A Deep CNN-LSTM Framework for Fast Video Coding

Soulef Bouaafia¹(✉) , Randa Khemiri¹, Fatma Ezahra Sayadi¹ ,
Mohamed Atri² , and Nouredine Liouane³

¹ Faculty of Sciences, University of Monastir, Monastir, Tunisia

{soulef.bouaafia, randa.khemiri}@fsm.rnu.tn, sayadi_fatma@yahoo.fr

² College of Computer Science, University of King Khalid, Abha, Saudi Arabia
matri@kku.edu.sa

³ National Engineering School of Monastir, University of Monastir, Monastir, Tunisia
nouredine.liouane@enim.rnu.tn

Abstract. High Efficiency Video Coding (HEVC) doubles the compression rates over the previous H.264 standard for the same video quality. To improve the coding efficiency, HEVC adopts the hierarchical quadtree structured Coding Unit (CU). However, the computational complexity significantly increases due to the full search for Rate-Distortion Optimization (RDO) to find the optimal Coding Tree Unit (CTU) partition. Here, this paper proposes a deep learning model to predict the HEVC CU partition at inter-mode, instead of brute-force RDO search. To learn the learning model, a large-scale database for HEVC inter-mode is first built. Second, to predict the CU partition of HEVC, we propose as a model a combination of a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) network. The simulation results prove that the proposed scheme can achieve a best compromise between complexity reduction and RD performance, compared to existing approaches.

Keywords: HEVC · CU partition · Deep learning · CNN · LSTM

1 Introduction

Over the last decades, we have noticed the success of deep learning in many application areas, where video and image processing has achieved a favorable outcome. The state-of-the-art video coding is High Efficiency Video Coding (HEVC), also known as H.265, which was standardized in 2013 [1]. Compared to its predecessor H.264/AVC standard, HEVC saves an average BitRate reduction of 50%, while maintaining the same video quality [2]. The hierarchical coding structure adopted in HEVC is the quadtree, including Coding Unit (CU), Prediction Unit (PU), and Transform Unit (TU) [3]. In this regard, the Coding Tree Unit (CTU) is the basic coding structure in which the size of the CTU is 64×64 . A CTU can be divided into multiple CUs of different sizes from 64×64 with a depth of 0 to 8×8 with a depth of 3.

This exhaustive splitting continues until the minimum possible size of a CU is reached in order to find the optimal CU depth. This process is known as Rate-Distortion Optimization (RDO) which is required for each CTU. Due to the full RDO search, the HEVC computational complexity has considerably increased, making encoding speed a crucial problem in the implementation of HEVC. Accordingly, to enhance the coding efficiency of HEVC, fast algorithms have been proposed for reducing the HEVC complexity caused by the quadtree partition. These fast methods can be summarized into two categories: heuristic and learning-based schemes.

In heuristic methods, some fast CU decision algorithms have been developed to simplify the RDO process towards reducing HEVC complexity. For example, Cho et al. [4] developed a Bayesian decision rule with low complexity and full RD cost-based fast CU splitting and pruning algorithm. With regard to HEVC inter prediction, Shen et al. in [5] proposed a fast inter-mode decision scheme using inter-level and spatio-temporal correlations in which the prediction mode, motion vector and RD cost were found strongly correlated. To reduce the HEVC complexity, a look-ahead stage based fast CU partitioning and mode decision algorithm was proposed in [6]. Based on pyramid motion divergence, authors in [7] introduced a fast algorithm to split CUs at the HEVC inter coding.

On the other hand, the search of the optimal mode decision can be modeled as a classification problems. In this regard, researchers adopted learning-based methods in classifying CU mode decision in order to reduce the computational complexity. Shen et al. [8] proposed a CU early termination algorithm for each level of the quadtree CU partition based on weighted SVM. In addition, a fuzzy SVM-based fast CU decision method was proposed by Zhu et al. in [9] to improve the coding efficiency. To reduce the HEVC complexity with deep structure, in [10], authors developed a fast CU depth decision in HEVC using neural networks to predict split or non-split for inter and intra-mode. Reinforcement Learning (RL) and deep RL are also applied in video coding to learn a classification task, and to find the optimal CU mode decision. In this study, an end-to-end actor-critic RL based CU early termination algorithm for HEVC was developed to improve the coding complexity [11].

For video coding, the adjacent video frames exhibit similarity in video content, in which the similarity decreases along with temporal distance between two frames. Figure 1 shows the temporal correlation of CTU partition across HEVC video frames. In HEVC inter-mode, there are long and short-term dependencies of the CU partition across neighboring frames. It is in this context that we propose in this paper Long Short-Term Memory (LSTM) networks to study the temporal dependency of the CU partition across adjacent frames.

Based on our previous work presented in [12], our deep CNN structure has been proposed to predict the CU splitting in order to reduce the coding performance of inter-mode HEVC. A large-scale training database was established in [12]. Therefore, to predict the inter-mode CU partition, this paper proposes a CNN-LSTM-based learning approach where the aim is to reduce the HEVC complexity in terms of RD performance and encoding time.

Fig. 1. Temporal similarity of CTU partition across HEVC video frames.

The paper is organized as follows: Sect. 2 introduces the proposed method, which reduces the HEVC complexity at inter prediction. The evaluation results are shown in Sect. 3. Section 4 concludes this paper.

2 Proposed Scheme

2.1 Database for Inter-mode

A large-scale database for CU partition of the inter-mode HEVC was established, to train the proposed model, as shown in [12]. However, to construct the database, we selected 114 video sequences with various resolutions (from 352×240 to 2560×1600) [15, 16]. These sequences are divided into three subsets: 86 sequences for training, 10 sequences for validation, and 18 sequences for test. All sequences in our database were encoded by HEVC reference software using the Low Delay P (LDP) (using encoder_lowdelay_P_main.cfg) at four Quantization Parameters (QP) $\{22, 27, 32, 37\}$. Therefore, corresponding to different QPs and CU sizes (64×64 , 32×32 , and 16×16), 12 sub-databases were obtained under LDP configuration.

2.2 CNN-LSTM Network

According to the temporal correlation of the CU partition of adjacent frames, the proposed scheme is introduced in this section. The proposed LSTM network learns the long and short-term dependencies of the CU partition across frames. In our previous proposed method [12], all parameters of Deep CNN are trained over the residual CTU and the ground-truth splitting of the CTUs, then the extracted features $(FC_{1-l})_{l=1}^3$ of Deep CNN are the input of LSTM network at frame t . These features $(FC_{1-l})_{l=1}^3$ are extracted at the first fully connected layer of Deep CNN [12]. The proposed algorithm that combines CNN and LSTM is shown in Fig. 2.

Fig. 2. Framework of the proposed algorithm.

The structure of the LSTM is composed of three LSTM cells corresponding to three levels splitting of each CU. Specifically, $\tilde{F}_1(CU, t)$ at level 1 indicates whether the CU of size 64×64 is split to sub-CUs of size 32×32 . At level 2, $\{\tilde{F}_2(CU_{i,t})\}_{i=0}^3$ determines the splitting of CUs from 32×32 to 16×16 and $\{\tilde{F}_3(CU_{i,j}, t)\}_{i,j=0}^3$ denotes the partitioning labels of CUs from 16×16 to 8×8 . At each level, the LSTM cells are followed by a fully connected layer. In addition, the output features of the LSTM cells denote by $(FC'_l)_{l=1}^3$ at frame t . However, the next level LSTM cell is activated to make decisions on the next four CUs at the next level, when the CU of the current level is predicted to be split. Otherwise, the prediction on splitting the current CU is terminated early. Finally, the predicted CU partition of three levels is represented by the combination of $\tilde{F}_1(CU, t)$, $\{\tilde{F}_2(CU_{i,t})\}_{i=0}^3$ and $\{\tilde{F}_3(CU_{i,j}, t)\}_{i,j=0}^3$.

The LSTM model learns the long short-term dependency of CTU partition across frames when the CTU partition is predicted. In fact, the LSTM cell consists of three gates: the input gate $\mathbf{i}_l(t)$, the forget gate $\mathbf{f}_l(t)$ and the output gate $\mathbf{o}_l(t)$. In particular, $FC_{1-l}(t)$ denotes the input features of the LSTM cell at frame t and $FC'_l(t-1)$ is the output features of the LSTM cell of frame $t-1$ at level l . In the following, these three gates are presented by:

$$\begin{aligned}
\mathbf{i}_l(t) &= \sigma(\mathbf{W}_i \cdot [\mathbf{FC}_{1-l}(t), \mathbf{FC}'_l(t-1)] + \mathbf{b}_i) \\
\mathbf{o}_l(t) &= \sigma(\mathbf{W}_o \cdot [\mathbf{FC}_{1-l}(t), \mathbf{FC}'_l(t-1)] + \mathbf{b}_o) \\
\mathbf{f}_l(t) &= \sigma(\mathbf{W}_f \cdot [\mathbf{FC}_{1-l}(t), \mathbf{FC}'_l(t-1)] + \mathbf{b}_f)
\end{aligned} \tag{1}$$

where the sigmoid function denotes by $\sigma(\cdot)$. $\{\mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_f\}$ are the weights and $\{\mathbf{b}_i, \mathbf{b}_o, \mathbf{b}_f\}$ are the biases for three gates. At frame t , the state $\mathbf{c}_l(t)$ of the LSTM cell can be updated by:

$$\begin{aligned}
\mathbf{c}_l(t) &= \mathbf{i}_l(t) \odot \text{Tanh}(\mathbf{W}_c \odot [\mathbf{FC}_{1-l}(t), \mathbf{FC}'_l(t-1)] + \mathbf{b}_c) \\
&\quad + \mathbf{f}_l(t) \odot \mathbf{c}_l(t-1)
\end{aligned} \tag{2}$$

where $\odot$ signifies the element-wise multiplication. The output of the LSTM cell $\mathbf{FC}'_l(t)$ can be determined as follows:

$$\mathbf{FC}'_l(t) = \mathbf{o}_l(t) \odot \mathbf{c}_l(t) \tag{3}$$

In the training phase, the LSTM model was trained from the training set of the inter database, which minimizes the loss function between the ground truth and the prediction of CTU partition. Here, the cross entropy is adopted as the loss function. Then, 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 the loss function.

3 Experimental Results

This section introduces the experimental results to evaluate the encoding performance of our proposed approach. Our experiments were integrated in HEVC reference test model HM, which were tested on JCT-VC video sequences from Class A to Class E at two QPs $\{22, 37\}$ using the LDP configuration. In order to validate the performance of the proposed scheme, all simulations were carried out on windows 10 OS platform with Intel®core TM i7-3770 @ 3.4 GHz CPU and 16 GB RAM. We also use the NVIDIA GeForce GTX 480 GPU to dramatically improve speed during the training phase of the network model.

The Tensorflow-GPU deep learning framework was used in the training process. We first 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. Second, the learning rate was set to 0.01, changing every 2,000 iterations to train the LSTM model. Then, the LSTM length was set to $T=20$. Finally, the trained model can be used to predict the inter-mode CU partition for HEVC.

For the test, to further enhance the RD performance and the complexity reduction at inter-mode, the bi-threshold decision scheme was adopted at three levels. Note that the upper and lower thresholds at level l represents by $\{\gamma_l\}_{l=1}^3$ and $\{\bar{\gamma}_l\}_{l=1}^3$. The predicted CU partition probability is output by the LSTM model at different levels ($P_l(CU)$). Therefore, the CU decides to be split when

Table 1. Performance comparison between the proposed scheme and the state-of-the-art approaches

Class	Sequence	Approaches	ΔBR (%)	$\Delta PSNR(dB)$	ΔT (%)
A	Traffic	[12]	-0.458	-0.059	-56.66
		[13]	3.25	-0.081	41.8
		[14]	1.990	-0.052	-59.01
		[Proposed Scheme]	0.376	-0.101	-61.27
B	BasketballDrive	[12]	1.335	-0.031	-50.43
		[13]	2.01	-0.049	44.7
		[14]	2.268	-0.052	-53.92
		[Proposed Scheme]	1.528	-0.022	-52.05
C	PartyScene	[12]	-0.436	-0.076	-51.51
		[13]	3.12	-0.131	41.4
		[14]	1.011	-0.039	-48.27
		[Proposed Scheme]	0.494	-0.029	-58.48
D	BQSquare	[12]	-2.012	-0.128	-53.27
		[13]	4.15	-0.149	46.6
		[14]	0.770	-0.028	-48.85
		[Proposed Scheme]	0.219	-0.050	-59.52
E	Johnny	[12]	-0.181	-0.134	-64.43
		[13]	0.82	-0.052	48.7
		[14]	1.691	-0.038	-63.48
		[Proposed Scheme]	1.371	-0.028	-70.85
Average		[12]	-0.350	-0.085	-55.26
		[13]	2.67	-0.092	44.64
		[14]	1.546	-0.048	-54.70
		[Proposed Scheme]	0.797	-0.046	-60.43

$P_l(CU) > \gamma_l$; if $P_l(CU) < \bar{\gamma}_l$, the CU is not split. In this way, this considerably reduces the HEVC complexity by skipping the most redundant checking of RD cost.

The RD performance analysis is performed on the basis of the average PSNR ($\Delta PSNR$) gain and the average BitRate ($\Delta BitRate$) reduction. Additionally, the complexity reduction is the critical metric for the performance evaluation at HEVC inter-mode. Let ΔT the encoding time reduction. All performance metrics are written as:

$$\Delta PSNR = PSNR_P - PSNR_o \text{ (dB)} \quad (4)$$

$$\Delta BitRate = \frac{BR_P - BR_o}{BR_o} \times 100 \text{ (\%)} \quad (5)$$

Fig. 3. CU partition result of BasketballPass with $QP = 37$. frame = 20. LDP configuration.

$$\Delta T = \frac{T_P - T_o}{T_o} \times 100 (\%) \quad (6)$$

Table 1 demonstrates the performance comparison of the proposed scheme and three other state-of-the-arts schemes.

From this table, we can conclude that the proposed scheme is better in terms of computational complexity reduction than other state-of-the-arts schemes. Specifically, the time saving of our approach is 60.43% on average, which exceeds the 55.26% obtained by [12], the 44.64% achieved by [13], and 54.70% of [14], respectively. On the other hand, the proposed approach can reduce the PSNR performance by -0.046 dB, which is better than -0.085 dB of [12], -0.092 dB of [13] and -0.048 of [14]. Furthermore, the approaches [12] achieve better value of BitRate 0.350% than ours. Meanwhile, the proposed approach outperforms the existing approaches [13, 14] in terms of BitRate by 0.797%.

As shown in this table, the proposed approach can reduce the complexity reduction by 70.85% for class E video sequences, since these sequences have low motion activities and homogeneous regions, where the blocks CU partition is larger. From the overall performance assessment, our proposed fast scheme provides competitive HEVC coding efficiency tradeoffs compared to state-of-the-art approaches.

Figure 3 shows the comparison between the CU partition predicted by the proposed deep learning model and the ground truth partition when QP equals 37 under the LDP configuration.

4 Conclusion

This paper proposed a deep learning approach to predict the CU partition, which combines the CNN-LSTM network to reduce inter-mode HEVC complexity. To train the proposed model, the inter database was built. According to the temporal correlation of the CU partition of neighboring frames, we developed a

new LSTM architecture to learn the long and short-term dependencies of the CU partition across frames, instead of brute-force RDO search. In summary, the proposed scheme saves a significant encoding complexity compared to other state-of-the-art works.

References

1. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding. *IEEE Trans. Circuits Syst. Video Technol.* **22**, 1649–1668 (2012)
2. Khemiri, R., Kibeya, H., Sayadi, F.E., Bahri, N., Atri, M., Masmoudi, N.: Optimisation of HEVC motion estimation exploiting SAD and SSD GPU-based implementation. *IET Image Process.* **12**(2), 243–253 (2017)
3. Khemiri, R., et al.: Fast Motion Estimation's Configuration Using Diamond Pattern and ECU, CFM, and ESD, Modes for Reducing HEVC Computational Complexity. *IntechOpen, Digital Imaging Book*, London, pp. 1–17 (2019)
4. Cho, S., Kim, M.: Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding. *IEEE TCSVT* **23**, 1555–1564 (2013)
5. Shen, L., Zhang, Z., Liu, Z.: Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatio-temporal correlations. *IEEE Trans. Circuits Syst. Video Technol.* **24**, 1709–1722 (2014)
6. Cebrián-Márquez, G., Martínez, J.L., Cuenca, P.: Adaptive inter CU partitioning based on a look-ahead stage for HEVC. *Sig. Process. Image Commun.* **76**, 97–108 (2019)
7. Xiong, J., Li, H., Wu, Q., Meng, F.: A fast HEVC inter CU selection method based on pyramid motion divergence. *IEEE Trans. Multimed.* **16**, 559–564 (2014)
8. Shen, X., Yu, L.: CU splitting early termination based on weighted SVM. *EURASIP J. Image Video Process.* **4**, 1–11 (2013)
9. Zhu, L., Zhang, Y., Kwong, S., Wang, X., Zhao, T.: Fuzzy SVM based coding unit decision in HEVC. *IEEE Trans. Broadcast* **64**, 681–694 (2017)
10. Kim, K., Ro, W.: Fast CU depth decision for HEVC using neural networks. *IEEE Trans. Circuits Syst. Video Technol.* (2018). <https://doi.org/10.1109/TCSVT.2018.2839113>
11. Li, N., Zhang, Y., Zhu, L., Luo, W., Kwong, S.: Reinforcement learning based coding unit early termination algorithm for high efficiency video coding. *J. Visual Commun. Image R* **60**, 276–286 (2019)
12. Bouaafia, S., Khemiri, R., Sayadi, F.E., Atri, M.: Fast CU partition based machine learning approach for reducing HEVC complexity. *J. Real-Time Image Process.* **7**, 185–196 (2019)
13. Li, Y., Liu, Z., Ji, X., Wang, D.: CNN based CU partition mode decision algorithm for HEVC inter coding. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 993–997 (2018)
14. Xu, M., Li, T., Wang, Z., Deng, X., Yang, R., Guan, Z.: Reducing complexity of HEVC: a deep learning approach. *IEEE Trans. Image Process.* **27**(10), 5044–59 (2018)
15. Bossen, F.: Common test conditions and software reference configurations. Document JCTVC-L1100, Joint Collaborative Team on Video Coding (2013)
16. Xiph.org.: Xiph.org Video Test Media (2017). <https://media.xiph.org/video/derf>