Hindawi
Scientific Programming
Volume 2021, Article ID 6628041, 10 pages
https://doi.org/10.1155/2021/6628041

Research Article

CNN-LSTM Learning Approach-Based Complexity Reduction for High-Efficiency Video Coding Standard

Soulef Bouaafia,[1] **Randa Khemiri**,[1,2] **Amna Maraoui**,[1] **and Fatma Elzahra Sayadi**[1,3]

[1]*Electronics and Microelectronics Laboratory, University of Monastir, Environment Street 5019, Monastir, Tunisia*
[2]*Higher Institute of Computer Science and Multimedia of Gabes, University of Gabes, Gabes, Tunisia*
[3]*National Engineering School of Sousse, University of Sousse, Sousse, Tunisia*

Correspondence should be addressed to Soulef Bouaafia; soulefbouaafia@gmail.com

Received 1 December 2020; Revised 15 February 2021; Accepted 5 March 2021; Published 25 March 2021

Academic Editor: Manuel E. Acacio Sanchez

Copyright © 2021 Soulef Bouaafia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

High-Efficiency Video Coding provides a better compression ratio compared to earlier standard, H.264/Advanced Video Coding. In fact, HEVC saves 50% bit rate compared to H.264/AVC for the same subjective quality. This improvement is notably obtained through the hierarchical quadtree structured Coding Unit. However, the computational complexity significantly increases due to the full search Rate-Distortion Optimization, which allows reaching the optimal Coding Tree Unit partition. Despite the many speedup algorithms developed in the literature, the HEVC encoding complexity still remains a crucial problem in video coding field. Towards this goal, we propose in this paper a deep learning model-based fast mode decision algorithm for HEVC intermode. Firstly, we provide a deep insight overview of the proposed CNN-LSTM, which plays a kernel and pivotal role in this contribution, thus predicting the CU splitting and reducing the HEVC encoding complexity. Secondly, a large training and inference dataset for HEVC intercoding was investigated to train and test the proposed deep framework. Based on this framework, the temporal correlation of the CU partition for each video frame is solved by the LSTM network. Numerical results prove that the proposed CNN-LSTM scheme reduces the encoding complexity by 58.60% with an increase in the BD rate of 1.78% and a decrease in the BD-PSNR of -0.053 dB. Compared to the related works, the proposed scheme has achieved a best compromise between RD performance and complexity reduction, as proven by experimental results.

1. Introduction

Nowadays, there is the emerging technology of new generation digital media and the rapid development of multimedia applications, such as HD and UHD surveillance camera applications in smart city, and the speedy growth of the smart connected devices (IoT) that stream video in a real-time manner. Thus, its popularity has drawn attention from both industry and the academic community. However, computational devices capacity, such as CPU and GPU, and memory capacities have been challenged by the dramatically increasing multimedia data. In this context, video content growth made an urgent requirement for an efficient coding technology that can support this technological outbreak and avoid performance degradation while maintaining a high quality level.

High-Efficiency Video Coding (HEVC) is the sophisticated video coding standard, also known as H.265, standardized in 2013 [1]. Compared to the Advanced Video Coding H.264/AVC standard, HEVC saves 50% bit rate for the same subjective quality [2]. HEVC adopts a flexible hierarchical structure, called quadtree, which includes Coding Unit (CU), Prediction Unit (PU), and Transform Unit (TU) [3]. In this regard, the basic coding structure is the Coding Tree Unit (CTU). The CTU size that ranges from 64×64 to 8×8 can be divided into several CUs of different sizes from 64×64 with a depth of 0 to 8×8 with a depth of 3. Figure 1 illustrates the structure of the hierarchical quadtree.

In addition, HEVC offers two partition modes: intracoding and intercoding units. In fact, the intercoding is the most critical module in HEVC due to its computational complexity when searching the optimal prediction mode. In

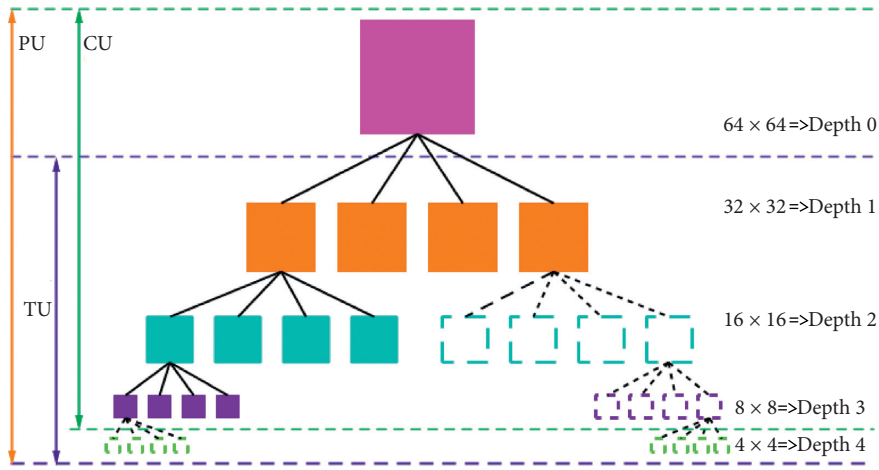

Figure 1: Hierarchical quadtree partition.

order to find the best CU depth, the exhaustive search in each CTU continues until the minimum possible CU size is reached. The latter is known as Rate-Distortion Optimization (RDO). In HEVC, the RDO is computed from all possible depth levels, in which the best CU modes are determined via the RDO minimum. Due to the full RDO search, the HEVC computational complexity has considerably increased, making compression speed a crucial problem in HEVC implementation. Therefore, it is necessary to reduce the intercoding HEVC complexity.

To this end, recent researches have been proposed to reduce coding complexity while reducing RD performance. These researches are based on either classic or deep learning techniques [4–9]. In [4], a fast method-based early CU termination and search range adjustment is proposed by Tai et al. to optimize the encoding efficiency. In the same way, the authors in [5] have developed a fast scheme for HEVC intermode using motion vector information, which aims to accelerate computational complexity. On the other hand, the past few years have seen the success of deep learning technology in many application areas, where video coding has achieved favorable outcome [6, 7]. For HEVC intracoding, Chen et al. [8] suggested a fast-learned algorithm-based asymmetric kernel CNN. This approach has achieved better encoding efficiency, as demonstrated by the experimental results. Regarding the intercoding, the authors in [9] developed a machine learning tool in order to predict the CU mode partition, which provides a good tradeoff between encoding time and RD performance. All these approaches did not model the temporal correlation in video frames at intercoding.

In this light, this paper proposes a deep learning tool that reduces HEVC complexity in terms of encoding time and RD performances. The main contribution consists of a structural combination between the CNN and the LSTM networks. The former is proposed to predict CU splitting and to reduce the performance of HEVC encoding. In HEVC intercoding, there are long-term and short-term dependencies of the intercoding CU splitting between neighboring video frames. Unfortunately, the deep CNN does not explore this temporal correlation; for these reasons, the LSTM network must be in place. That is how the CNN-LSTM-based learning approach is proposed, which predicts the intercoding CU partition, instead of the classical RDO search.

The remainder of this paper is structured as follows: Section 2 introduces an overview including deep learning algorithms in video coding and the heuristic methods. The proposed scheme is presented in Section 3, for reducing the HEVC complexity at interprediction. Section 4 shows the experimental results, while Section 5 concludes this paper.

2. Related Work Overview

To optimize the HEVC coding efficiency, fast methods have been suggested for reducing the HEVC complexity caused by the quadtree partition. These fast methods can be summarized into two classes: heuristic approaches and machine-learning-based schemes [10–21].

In heuristic approaches, some fast CU decision schemes have been developed to simplify the RDO process towards reducing HEVC complexity [10–15]. For example, Cho and Kim [10] proposed a Bayesian rule-based fast CU partition and pruning algorithm. With regard to HEVC intercoding, Shen et al. in [11] developed a fast intercoding decision scheme using interlevel and spatiotemporal correlations in which the motion vector, RD cost, and prediction mode were found to be strongly correlated. To reduce the HEVC complexity, a fast CU partitioning and mode decision method using a look-ahead stage is proposed in [12]. The authors in [13] introduced a fast algorithm to split CUs at the HEVC intercoding based on pyramid motion divergence. To overcome encoding complexity for intercoding HEVC, based on temporal and spatial correlation, a fast CU size decision scheme is proposed by Zhang et al. in [14]. In addition, the authors introduced in [15] an adaptive motion search range method to reduce the HEVC encoding efficiency.

On the other hand, the search of the optimal CU prediction mode can be modeled as classification problem. In this regard, researchers adopted learning-based methods in classifying CU mode decision in order to reduce the

computational complexity [16–22]. Shen and Yu [16] proposed a CU early termination algorithm for each level of the quadtree CU partition based on weighted SVM. In addition, a fast CU decision method based on fuzzy SVM is suggested by Zhu et al. in [17] to enhance the coding efficiency. For complexity reduction, in [18], the authors developed a neural networks-based fast CU mode decision to predict the split for intramode and intermode. Similarly, in [19], Xu et al. suggested a hierarchical CU depth decision based on the LSTM network, predicting HEVC CU splitting for H.264 to HEVC transcoding. Reinforcement learning (RL) and deep RL are also adopted in video coding to learn a classification task and to find the optimal CU mode decision. In [20], a CU early termination algorithm for HEVC was developed using an end-to-end actor-critic RL to improve the coding complexity.

For video coding, the similarity in video content is shown by the adjacent frames in video sequence, which decreases along with temporal distance between two images. In this article, we develop an LSTM network to study the CU partition correlation at intercoding. This is because the deep CNN proposed in [9] does not explore the temporal information of CU partition for each HEVC frame. Then, we combine a CNN-LSTM learning scheme to predict the intercoding CU splitting, which reduces the computational complexity of HEVC [9].

3. Proposed Framework Based on Deep Learning

In this section, we first start by awarding the intercoding dataset needed for the model learning process. Then, we introduce the proposed CNN-LSTM network-based scheme to predict the intercoding CU partition for HEVC, thus reducing the encoding complexity.

3.1. Database for the Intercoding. The HEVC intermode CU partition dataset has been created to learn the proposed model. However, 114 video sequences are selected with various resolutions (from 352×240 to 2560×1600) [23–25] to construct the database. The latter is made up of three labeled groups (sequences) including 86, 10, and 18 video sequences for training, validation, and testing, respectively. HEVC encoder is used to compress the database video sequences at four Quantization Parameters (QPs) {22, 27, 32, 37}, using the Low-Delay P (LDP) configuration [26]. To provide more details, interested reader can refer to the previous paper in [9].

3.2. CNN-LSTM Network. According to the correlation of the HEVC CU splitting of adjacent frames, the proposed scheme is introduced in this section. The proposed LSTM network learns the interframe temporal correlation for each video sequence. In addition, the proposed algorithm that combines CNN-LSTM is presented in Figure 2. The deep CNN is composed of three convolutional layers (Conv1, Conv2, and Conv3), a concatenated vector, and a fully connected layer. As presented in [9], the deep CNN parameters are learned based on the ground truth and the residual CTU, and then the extracted features $(FC_{1-l})_{l=1}^{3}$ of the deep CNN are the input of the proposed LSTM network at frame t. These features $(FC_{1-l})_{l=1}^{3}$ are extracted at the first fully connected layer of the deep CNN.

We can see from Figure 2 that the architecture of the LSTM is composed of three LSTM cells corresponding to three levels splitting of each CU. Specifically, $\widetilde{F}_1(CU,t)$ at level 1 indicates whether the 64×64 size CU will be split into 32×32 size sub-CUs or not. At level 2, $\{\widetilde{F}_2(CU_i,t)\}_{i=0}^{3}$ and $\{\widetilde{F}_3(CU_{i,j},t)\}_{i,j=0}^{3}$ designate, respectively, the CPUs partitioning labels from 32×32 to 16×16 and from 16×16 to 8×8. At each level, two fully connected layers, containing a hidden layer and an output layer follow the LSTM cells. In addition, the output features of the LSTM cells are denoted by $(FC_l')_{l=1}^{3}$ at frame t. However, the next-level LSTM cell is activated to decide on the next four sub-CUs, if the CU of the current level is split. Otherwise, the prediction on splitting the current CU is terminated early. Finally, the predicted CU partition of three levels is denoted by $\widetilde{F}_1(CU,t)$, $\{\widetilde{F}_2(CU_i,t)\}_{i=0}^{3}$, and $\{\widetilde{F}_3(CU_{i,j},t)\}_{i,j=0}^{3}$, as shown in Figure 2. The ReLU and the sigmoid activation functions are used to activate the hidden and the output layers, respectively [27].

The LSTM model learns the long short-term dependency of the CTU depths when the CTU partition is predicted. Then, the LSTM cell consists of three gates, as shown in Figure 3: the input gate $i_l(t)$, the forget gate $f_l(t)$, and the output gate $o_l(t)$, respectively. At level l, $FC_{1-l}(t)$ represents the deep CNN input features at frame t, and $FC_l'(t-1)$ represents the output features of the LSTM model of frame t-1. In equations (1), 2, and (3), these three gates are presented:

$$i_l(t) = \sigma\left(w_i \cdot \left[FC_{1-l}(t), FC_l'(t-1)\right] + b_i,\right. \tag{1}$$

$$o_l(t) = \sigma\left(w_o \cdot \left[FC_{1-l}(t), FC_l'(t-1)\right] + b_o,\right. \tag{2}$$

$$f_l(t) = \sigma\left(w_f \cdot \left[FC_{1-l}(t), FC_l'(t-1)\right] + b_f,\right. \tag{3}$$

where the sigmoid function is denoted by $\sigma(\cdot)$. $\{w_i, w_o$ and $w_f\}$ are the weights and $\{b_i, b_o$ and $b_f\}$ are the biases for the three gates. At frame t, the state $c_l(t)$ of the LSTM cell can be updated by

$$c_l(t) = i_l(t) \circ \tanh\left(w_c \cdot \left[FC_{1-l}(t), FC_l'(t-1)\right] + b_c + f_l(t) \circ c_l(t-1).\right. \tag{4}$$

The element-wise product is designated by $\circ$. The biases and the weights for the state of the LSTM cell are w_c and b_c. In the following, the LSTM cell output $FC_l'(t)$ can be defined in the following equation:

$$FC_l'(t) = o_l(t) \circ \tanh\left(c_l(t)\right). \tag{5}$$

3.3. Loss Function. In the training process, the training set of the intercoding dataset is used to train the LSTM network, where the trained model minimizes the loss function between the CTU partition prediction and the ground truth. Figure 4 shows the learning process. The Stochastic Gradient

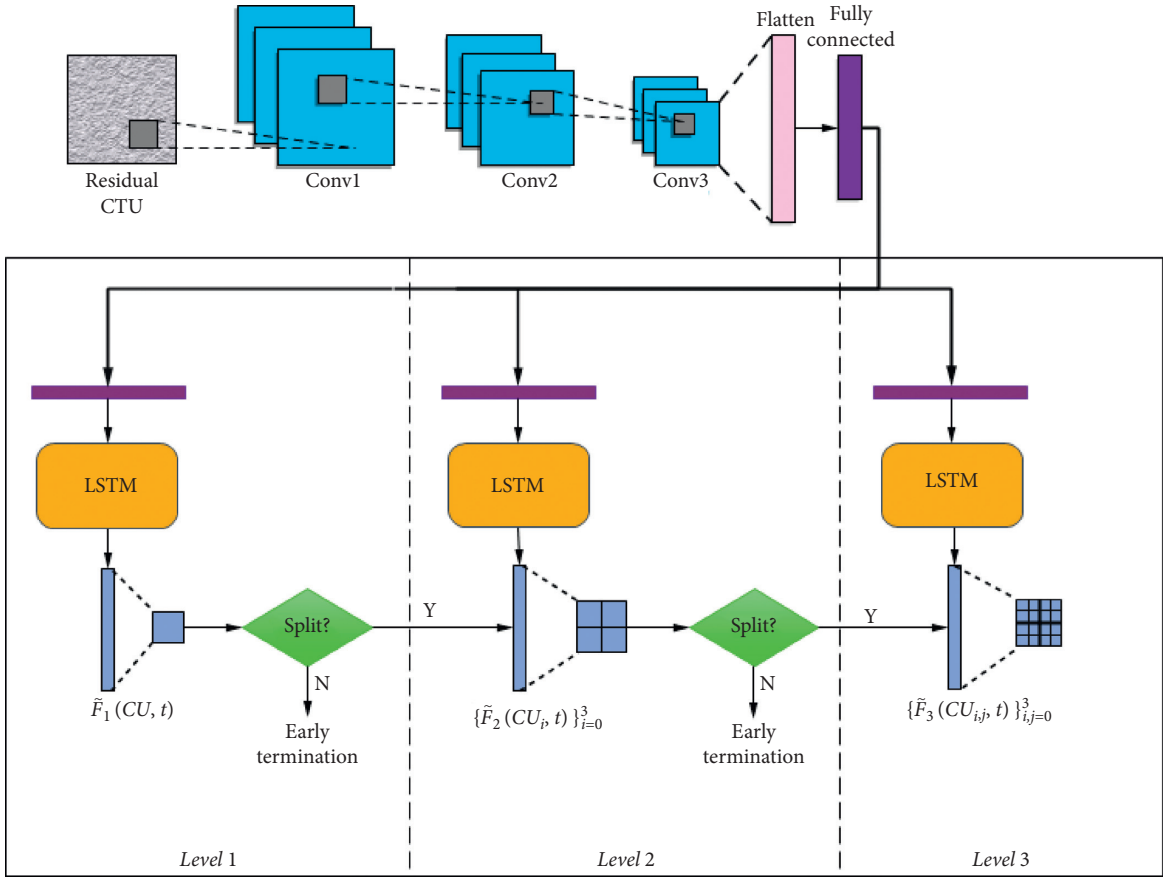

Figure 2: Proposed scheme framework.

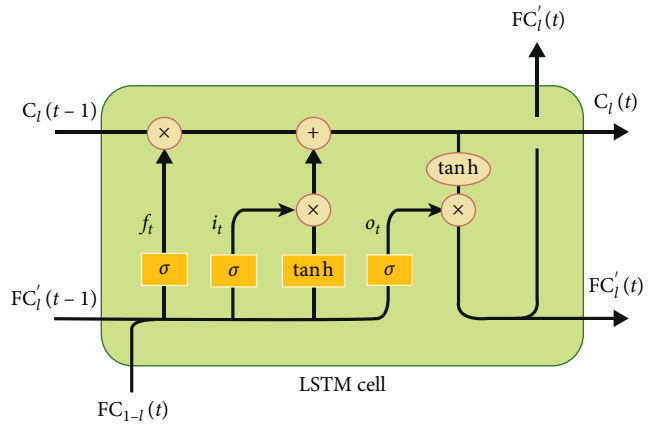

Figure 3: LSTM cell.

Descent (SGD) algorithm is considered to be the powerful optimization algorithm to learn the network structure through their feedforward and backward subprocess, where the cross entropy is selected to be the cost function of the gradient error calculator designated by $Y(\cdot, \cdot)$ in (6). At frame t, the loss function Ln (t) for the n-th sample CU is written as follows:

$$L_n(t) = Y\left(F_1^n(CU,t), \widetilde{F}_1^n(CU,t)\right) + \sum_{i,j\in\{0,1,2,3\}} Y\left(F_2^n(CU_i,t), \widetilde{F}_2^n(CU_i,t)\right) + \sum_{i,j\in\{0,1,2,3\}} Y\left(F_3^n\left(CU_{i,j},t\right), \widetilde{F}_3^n\left(CU_{i,j},t\right)\right). \quad (6)$$

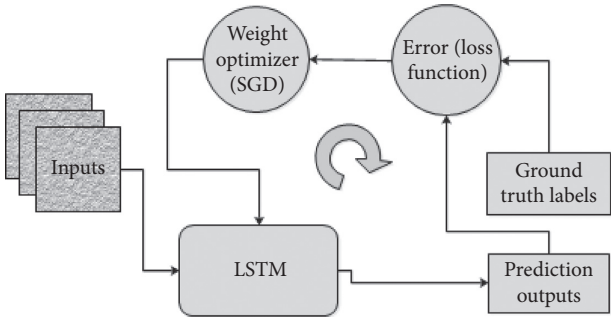

Figure 4: Learning process.

However, over N training samples alongside the T-frames, the LSTM network can be learned by optimizing the cost function.

$$L = \frac{1}{NT} \sum_{n=0}^{N} \sum_{t=0}^{T} L_n(t). \tag{7}$$

4. Experimental Results

4.1. Experimental Setting. In this section, we present the obtained results to validate the coding efficiency of the proposed deep learning framework. Our experiments were performed in the HM16.5 reference test model [26], which were tested on 18 JCT-VC videos from class A to class E with four QPs {22, 27, 32, 37}, using the LDP configuration. The number of frames used for each video sequence is 100. All implementations were executed on Windows 10 OS platform with Intel ®core TM i7-3770 @ 3.4 GHz CPU and 16 GB RAM, in which the compression efficiency of the proposed scheme is evaluated. To accelerate the speed of the network model-training phase, we also used the NVIDIA GeForce GTX 480 GPU, but it was not used in the HEVC complexity reduction test. In the experiments, the TensorFlow-GPU deep learning framework was used. The simulation parameters were defined as follows: batch size, learning rate, and LSTM length (T) were set to 64, 0.001, and $T = 20$, respectively. Finally, the trained model was saved to be used after (in the framework), which aims to predict the intercoding CU partition.

For the test, the LSTM model operates in stages; that is, when the prediction of the CU partition at frame t-1 has been completed, the state and the output of the frame t are computed. To further enhance the RD performance and reduce the intercoding computational complexity, the bithreshold decision scheme was adopted at three levels. Note that the upper and the lower thresholds at level l are represented by $\{\gamma_l\}_{l=1}^{3}$ and $\{\overline{\gamma}_l\}_{l=1}^{3}$. At three levels, the LSTM network provides the predicted CU partition probability (P_l (CU)). Consequently, the CU decides to be split only when $P_l(CU) > \gamma_l$. In this way, the HEVC complexity is reduced considerably by skipping the most redundant verification of the RD cost.

4.2. Evaluation Criteria. The RD performance analysis is performed based on the Bjøntegaard Delta rate (BD rate) and the Bjøntegaard Delta Peak Signal-to-Noise Ratio (BD-PSNR) [28]. The BD rate represents the average bit rate savings calculated between two RD curves for the same video quality, where negative BD-rate values indicate actual bit rate savings and positive values indicate how much the bit rate is increased. BD-PSNR is the overall PSNR difference of RD curves with the same bit rate in decibel, not forgetting that the encoding time is modeled as the critical metric for the validation performance of the HEVC at intermode, as shown in the following equation:

$$\Delta T = \frac{T_P - T_o}{T_o} \times 100\,(\%), \tag{8}$$

where T_p and T_o are the execution times of the proposed approach and the original HEVC, respectively.

4.3. Simulations and Results Analysis. Table 1 demonstrates the achieved results of the fast proposed scheme compared to the original HEVC under LDP configuration in terms of BD-PSNR, BD rate, and time saving, respectively. It can be observed from this table that the results concerning computational-complexity reduction are significant and reach up to 75% for some sequences. As shown above, the proposed scheme reduces the execution time on average by 58.60% with a maximum of 74.64% for class E, since the activities displayed in these sequences are with low motion, which leads to larger partitions. A minimum of 52.48% is obtained in class C, since the video sequences in this class have high motion and rich textures. This clearly proves that the proposed method can adapt well to video content with low motion and gives higher speedup when compared with original HEVC. Concerning the RD performance of our approach, the BD rate is averagely 1.78% with negligible decrease in BD-PSNR, around −0.053 dB, compared with original HEVC. In summary, the proposed CNN-LSTM model is better in terms of RD performance and HEVC computational-complexity reduction.

Figure 5 gives the RD curves of the suggested approach and the original HEVC for different video sequences with ultra-high-definition sequence and the high-definition sequence, respectively. In this figure, the difference of RD performance between the original HEVC and the proposed algorithm is very small for all QPs. This justifies that the learning technique can well adapt to the different bit-rate points for ultra-high-definition and high-definition videos.

Furthermore, Figure 6 reports the time saving of "Traffic" (2560 × 1600) and "BasketballDrive" (1920 × 1080) while varying QP. It can be noted that the encoding time increases proportionally, while the QP value increases. However, the proposed CNN-LSTM model outperforms the original HEVC in terms of complexity reduction.

For further performance evaluation of the proposed scheme, Table 2 shows the coding performance between the proposed CNN-LSTM framework and the deep CNN [9]. The proposed scheme CNN-LSTM is better than the deep CNN in terms of computational complexity and BD-PSNR performance. Specifically, the execution time of our method is 58.60% on average, which exceeds 53.99% when using CNN only [9]. On the other hand, the proposed approach

Table 1: Simulation results of the proposed scheme versus original HEVC.

Class	Sequence	CNN-LSTM versus Original HM		
		BD rate (%)	BD-PSNR (dB)	ΔT (%)
A (2560 × 1600)	PeopleOnStreet	1.70	−0.017	−48.88
	Traffic	1.53	−0.059	−66.38
Average class A		1.61	−0.038	−57.63
B (1920 × 1080)	Kimono	1.65	−0.052	−47.77
	ParkScene	2.79	−0.081	−70.82
	Cactus	1.73	−0.033	−53.85
	BQTerrace	1.75	−0.030	−65.62
	BasketballDrive	2.02	−0.045	−52.77
Average class B		1.98	−0.048	−58.16
C (832 × 480)	BasketballDrill	1.67	−0.061	−48.23
	BQMalls	1.38	−0.090	−48.07
	PartyScene	0.96	−0.038	−59.30
	RaceHorses	1.47	−0.055	−54.32
Average class C		1.37	−0.061	−52.48
D (416 × 240)	BasketballPass	1.26	−0.056	−56.67
	BQSquare	1.27	−0.046	−60.79
	BlowingBubbles	0.97	−0.034	−50.14
	RaceHorses	1.60	−0.018	−50.33
Average class D		1.27	−0.038	−54.48
E (1280 × 720)	FourPeople	2.71	−0.071	−72.42
	Johnny	2.46	−0.083	−73.59
	KristenAndSara	2.93	−0.094	−74.91
Average class E		2.7	−0.082	−74.64
Average		1.78	−0.053	−58.60

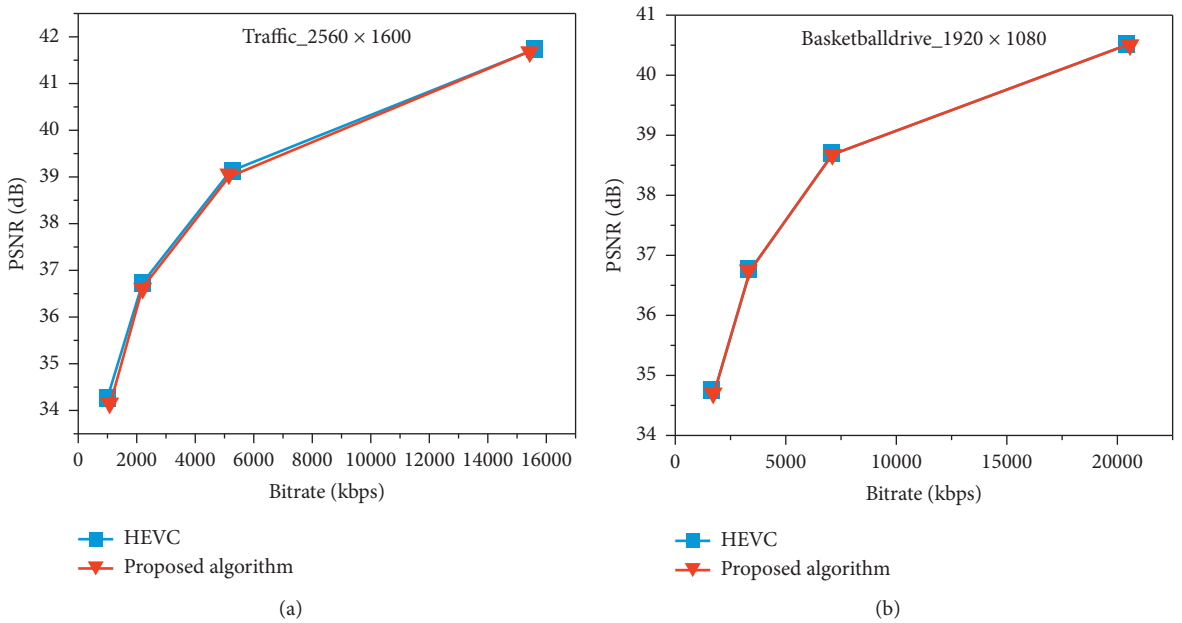

Figure 5: RD curves of the proposed approach and the original HEVC.

can reduce the BD-PSNR performance by −0.053 dB, which is better than −0.057 dB achieved by [9]. Furthermore, our proposed approach has an average BD-rate performance of 1.78%, better than that of [9], 1.80%. In our experiments, we note that the proposed deep learning achieves high HEVC complexity reduction at intercoding, because it is able to predict all the CU splitting of an entire CTU at the same time. The proposed algorithm also performs well in terms of BD-PSNR performance, due to the high accuracy of the predicted CU partition. Consequently, the learning scheme

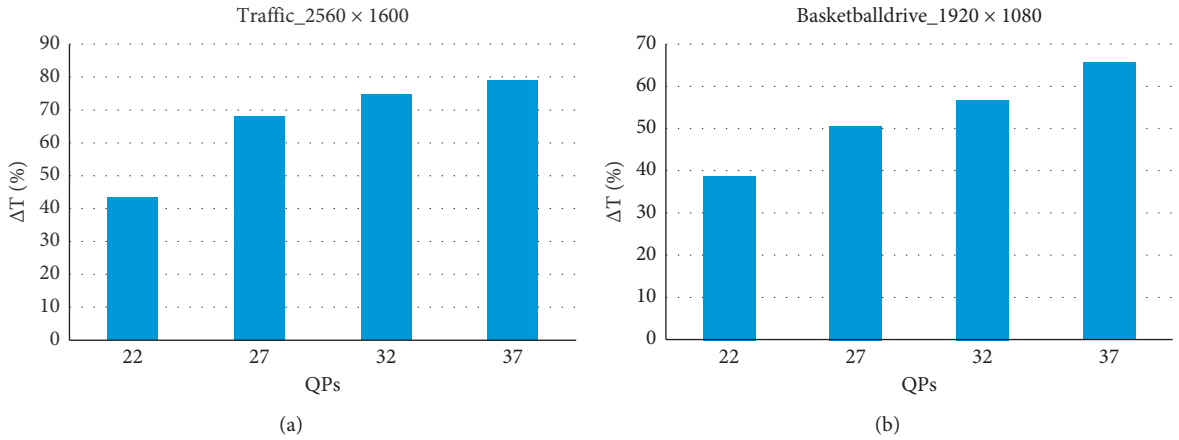

Figure 6: Complexity reduction of original HM and proposed approach under QPs {22, 27, 32, 37} for "Traffic" and "BasketballDrive."

Table 2: Performances evaluation [CNN-LSTM versus deep CNN].

Class	Sequence	Deep CNN [9]			Proposed CNN-LSTM		
		BD rate (%)	BD-PSNR (dB)	ΔT (%)	BD rate (%)	BD-PSNR (dB)	ΔT (%)
A (2560 × 1600)	PeopleOnStreet	1.20	−0.051	−50.67	1.70	−0.017	−48.88
	Traffic	1.49	−0.041	−57.90	1.53	−0.059	−66.38
B (1920 × 1080)	Kimono	1.38	−0.044	−43.26	1.65	−0.052	−47.77
	ParkScene	1.43	−0.041	−64.14	2.79	−0.081	−70.82
	Cactus	2.44	−0.047	−52.57	1.73	−0.033	−53.85
	BQTerrace	2.22	−0.034	−58.43	1.75	−0.030	−65.62
	BasketballDrive	2.28	−0.051	−51.30	2.02	−0.045	−52.77
C (832 × 480)	BasketballDrill	1.43	−0.052	−53.54	1.67	−0.061	−48.23
	BQMall	2.24	−0.085	−52.25	1.38	−0.090	−48.07
	PartyScene	1.48	−0.057	−51.54	0.96	−0.038	−59.30
	RaceHorses	1.41	−0.053	−42.22	1.47	−0.055	−54.32
D (416 × 240)	BasketballPass	1.85	−0.083	−52.42	1.26	−0.056	−56.67
	BQSquare	2.09	−0.073	−52.79	1.27	−0.046	−60.79
	BlowingBubbles	1.71	−0.061	−46.55	0.97	−0.034	−50.14
	RaceHorses	1.32	−0.058	−38.01	1.60	−0.018	−50.33
E (1280 × 720)	FourPeople	1.06	−0.029	−67.54	2.71	−0.071	−72.42
	Johnny	3.99	−0.083	−69.66	2.46	−0.083	−73.59
	KristenAndSara	1.31	−0.082	−67.20	2.93	−0.094	−74.91
Average		**1.80**	**−0.057**	**−53.99**	**1.78**	**−0.053**	**−58.60**

based on CNN-LSTM achieves a good compromise between RD performance and coding complexity in order to predict intermode CU partition of HEVC. This is mainly due to the LSTM ability to resolve the temporal correlation through adjacent frames.

For more evaluation, the reducing complexity of CNN-LSTM versus deep CNN under "Traffic" (2560 × 1600) and "BasketballDrive" (1920 × 1080) video sequences at LDP configuration is proved in Figure 7. As shown in this figure, the proposed approach allows higher encoding time when the QP value increases from 22 to 37. Overall, the proposed deep learning approach outperforms the deep CNN in terms of time saving. Consequently, the proposed scheme is better for reducing the HEVC complexity of intercoding and for finding an optimal CU partition, compared to traditional RDO research.

4.4. Comparative Performance. To evaluate the encoding performance of the proposed learning approach, our experimental results are compared to the other state-of-the-art methods, such as reinforcement-learning-based scheme [20], random-forests-based scheme [28], and deep-learning-approach-based HEVC complexity reduction [29]. Table 3 summarizes the proposed scheme's performance compared to the works based on learning technique cited in [20, 28] and [29].

In this table, the proposed scheme outperforms other schemes in terms of complexity-RD performance. In [20], Li et al. proposed a CU early termination based on reinforcement learning to reduce the computational complexity for HEVC. In [29], Tahir et al. developed a fast method for reducing HEVC encoding time based on random forest classifier. Xu et al. [30] proposed a fast CU partition

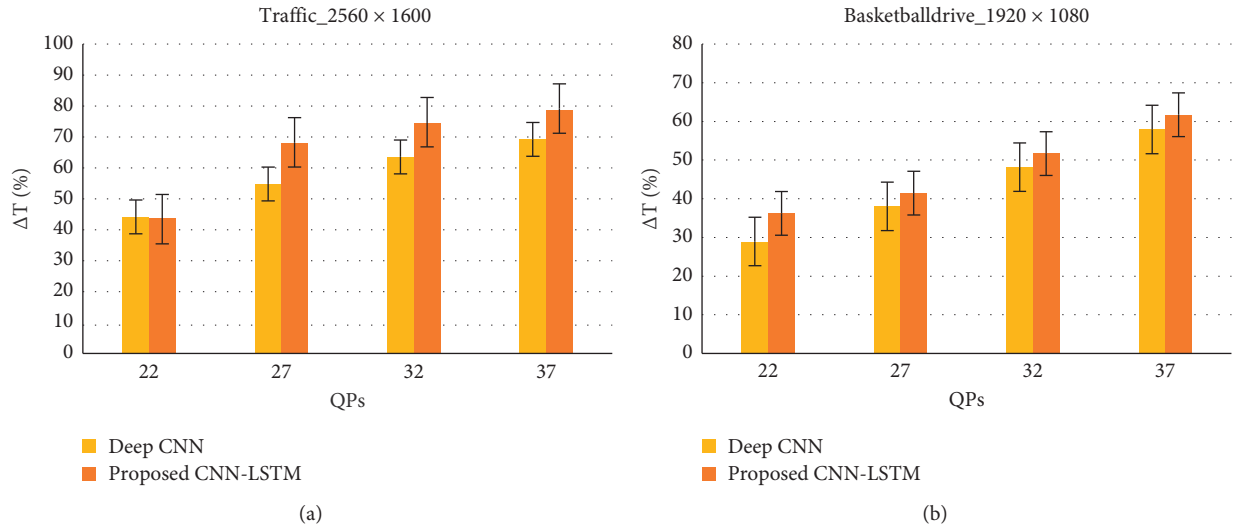

Figure 7: Encoding time of the proposed CNN-LSTM and deep CNN.

Table 3: Comparative performance between the proposed approach and the state-of-the-art works.

Sequence	[30]			[20]			[29]			Proposed approach		
	BD rate (%)	BD-PSNR (dB)	ΔT (%)	BD rate (%)	BD-PSNR (dB)	ΔT (%)	BD rate (%)	BD-PSNR (dB)	ΔT (%)	BD rate (%)	BD-PSNR (dB)	ΔT (%)
PeopleOnStreet	1.05	−0.045	−47.50	5.45	−0.250	29.84	1.85	−0.085	27.22	1.70	−0.017	−48.88
Traffic	1.99	−0.052	−60.60	—	—	—	3.20	−0.102	59.66	1.53	−0.059	−66.38
Kimono	1.49	−0.048	−56.03	0.35	−0.010	34.74	2.43	−0.079	50.49	1.65	−0.052	−47.77
ParkScene	1.47	−0.042	−58.72	2.84	−0.090	46.42	—	—	—	2.79	−0.081	−70.82
Cactus	2.07	−0.043	−56.87	2.79	−0.065	43.22	2.46	−0.060	53.06	1.73	−0.033	−53.85
BQTerrace	1.09	−0.017	−60.01	2.15	−0.038	38.70	1.74	−0.032	51.89	1.75	−0.030	−65.62
BasketballDrive	2.268	−0.052	−55.84	2.06	−0.046	39.45	1.93	−0.046	49.16	2.02	−0.045	−52.77
BasketballDrill	1.953	−0.072	−55.19	3.90	−0.148	32.12	2.24	−0.089	46.55	1.67	−0.061	−48.23
BQMall	1.914	−0.071	−50.74	5.56	−0.227	37.11	1.86	−0.071	43.32	1.38	−0.090	−48.07
PartyScene	1.011	−0.039	−46.83	4.74	−0.210	31.58	2.58	−0.108	30.33	0.96	−0.038	−59.30
RaceHorses	0.872	−0.032	−46.22	2.10	−0.180	26.03	1.25	−0.048	27.08	1.47	−0.055	−54.32
BasketballPass	1.45	−0.066	−49.04	—	—	—	3.12	−0.147	38.24	1.26	−0.056	−56.67
BQSquare	0.77	−0.028	−46.91	3.38	−0.145	35.72	3.02	−0.117	34.30	1.27	−0.046	−60.79
BlowingBubbles	1.29	−0.044	−45.62	3.41	−0.136	24.73	3.99	−0.154	39.87	0.97	−0.034	−50.14
RaceHorses	1.11	−0.047	−41.86	—	—	—	—	—	—	1.60	−0.018	−50.33
FourPeople	1.83	−0.052	−64.37	1.66	−0.058	65.28	1.83	−0.063	78.98	2.71	−0.071	−72.42
Johnny	1.69	−0.038	−66.49	0.90	−0.020	64.05	5.45	−0.139	79.31	2.46	−0.083	−73.59
KristenAndSara	1.55	−0.045	−67.23	1.58	−0.050	64.67	3.30	−0.108	77.58	2.93	−0.094	−74.91
Average	**1.49**	**−0.046**	**−54.2**	**2.56**	**−0.099**	**43.33**	**2.97**	**−0.107**	**54.57**	**1.78**	**−0.053**	**−58.60**
FoM	**2.74**			**5.90**			**5.44**			**3.03**		

algorithm for HEVC, including interprediction and intraprediction, based on deep learning approach for reducing HEVC complexity. Specifically, the proposed approach achieves a maximum execution time of 75% and 58.60% on average and gives an increase in BD rate of 1.78% with a little reduction in BD-PSNR of −0.053 dB.

In fact, the proposed approach achieves a higher computational-complexity reduction for video sequences with low-motion activities and homogeneous regions, where the blocks CU partition is larger and the percentage of splitting cases is lower, such as "KristenAndSara" video sequence. Similarly, the existing methods prove a high encoding time for class E video sequences. For example, [20] achieves 64% encoding complexity and 1.58% BD-rate increase for sequence "KristenAndSara," as shown in Table 3. For the same sequence, [29] gives 77% time saving with an increase in the BD rate of 3.30% on average of four QPs. In addition, the work proposed in [30] achieves 67.23% encoding time with 1.55% BD rate on average.

With regard to the ultra-high-definition sequences like "PeopleOnStreet," the computational-complexity reduction of our proposed approach is slightly lower, since these sequences have high motion and camera movement, which are encoded in a small CU partition. Hence, the proposed scheme performs better in terms of both RD performance and complexity reduction of HEVC as compared to the

previous works. Overall, all approaches are better adapted to low-motion video content.

On the other hand, in average, 43% time saving is reduced by [20] with an increase in BD rate of 2.56% and a decrease in BD−PSNR of −0.099 dB. The proposed method presented in [29] allows 54.57% encoding time, while the BD rate increases by 2.97% and the BD-PSNR degradation reaches −0.107 dB. Regarding the work invented in [30], the proposed method surpasses our proposed approach in terms of BD rate and BD-PSNR, while our proposed approach saves significant coding time of 58.60% compared to this work. When comparing our work to the state-of-the-art schemes in [20, 29] and [30], we can conclude that the proposed CNN-LSTM-based learning method proves the best coding efficiency of HEVC at intermode in order to predict the CU partition.

In summary, several existing works can achieve significant computational-complexity reduction with low BD rate and vice versa. Each method presents different values for computational-complexity reduction and BD rate. Therefore, an algorithm can achieve a tradeoff between complexity reduction and RD efficiency; we have used two Figures of Merit (FoM), BD rate and ΔT, a common procedure of computing proposed in [31, 32]:

$$\text{FoM} = \left|\frac{\text{BD\_}rate}{\Delta T}\right| \times 100. \tag{9}$$

FoM represents the ratio between the increase in the BD rate and the encoding time reduction, allowing direct comparison of competing algorithms. Therefore, FoM makes the best compromise between low-penalty and high-complexity reductions. Table 3 presents the FoM of the proposed approach compared to the existing methods. The lower value of FoM is desirable because it is interpreted as a best tradeoff between RD efficiency and computational-complexity reduction. Compared with related works, it can be observed that our proposed approach achieves a FoM ratio of around 3.03 and it is demonstrated that our proposed framework based on CNN-LSTM presents a good balance between low-penalty and high-complexity reductions.

5. Conclusion

This paper proposed a CNN-LSTM learning scheme to predict the optimal intercoding CU partition, thus maximizing the reduction in HEVC coding complexity. According to the temporal correlation of neighboring frames, we developed a new LSTM architecture to learn the long-term and short-term correlations of the intercoding CU partition. This model learns to find the best CU prediction modes instead of traditional RDO search. The achieved results demonstrate that the proposed approach based on deep learning reduces the computational complexity by 58.60% with an increase in BD rate by approximately 1.78% and the BD-PSNR decreases by -0.027 dB under LDP configuration. Consequently, the HEVC encoding complexity can be considerably reduced, when we replace the classical RDO search with the CNN-LSTM network to decide the CU splitting at intermode. In summary, the proposed scheme saves a significant encoding complexity, compared to other previous approaches based on machine learning tools.

Data Availability

The dataset used in this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

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

[2] R. Khemiri, H. Kibeya, F. E. Sayadi, N. Bahri, M. Atri, and N. Masmoudi, "Optimization of HEVC motion estimation exploiting SAD and SSD GPU-based implementation," *IET Image Processing*, vol. 12, no. 2, pp. 243–253, 2017.

[3] R. Khemiri, N. Bahri, F. Belghith et al., *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, UK, 2019.

[4] K. H. Tai, M. Y. Hsieh, M. J. Chen, C. Y. Chen, and C. H. Yeh, A fast HEVC encoding method using depth information of collocated CUs and RD cost characteristics of PU modes," *IEEE Transactions on Broadcasting*, vol. 63, no. 4, pp. 680–692, 2017.

[5] K. M. Lin, J. R. Lin, M. J. Chen, C. H. Yeh, C. A. Lee, Fast interprediction algorithm based on motion vector information for high efficiency video coding," *EURASIP Journal on Image and Video Processing*, vol. 1, p. 99, 2018.

[6] D. Liu, Y. Li, J. Lin, H. Li, and F. Wu, "Deep learning-based video coding: a review and a case study," 2019, https://arxiv.org/abs/1904.12462.

[7] D. Wang, S. Xia, W. Yang, and J. Liu, "Combining progressive rethinking and collaborative learning: a deep framework for in-loop filtering," 2020, https://arxiv.org/abs/2001.05651.

[8] J. Chen and S. W. Li, "Learned fast HEVC intra coding," 2019, https://arxiv.org/abs/1907.02287.

[9] S. Bouaafia, R. Khemiri, F. E. Sayadi, and M. Atri, "Fast CU partition-based machine learning approach for reducing HEVC complexity," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 185–196, 2019.

[10] S. Cho and M. Kim, "Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 9, pp. 1555–1564, 2013.

[11] L. Shen, Z. Zhang, and Z. Liu, "Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatio-temporal correlations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, pp. 1709–1722, 2014.

[12] G. Cebrián-Márquez, J. L. Martínez, and P. Cuenca, "Adaptive inter CU partitioning based on a look-ahead stage for HEVC," *Signal Processing: Image Communication*, vol. 76, pp. 97–108, 2019.

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

[14] Q. Zhang, J. Zhao, X. Huang, and Y. Gan, "A fast and efficient coding unit size decision algorithm based on temporal and spatial correlation," *Optik*, vol. 126, no. 21, pp. 2793–2798, 2015.
[15] H. Ding, F. Wang, W. Zhang, and Q. Zhang, "Adaptive motion search range adjustment algorithm for HEVC inter coding," *Optik*, vol. 127, no. 19, pp. 7498–7506, 2016.
[16] X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP J. Image Video Process*, vol. 4, pp. 1–11, 2013.
[17] L. Zhu, Y. Zhang, S. Kwong, X. Wang, and T. Zhao, "Fuzzy SVM based coding unit decision in HEVC," *IEEE Transaction on Broadcasting*, vol. 64, pp. 681–694, 2017.
[18] K. Kim and W. Ro, "Fast CU depth decision for HEVC using neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, in press.
[19] J. Xu, M. Xu, Y. Wei, Z. Wang, and Z. Guan, "Fast, 264 to HEVC Transcoding: a deep learning method," *IEEE Transaction on Multimedia*, 2018, in press.
[20] N. Li, Y. Zhang, L. Zhu, W. Luo, and S. Kwong, "Reinforcement learning based coding unit early termination algorithm for high efficiency video coding," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 276–286, 2019.
[21] M. Ramezanpour and F. Zargari, "Fast CU size and prediction mode decision method for HEVC encoder based on spatial features," *Signal, Image and Video Processing*, vol. 10, no. 7, pp. 1233–1240, 2016.
[22] S. Bouaafia, R. Khemiri, F. E. Sayadi, M. Atri, and N. Liouane, "A deep CNN-LSTM framework for fast video coding," in *Proceedings of the International Conference on Image and Signal Processing*, pp. 205–212, Springer, Marrakech, Morocco, June 2020.
[23] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-interest based conversational HEVC coding with hierarchical perception model of face," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 3, pp. 475–489, 2014.
[24] F. Bossen, *Common Test Conditions and Software Reference Configurations, Document JCTVC-L1100, Joint Collaborative Team on Video Coding*, vol. 12, 2013.
[25] 2017 Video Test Media. [Online] https://media.xiph.org/video/derf.
[26] *HEVC Test Model*, https://hevc.hhi.fraunhofer.de/svn/svn%20HEVCSoftware.
[27] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 315–323, Tokyo, Japan, March 2011.
[28] G. Bjontegaard, "Calculation of average PSNR differences between RD_curves," in *Proceedings of the Presented at the 13th VCEG-M33 Meeting*, Austin, TX, USA, April 2001.
[29] M. Tahir, I. A. Taj, P. A. Assuncao, and M. Asif, "Fast video encoding based on random forests," *Journal of Real-Time Image Processing*, vol. 17, pp. 1029–1049, 2020.
[30] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, "Reducing complexity of HEVC: a deep learning approach," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5044–5059, 2018.
[31] N. Najafabadi and M. Ramezanpour, "Mass center direction-based decision method for intraprediction in HEVC standard," *Journal of Real-Time Image Processing*, vol. 17, no. 5, pp. 1153–1168, 2020.
[32] B. Heidari and M. Ramezanpour, "Reduction of intra-coding time for HEVC based on temporary direction map," *Journal of Real-Time Image Processing*, vol. 17, no. 3, pp. 567–579, 2020.