extract features from data at all levels. The convolution layer is a mathematical operation that takes two inputs such as CU partition and a filters. In each layer, the convolution kernels of all three levels have the same size. In our work, at the first convolutional layer, 16 kernels are used to extract the low features maps for the CU partition. Following the first convolutional layer, the feature maps are convoluted twice with 2×2 kernels to generate features at a higher level. The strides of all the above convolutions are equal to the sizes of the corresponding kernels for non-overlap convolution.

The above design of the convolutional layer is in accordance with all possible non-overlap CUs at different sizes for

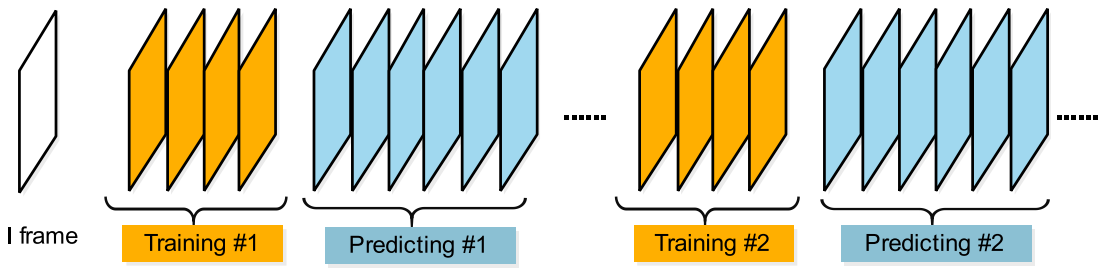


Fig. 4 Online training mode

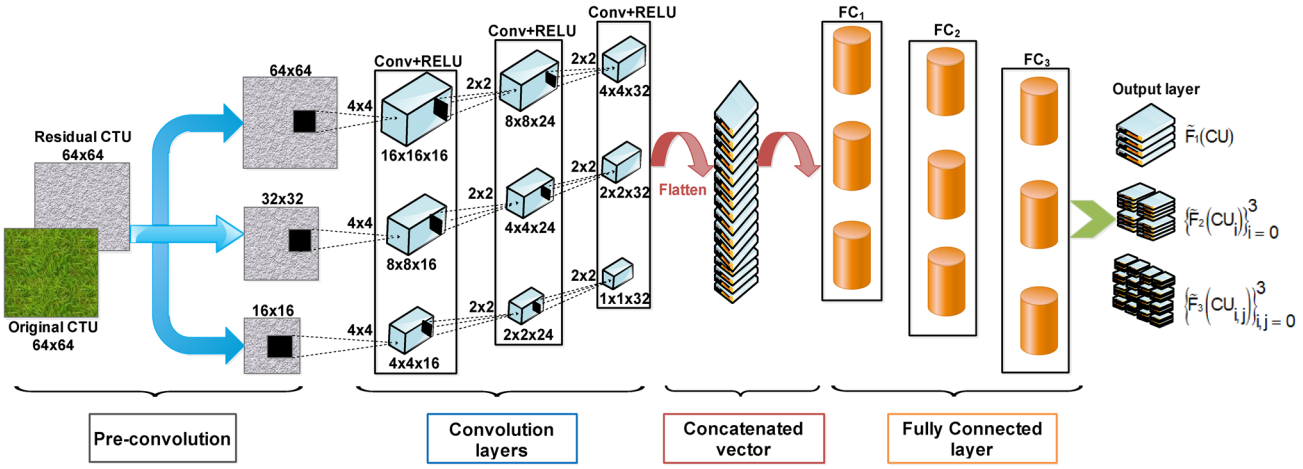


Fig. 5 Deep CNN architecture

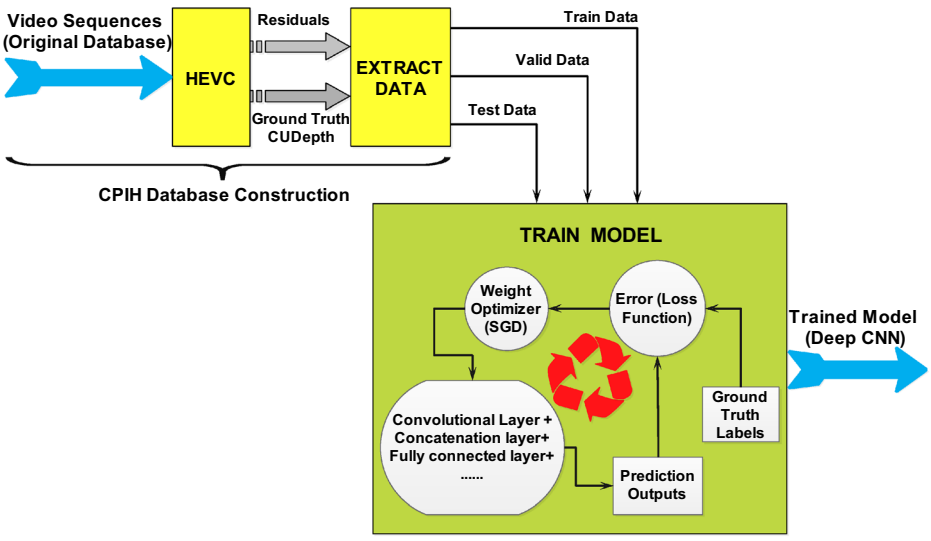
CTU partition. At the end of the convolution, through the concatenation layer, the final feature maps are concatenated together and then flatten into a vector. In the following fully connected layers, features generated from the whole CTU are all considered to predict the CU partition at each single level.

Finally, the concatenated vector flows through three fully connected layers as illustrated in Fig. 5, including two hidden layers and one output layer. The two hidden fully connected layers successively generate feature vectors denoted by $(\mathrm{FC}_l)_{l=1}^{3}$. The outputs of deep CNN are 1, 4, and 16 elements, such as the predicted binary labels $\tilde{F}_1(\mathrm{CU})$ in 1×1, $\{\tilde{F}_2(\mathrm{CU}_i)\}_{i=0}^{3}$ in 2×2 and $\{\tilde{F}_3(\mathrm{CU}_{i,j})\}_{i,j=0}^{3}$ in 4×4 at three levels, respectively. In deep CNN structure, the early termination may result in the calculation of the fully connected layers at levels 2 and 3 being skipped, thus saving computation time. Specifically, if CU is decided not to be split at level 1, the calculation of $\{\tilde{F}_2(\mathrm{CU}_i)\}_{i=0}^{3}$ is terminated early at level 2. At level 3, the $\{\tilde{F}_3(\mathrm{CU}_{i,j})\}_{i,j=0}^{3}$ does not need to be computed for the early termination, if $\{\mathrm{CU}_i\}_0^3$ are not all split. The function rectified linear units (ReLU) is used to activate all convolutional layers and hidden fully connected layers, since ReLU has better convergence speed [28]. Moreover, since all the labels for splitting or non-splitting are binary, all the output layers in three levels are activated with the sigmoid function.

4.4 Training phase

In this section, we present the training process for the proposed deep CNN as shown in Fig. 6. We train our model in a supervised learning manner, in which the deep CNN has been learned based on labeled data. In this context, we create

Fig. 6 Training process



the database for training the proposed model, which satisfy highly performances (high accuracy, low loss).

We establish a large-scale database for CU partition of the inter-mode HEVC (CPIH), to increase the prediction accuracy. However, to construct our CPIH database, we selected 114 raw video sequences with different resolutions (from 352×240 to 2560×1600) [29–32]. These sequences are gathered into three sub-sets: 86 sequences for training, 10 sequences for validation, and 18 sequences for test. Table 1 summarizes the chosen videos and the number of frames (41,349) in our CPIH database.

First, we encode the original database (114 video sequences) by HEVC (original HEVC encoder) common test condition at different quantization parameters (QP = 22, 27, 32, 37) using low delay P configuration (using encoder_lowdelay_P_main.cfg) to obtain the residue and the ground truth CU depth. The ground truth CU depth files contain the division probability of the entire sequences.

Second, to construct a training sample, the train, valid, and the test data are generated by implementing the 'EXTRACT DATA program'. The training data is used to train the model as before, where the validation data is used to determine when to stop the learning process. For the test data, 18 sequences of classes A–E from the Joint Collaborative Team on Video Coding (JCT-VC) are used to evaluate the performance of the proposed deep CNN [31].

As shown in Fig. 6, the TRAIN MODEL process summarizes the manner on how to train the model based on the CPIH database construction. The stochastic gradient descent algorithm with momentum (SGD) is used as a powerful optimization algorithm to update the network weights at each iteration and minimize gradient error between the ground truth labels and the prediction outputs. This process will continue until the loss function reaches a minimum value

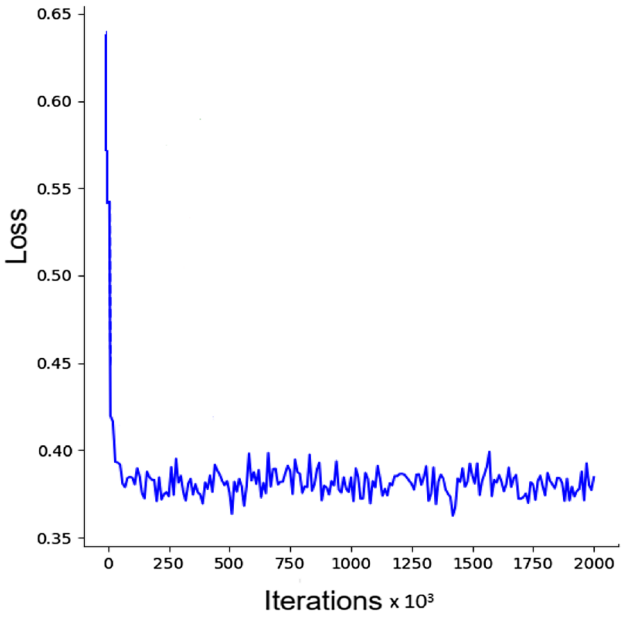


Fig. 7 Training loss

(Fig. 7). Furthermore, the deep CNN model is trained at four QPs by using different sizes of CU, which varies from 16×16 to 64×64.

For the learning of our deep CNN model, we assume that the cross entropy is applied as the loss function following Eqs. (4) and (5):

$$L = \frac{1}{N}\sum_{n}^{N} L_n, \tag{4}$$

where N is the number of training samples and L_n is the sum of the cross entropy:

Table 1 Sequences in CPIH database

Resolutions	Num. of videos	Num. of frames	Train data		Valid data		Test data	
			Num. of videos	Num. of frames	Num. of videos	Num. of frames	Num. of videos	Num. of frames
352×240 (SIF)	4	677	4	677	_	_	_	_
352×288 (CIF)	25	7080	23	6530	2	550	_	_
704×576 (4CIF)	5	2880	4	2280	1	600	_	_
720×486 (NTSC)	7	2100	6	1800	1	300	_	_
416×240 (240p)	4	1900	_	_	_	_	4	1900
832×480 (480p)	4	1900	_	_	_	_	4	1900
1280×720 (720p)	10	4227	5	1327	2	1100	3	1800
1920×1080 (1080p)	35	11,037	28	8417	2	540	5	2080
2048×1080 (2K)	18	9248	16	8048	2	1200	_	_
2560×1600 (WQXGA)	2	300	_	_	_	_	2	300
Average	114	41,349	86	29,079	10	4290	18	7980

$$L_n = Y(F_1^n(\mathrm{CU}), \tilde{F}_1^n(\mathrm{CU})) + \sum_{i\in\{0,1,2,3\}} Y(F_2^n(\mathrm{CU}_i), \tilde{F}_2^n(\mathrm{CU}_i)) + \sum_{i,j\in\{0,1,2,3\}} Y(F_3^n(\mathrm{CU}_{i,j}), \tilde{F}_3^n(\mathrm{CU}_{i,j})), \quad (5)$$

where Y denotes the cross entropy between the ground truth labels and the predicted labels. The labels predicted by our deep CNN are represented by $\{\tilde{F}_1^n(\mathrm{CU}), \{\tilde{F}_2^n(\mathrm{CU}_i)\}_{i=0}^3$ and $\{\tilde{F}_3^n(\mathrm{CU}_{i,j})\}_{i,j=0}^3\}_{n=0}^N$.

We use the Tensorflow-GPU deep learning framework to train our proposed deep CNN on an NVIDIA GeForce GTX 480 GPU that can dramatically improve speed during training compared to the CPU. We adopt a batch mode learning method with a batch size of 64 where the momentum of the stochastic gradient descent algorithm optimization is set to 0.9. To train our deep CNN, the base learning rate is set to decay exponentially to 0.01, changing every 1000 iterations. The total number of iterations was 2,000,000. Finally, the trained model (deep CNN) can be used to predict the CU partition (classes) at HEVC inter-mode.

5 Experimental results

In this section, we evaluate the performance of the proposed approaches. All experimental results are implemented in the HEVC reference software HM16.5 using random access (RA) and low delay P (LDP) configurations. In this regard, the QP values tested were 22, 27, 32 and 37 for encoding process. At inter-mode, our experiment was carried out using 18 video sequences of the JCT-VC standard test set [32], which include the following resolutions: 2560×1600 (A), 1920×1080 (B), 832×480 (C), 416×240 (D), and 1280×720 (E). Simulation results were conducted on windows 10 OS platform with Intel®core TM i7-3770 @ 3.4 GHz CPU and 16 GB RAM.

5.1 Performance metric

To evaluate our proposed algorithms, we use the most crucial performance metric of fast encoding, denoted the computational time saving (T), as shown in Eq. (6):

$$\Delta T = \frac{T_\mathrm{P} - T_\mathrm{o}}{T_\mathrm{o}} \times 100(\%), \quad (6)$$

where T_o is the computational time of the original HM, and T_p is the computational time of our proposed algorithm-based fast CU encoding.

Additionally, the RD performance is the critical metric for evaluation. We use the peak signal-to-noise ratio (PSNR) for objective video quality measurement and the BR compared to the original HM, which are defined as follows in Eqs. (7) and (8):

$$\Delta\mathrm{PSNR} = \mathrm{PSNRP} - \mathrm{PSNRo}\,(\mathrm{dB}), \quad (7)$$

$$\Delta\mathrm{BR} = \frac{\mathrm{BR_P} - \mathrm{BR_o}}{\mathrm{BR_o}} \times 100\,(\%). \quad (8)$$

5.2 Performance evaluation of our online SVM

Table 2 summarizes the performance comparison between our proposed scheme and the original HEVC under LDP and RA configurations, respectively.

According to simulation results, our proposed online SVM helps reduce HEVC complexity and improve the RD performance significantly. As it can be observed, the RA configuration performs better results in terms of coding efficiency and time reduction on average compared to LDP configuration. With regard at computational complexity, our scheme allows a maximum time saving of 62.21% with an average of 53.14% using random access configuration. Also, it achieves on average more than 1.269% in terms of BR with almost negligible decrease in PSNR. While by using LDP configuration, our approach saves 52.28% in execution time with a loss of 1.928% in the BR.

These results confirm the robustness of our algorithm in reducing the complexity and coding efficiency of inter-mode HEVC. This refers to the optimized method of finding the best partition with the optimal RD in a significant time compared to the standard method which calculates all the partitions of the CU.

For more evaluation, Fig. 8 shows the HEVC complexity reduction and RD performance at different video classes under RA and LDP configurations.

The BR of our proposed method is averagely better for the RA configuration than for the LDP configuration. On the other hand, the proposed scheme performs much better in the LDP configuration in terms of PSNR compared to the RA configuration. As shown in Fig. 8, at the RA configuration, our approach is able to reduce the encoding time at all video sequences compared to the LDP configuration.

5.3 Performance evaluation with online SVM and deep CNN

Table 3 gives a comparison of our two proposed methods; deep CNN and online SVM, in terms of complexity reduction and RD performance using LDP configuration.

The experimental results show that our deep CNN model achieves a significant complexity reduction in the range of 53.99% with 0.195% BR compared to the online SVM at LDP configuration. On the other hand, our online SVM

Table 2 Performance analysis of our fast online SVM

Class	Sequence	Random access			Low delay P		
		ΔBR (%)	ΔPSNR (dB)	Δ*T* (%)	ΔBR (%)	ΔPSNR (dB)	Δ*T* (%)
A	PeopleOnStreet	1.765	−0.0685	−56.15	2.235	−0.055	−56.56
	Traffic	0.937	−0.0355	−58.23	2.022	−0.0478	−58.07
B	Kimono	0.750	−0.027	−54.22	0.449	−0.020	−44.18
	ParkScene	0.840	−0.0297	−54.37	0.790	−0.023	−52.60
	Cactus	0.757	−0.024	−48.63	0.717	−0.019	−41.38
	BQTerrace	0.294	−0.016	−58.52	0.328	−0.022	−41.45
	BasketballDrive	0.761	−0.0277	−55.07	0.583	−0.023	−51.17
C	BasketballDrill	1.207	−0.0475	−57.38	1.440	−0.047	−55.87
	BQMall	2.357	−0.0833	−62.21	3.313	−0.058	−55.96
	PartyScene	1.275	−0.0543	−54.14	2.415	−0.062	−52.93
	RaceHorses	2.202	−0.0945	−55.90	2.650	−0.053	−51.25
D	BasketballPass	1.745	−0.076	−53.05	3.167	−0.067	−55.49
	BQSquare	1.305	−0.0465	−57.03	3.692	−0.086	−55.92
	BlowingBubbles	1.509	−0.06	−56.55	2.207	−0.067	−51.87
	RaceHorses	3.293	−0.1488	−55.75	4.605	−0.091	−50.81
E	FourPeople	0.366	−0.0188	−56.13	1.017	−0.014	−51.90
	Johnny	0.793	−0.0263	−58.84	1.729	−0.036	−60.12
	KristenAndSara	0.673	−0.023	−57.75	1.357	−0.028	−53.57
Average		1.269	−0.050	−53.14	1.928	−0.045	−52.28

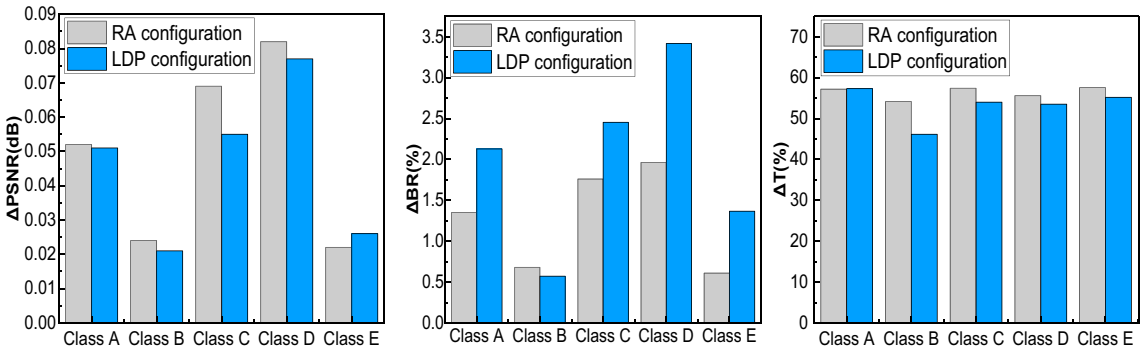


Fig. 8 Complexity reduction and RD performance at different video classes under RA and LDP configurations

demonstrates significant coding losses in BR of 1.928% and an increase on average of 52.28% in time reduction. As it can be seen, our proposed deep CNN obtains significantly best results in terms of execution time for the sequence of class E, this is caused by the low motion activities displayed in these sequences, which leads to larger partitions. For the same reason, it is possible to observe a slightly higher encoding time for high-resolution sequences compared with those of lower resolution.

In summary, from the overall performance evaluation we can find that the proposed method deep CNN outperforms the online SVM in terms of both complexity reduction and RD performance of inter-mode HEVC, as seen in Table 3. This implies that the proposed deep CNN is robust in reducing complexity of inter-mode HEVC when compared to the online SVM. This refers that the CNN works well with visual images recognition whereas SVM is used widely in classification problems. Also, it is difficult to parallelize SVM but the CNN architecture inherently supports parallelization.

For more evaluation, the complexity reduction and the RD performance of deep CNN versus online SVM at the LDP configuration is shown in Fig. 9.

As shown in Fig. 9, the deep CNN performances exceed in terms of BR those of online SVM for all test sequences. However, online SVM is superior to deep CNN in terms of PSNR. With regard to complexity reduction, for classes B and E, deep CNN is more efficient than online SVM.

Table 3 Performances comparison between deep CNN and online SVM

Class	Sequence	Deep CNN			Online SVM		
		ΔBR (%)	ΔPSNR (dB)	Δ*T* (%)	ΔBR (%)	ΔPSNR (dB)	Δ*T* (%)
A	PeopleOnStreet	0.572	−0.030	−50.67	2.235	−0.055	−56.56
	Traffic	−0.571	−0.061	−57.90	2.022	−0.0478	−58.07
B	Kimono	0.728	−0.025	−43.26	0.449	−0.020	−44.18
	ParkScene	−0.401	−0.052	−64.14	0.790	−0.023	−52.60
	Cactus	0.412	−0.052	−52.57	0.717	−0.019	−41.38
	BQTerrace	−2.606	−0.071	−58.43	0.328	−0.022	−41.45
	BasketballDrive	1.130	−0.031	−51.30	0.583	−0.023	−51.17
C	BasketballDrill	−0.044	−0.056	−53.54	1.440	−0.047	−55.87
	BQMall	1.019	−0.051	−52.25	3.313	−0.058	−55.96
	PartyScene	−0.709	−0.085	−51.54	2.415	−0.062	−52.93
	RaceHorses	0.529	−0.040	−42.22	2.650	−0.053	−51.25
D	BasketballPass	0.690	−0.054	−52.42	3.167	−0.067	−55.49
	BQSquare	−2.647	−0.162	−52.79	3.692	−0.086	−55.92
	BlowingBubbles	−0.327	−0.071	−46.55	2.207	−0.067	−51.87
	RaceHorses	0.754	−0.032	−38.01	4.605	−0.091	−50.81
E	FourPeople	−0.659	−0.062	−67.54	1.017	−0.014	−51.90
	Johnny	−0.361	−0.124	−69.66	1.729	−0.036	−60.12
	KristenAndSara	−1.026	−0.070	−67.20	1.357	−0.028	−53.57
Average		−0.195	−0.063	−53.99	1.928	−0.045	−52.28

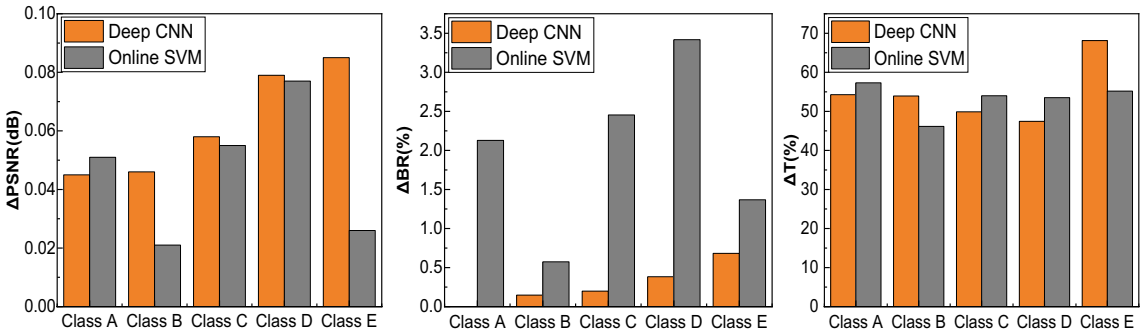


Fig. 9 Complexity reduction and RD performance of deep CNN versus online SVM under LDP configuration

Table 4 Results of our deep CNN model compared with two state-of-the-art methods

Class	Deep CNN			[14]			[19]		
	ΔBR (%)	ΔPSNR (dB)	Δ*T* (%)	ΔBR (%)	ΔPSNR (dB)	Δ*T* (%)	ΔBR (%)	ΔPSNR (dB)	Δ*T* (%)
Class A	0.0005	−0.045	−54.28	1.766	−0.059	−48.92	5.637	−0.179	−47.94
Class B	−0.147	−0.046	−53.94	1.952	−0.047	−46.52	4.026	−0.093	−48.36
Class C	0.198	−0.058	−49.88	1.647	−0.058	−39.50	3.408	−0.123	−38.69
Class D	−0.382	−0.079	−47.44	1.194	−0.045	−29.09	2.166	−0.083	−30.15
Class E	−0.682	−0.085	−68.13	2.571	−0.065	−68.15	3.796	−0.097	−57.09
Average	−0.195	−0.063	−53.99	1.826	−0.055	−46.44	3.806	−0.115	−44.44

5.4 Comparison with the state-of-the-art

In this section, the obtained results are compared to the other state-of-the-art approaches. Table 4 gives the comparison of the proposed algorithm with the other state-of-the-art methods.

With regard to complexity reduction, we note that ΔT results are averaged over four QPs {22, 27, 32 and 37} at each class. As seen in this table, ΔT of our proposed method achieves 53.99% on average, which is superior to 46.44% obtained by Zhang et al. [14], and 44.44% obtained by Mallikarachchi et al. [19]. Therefore, our scheme achieves a largest complexity reduction at inter-mode HEVC than the other two approaches.

Additionally to the complexity reduction, the RD performance is considered as a critical metric for evaluation. Table 4 lists the results of the ΔPSNR and ΔBR for evaluating RD performance. As shown in this table, the ΔPSNR of our deep CNN method averages −0.063 dB, which is better than −0.115 dB of [14]. On the other hand, the ΔBR of our method is averagely −0.195%, which outperforms 1.826% of [14] and 3.806% of [19].

Consequently, our proposed algorithm outperforms other state-of-the-art approaches [14] and [19] in terms of both time reduction and RD performance.

6 Conclusion

In this paper, we proposed a fast CU partition based on machine learning approaches to reduce the HEVC complexity of inter-mode. An online SVM-based fast CU partition method was proposed for reducing the encoding complexity of HEVC. Then, to predict the CU partition of HEVC, a deep CNN was proposed, which reduces the HEVC complexity at inter-mode. In experiment results, the online SVM reduces the execution time by 52.28% on average with an increase in the BR of 1.928%. However, deep CNN model improves the RD performance with 0.195% in the BR saving and archives on average 53.99% of time saving under LDP configuration. Consequently, our deep CNN scheme performs better trade-off between RD performance and complexity reduction compared to online SVM. The comparative results demonstrate that the proposed deep CNN proves its effectiveness in reducing the HEVC complexity.

References

1. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
2. Khemiri, R., Kibeya, H., Sayadi, F.E., Bahri, N., Atri, M., Masmoudi, N.: Optimization of HEVC motion estimation exploiting SAD and SSD GPU-based implementation. IET Image Proc. **12**(2), 243–253 (2017)
3. Cebrián-Márquez, G., Martinez, J.L., Cuenca, P.: Adaptive inter CU partitioning based on a look-ahead stage for HEVC. Signal Process. Image Commun. **76**, 97–108 (2019)
4. Wang, S., Luo, F., Ma, S., Zhang, X., Wang, S., Zhao, D., Gao, W.: Low complexity encoder optimization for HEVC. J. Vis. Commun. Image Represent. **35**, 120–131 (2016)
5. Xiong, J., Li, H., Wu, Q., Meng, F.: A fast HEVC inter CU selection method based on pyramid motion divergence. IEEE Trans. Multimed. **16**(2), 559–564 (2014)
6. Cho, S., Kim, M.: Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding. IEEE Trans. Circuits Syst. Video Technol. **23**(9), 1555–1564 (2013)
7. Shen, X., Yu, L., Chen, J.: Fast coding unit size selection for HEVC based on Bayesian decision rule. In: Proceedings of Picture Coding Symposium, pp. 453–456 (2012)
8. Li, Y., Yang, G., Zhu, Y., Ding, X., Sun, X.: Adaptive inter CU depth decision for HEVC using optimal selection model and encoding parameters. IEEE Trans. Broadcast. **63**(3), 535–546 (2017)
9. Fernández, D.G., Del Barrio, A.A., Botella, G., Garcia, C.: Fast and effective CU size decision based on spatial and temporal homogeneity detection. Multimed. Tools Appl. **77**(5), 5907–5927 (2018)
10. 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), 407–418 (2016)
11. Ahn, Y.J., Sim, D.: Square-type-first inter-CU tree search algorithm for acceleration of HEVC encoder. J. Real Time Image Process. **12**(2), 419–432 (2016)
12. Corrêa, G., Assuncao, P.A., Agostini, L.V., da Silva Cruz, L.A.: Fast HEVC encoding decisions using data mining. IEEE Trans. Circuits Syst. Video Technol. **25**(4), 660–673 (2015)
13. Grellert, M., Zatt, B., Bampi, S., da Silva Cruz, L.A.: Fast coding unit partition decision for HEVC using support vector machines. IEEE Trans. Circuits Syst. Video Technol. **29**, 1741–1753 (2018)
14. 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), 2225–2238 (2015)
15. 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), 547–561 (2017)
16. Zhu, L., Zhang, Y., Kwong, S., Wang, X., Zhao, T.: Fuzzy SVM-based coding unit decision in HEVC. IEEE Trans. Broadcast. **64**(3), 681–694 (2017)
17. Liu, Z., Yu, X., Gao, Y., Chen, S., Ji, X., Wang, D.: CU partition mode decision for HEVC hardwired intra encoder using convolution neural network. IEEE Trans. Image Process. **25**(11), 5088–5103 (2016)
18. Hu, Q., Shi, Z., Zhang, X., Gao, Z.: Fast HEVC intra mode decision based on logistic regression classification. In: Proceedings of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1–4 (2016)
19. 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. Circuits Syst. Video Technol. **28**(3), 693–705 (2018)
20. Amer, H., Rashwan, A., Yang, E.H.: Fully connected network for HEVC CU split decision equipped with Laplacian transparent

composite model. In: Picture Coding Symposium (PCS), pp. 189–193 (2018)
21. Laude, T., Ostermann, J.: Deep learning-based intra prediction mode decision for HEVC. In: Proceedings of Picture Coding Symposium (PCS), pp. 1–5 (2016)
22. Li, T., Xu, M., Deng, X.: A deep convolutional neural network approach for complexity reduction on intra-mode HEVC. In: Proceedings of IEEE International Conference on Multimedia and Expo (ICME), pp. 1255–1260 (2017)
23. Wang, Y., Fan, X., Jia, C., Zhao, D., Gao, W.: Neural network based inter prediction for HEVC. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2018)
24. Khemiri, R., Bahri, N., Belghith, F., Bouaafia, S., Sayadi, F.E., Atri, M., Masmoudi, N.: Fast Motion Estimation's Configuration Using Diamond Pattern and ECU, CFM, and ESD, Modes for Reducing HEVC Computational Complexity, pp. 1–17. IntechOpen, "Digital Imaging" Book, London (2019)
25. Khemiri, R., Kibeya, H., Loukil, H., Sayadi, F.E., Atri, M., Masmoudi, N.: Real-time motion estimation diamond search algorithm for the new high efficiency video coding on FPGA. Analog Integr. Circuits Signal Process. **94**, 259–276 (2018)
26. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
27. Huang, S., Cai, N., Pacheco, P.P., Narrandes, S., Wang, Y., Xu, W.: Applications of support vector machine (SVM) learning in cancer genomics. Cancer Genom. Proteom. **15**(1), 41–51 (2018)
28. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of IEEE International Workshop on Acoustic Signal Enhancement (IWAENC), pp. 315–323 (2011)
29. Xu, M., Deng, X., Li, S., Wang, Z.: Region-of-interest based conversational HEVC coding with hierarchical perception model of face. IEEE J. Sel. Top. Signal Process. **8**(3), 475–489 (2014)
30. Ohm, J.R., Sullivan, G.J., Schwarz, H., Tan, T.K., Wiegand, T.: Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC). IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1669–1684 (2012)
31. Xiph.org.: Xiph.org Video Test Media. [Online]. https://media.xiph.org/video/derf (2017). Accessed 15 June 2019
32. Bossen, F.: Common test conditions and software reference configurations. Document JCTVC-L1100, Joint Collaborative Team on Video Coding, (2013)

Soulef Bouaafia received her Master degree in Micro and Nano-electronics in 2018 from the Faculty of Science of Monastir. She is currently a Ph.D. student in the Laboratory of Electronics and Micro-Electronics (LEμE), University of Monastir. Her current research interests include image and video processing, HEVC, Post-HEVC standards, on embedded systems.

Randa Khemiri received her Ph.D. in Micro-Electronics from the Faculty of Science of Monastir (FSM), in 2017. In 2013, she obtained her Master degree in Micro and Nano-electronics from the Faculty of Science of Monastir. She is a member of Electronics and Micro-Electronics Laboratory (LEμE). Her current research interests include image and video processing, on embedded systems (FPGA, GPU), algorithms implementation on parallel architectures.

Fatma Ezahra Sayadi received her Ph.D. Degree in Micro-Electronics from Faculty of Science of Monastir, Tunisia in collaboration with the LESTER Laboratory, University of South Brittany Lorient France, in 2006 and her HDR in 2018. She is currently a member of the Laboratory of Electronics and Micro-electronics. Her research includes image and video processing in graphics processor, motion tracking and pattern recognition, circuit and system design.

Mohamed Atri received his Ph.D. Degree in Micro-Electronics from Faculty of Science of Monastir, Tunisia, in 2001 and his HDR in 2011. He is currently a Full Professor at the College of Computer Science, King Khalid University, Abha (KSA). His research includes circuit and system design, pattern recognition, image and video processing.