J. Vis. Commun. Image R. 60 (2019) 276–286

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Reinforcement learning based coding unit early termination algorithm for high efficiency video coding ☆

Na Li [a], Yun Zhang [a,*], Linwei Zhu [a], Wenhan Luo [b], Sam Kwong [c,d]

[a] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[b] Tencent AI Lab, Shenzhen, China
[c] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, China
[d] City University of Hong Kong Shenzhen Research Institute, Shenzhen, China

ARTICLE INFO

Article history:
Received 6 October 2018
Revised 27 December 2018
Accepted 17 February 2019
Available online 20 February 2019

Keywords:
Coding tree unit
Early termination
High efficiency video coding
Markov decision processing
Actor-critic
Reinforcement learning

ABSTRACT

In this paper, we propose a Reinforcement Learning (RL) based Coding Unit (CU) early termination algorithm for High Efficiency Video Coding (HEVC). RL is utilized to learn a CU early termination classifier independent of depths for low complexity video coding. Firstly, we model the process of CU decision as a Markov Decision Process (MDP) according to the Markov property of CU decision. Secondly, based on the MDP, a CU early termination classifier independent of depths is learned from trajectories of CU decision across different depths with the end-to-end actor-critic RL algorithm. Finally, a CU decision early termination algorithm is introduced with the learned classifier, so as to reduce computational complexity of CU decision. We implement the proposed scheme with different neural network structures. Two different neural network structures are utilized in the implementation of RL based video encoder, which are evaluated to reduce video coding complexity by 34.34% and 43.33%. With regard to Bjøntegaard delta peak signal-to-noise ratio and Bjøntegaard delta bit rate, the results are −0.033 dB and 0.85%, −0.099 dB and 2.56% respectively on average under low delay B main configuration, when compared with the HEVC test model version 16.5.

1. Introduction

High Efficiency Video Coding (HEVC) [1] is the ongoing video coding standard developed by the Joint Collaborative Team on Video Coding (JCT-VC), which makes a big step on compression efficiency to reduce half bit rate of H.264/AVC while maintaining the same video quality. The advantage of HEVC is the compression capability of supporting 4K Ultra High Definition (UHD) of 3840×2160 or 4096×2160 resolutions, and up to 8K UHD of 8192×4320 resolution. However, the deployment complexity of HEVC restricts its worldwide application in many emerging real-time applications, such as live video broadcasting, real-time video chatting, as well as applications on mobile platforms with limited power and computing resources, *e.g.*, smart phones and drones. Reducing the encoder's computational complexity with acceptable coding performance losses attracts great attention in the academical and industrial society.

Recursive Coding Tree Unit (CTU) partition technology plays a key role in improving coding efficiency of HEVC in comparison with H.264/AVC. Each frame of the video sequences is partitioned into blocks of different sizes, noted as Coding Units (CUs). Quad-tree is adopted as the partition structure towards flexible CU partition. However, brute-force searching of the quad-tree based on Rate Distortion Optimization (RDO) for the optimal CU combination is time consuming. Thus, fast CU decision algorithms are proposed to reduce Rate Distortion (RD) cost comparison for video coding parameter decisions meanwhile guarantee compression quality, such as predictions of CUs, Prediction Units (PUs) and Transform Units (TUs). CU decision early termination as one group of fast CU decision algorithms, has been studied for its simplicity and efficiency.

The state-of-the-art researches on fast CU decisions focus on optimizing the CU decision by dividing and controlling the recursive RD cost comparison process for each CU depth separately. The encoder performance for each depth is separately optimized before being implemented to optimize the overall CU performance. Existing methods on fast CU decision can be divided into two categories, i.e., statistical methods and machine learning based methods.

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail addresses: na.li1@siat.ac.cn (N. Li), yun.zhang@siat.ac.cn (Y. Zhang), lw.zhu@siat.ac.cn (L. Zhu), whluo.china@gmail.com (W. Luo), cssamk@cityu.edu.hk (S. Kwong).

https://doi.org/10.1016/j.jvcir.2019.02.021
1047-3203/

To statistical methods, statistical relationships among neighboring CUs are considered [2,3]. Jung et al. [4] proposed a fast mode decision method based on ordering modes adaptively for CU decisions. Kim et al. [5] adjusted the threshold for early terminating the RD cost comparison so as to reduce decision complexity of partitioning current CU, which considers both the spatial correlation of CU depths and the probability of the SKIP mode among neighboring CUs. Ahn et al. [6] introduced a method on combining CU early termination and fast mode decision to reduce encoder's complexity of HEVC. Shen et al. [7] introduced an adaptive CU early termination method based on the texture homogeneity.

To machine learning based methods, Support Vector Machine (SVM) [8–12], Bayesian Theory [13–15], Markov Model [16,17], Decision Tree (DT) [18,19], and Neural Network [20–22] were adopted to model CU decision as classifiers, which were learned from CU decision samples for developing fast CU decision algorithms. Zhang et al. [8,9] proposed a three-output joint classifier consisting of multiple binary SVM classifiers with different parameters. Zhu et al. [10] proposed to model the recursive CU decision as a binary SVM classifier, whereas variable PU modes selection was modeled with multi-class SVM. Zupancic et al. [13] proposed a novel scheme comprising two types of block testing orders, including the normal CU visiting order and the reverse CU visiting order. Naïve Bayes algorithm is implemented for reverse CU decision optimization. Shen et al. [14] proposed to learn the CU decision scheme with Bayesian decision based on two-class formulation of CU decision prediction. Kim et al. [15] jointly utilized on-line and off-line learning to avoid unnecessary RD cost comparison based on the Bayesian decision rule. Markov Random Field (MRF) was applied in [16] to incorporate the features and the neighboring information, so as to reduce CU decision complexity for coding inter frames. Chen et al. [17] smoothed the copying-based prediction through theoretical analyses of the optimal weights of filters with first-order Gaussian Markov model. Decision trees were built and implemented to reduce the encoder decision complexity through early terminating the recursive RD cost comparison process in [18]. For intra-frame coding extension of screen content in HEVC, Duanmu et al. [19] designed decision tree classifiers with chosen features to distinguish different types of blocks. Taking into account the prediction accuracy, encoder memory consumption and model simplicity, fast CU decision prediction was modeled by Duanmu et al. [20] as a two layer neuron network classifier for screen content compression. Liu et al. [21] utilized Convolutional Neural Network (CNN) to exploit topology information for CU decision. To reduce the encoder complexity for HEVC with deep structure, Xu et al. [22] proposed a hierarchical CU decision map to predict CTU partition patterns with one pass of convolution computation.

RL and deep RL are also applied in video coding control and optimization separately [23,24]. Helle et al. [23] proposed to learn a set of binary classifiers for nodes in the tree. RL is adopted to learn binary classifier, which optimized CU decision separately for different depths. To utilize RL for video encoder control, CU decision was set as state-action pair in [24]. However, the joint prediction accuracy of CUs at different depths is not the same as the overall prediction accuracy of CUs across different depths.

This motivates us to learn a CU decision algorithm independent of depths regarding to the cumulative performance of the overall long-term CU decision trajectories. Primarily, we make the following contributions.

(1) We model the RD cost comparison of CU decision as a MDP according to the Markov property across CU depths, towards low complexity video coding.
(2) We learn a CU early termination classifier with the end-to-end actor-critic RL algorithm from trajectories of CU decision, which is independent of CU depths and approximated with one hidden layer neural network.
(3) A CU early termination algorithm which requires negligible computational computational overhead in comparison to the whole CTU partition process is derived from the CU early termination classifier.

The paper is organized as follows. Section 2 presents the motivation and analyses. The proposed framework of encoding with RL based CU early termination algorithms is introduced in Section 3. A CU early termination classifier independent of depths is learned with RL based on the MDP for CU early termination in Section 4. The CU early termination algorithm derived from the CU early termination classifier is proposed for low complexity video coding in Section 5. Experimental results are presented in Section 6 to demonstrate the efficiency of the proposed RL based CU early termination algorithm. Section 7 concludes this paper.

2. Motivation and analyses

Fig. 1 presents a full quad-tree with hierarchical coding block partition structure corresponding to CU decision for one CTU in HEVC. CU of larger size at lower depth, *e.g.*, CU of size 64×64, is grouped to D0. CUs at the full quad-tree are distinguished by the proposed L_{ID} in Fig. 1 from different depths and locations, where $L_{ID} \in \{0, \ldots, 84\}$. CU at D0 is recursively partitioned into a subset of CUs of size ranging from 64×64 to 8×8. The optimal combination of CUs for one CTU partition is selected through brute-force searching of CU from different depths. There are $(2^4+1)^4+1 =$ 83,522 combinations of CUs to be checked before obtaining the optimal combination of CUs.

To lower the complexity of CU partition, CU early termination is discussed in this paper, which is to determine whether the rest of RD cost comparison is eliminated or not for CU partition given the current CU. The potential Time Saving (*TS*) ratio of CU early termination is statistically analyzed, which is defined as

$$TS = \sum_{i=0}^{n}(T_i^{RDO} - T_i^{\mu}) / \sum_{i}^{n} T_i^{RDO} \times 100\%, \tag{1}$$

where n is the number of CTUs. T_i^{RDO} is the running time consumption of partitioning one CTU indexed i according to RD cost comparison. T_i^{μ} is the time consumption of partitioning CTU indexed with i through early terminating RD cost comparison for specific CU decision.

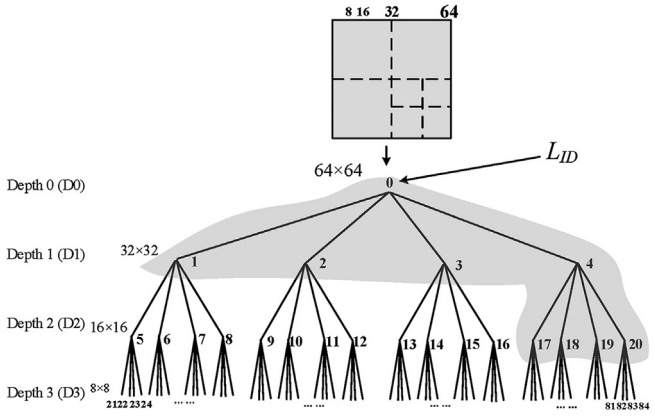

Fig. 1. Quad-tree structure of partitioning CTU into CUs. Each quad-tree has 85 nodes corresponding to candidate CUs indexed by L_{ID} in $\{0, 1, 2, \ldots, 84\}$.

Table 1
Ratio of different depths and potential time saving of applying early termination to D0, D1, D2 and D3.[unit: %].

Sequence	QP	CU depth distribution				Potential time saving
		D0	D1	D2	D3	
BasketballPass	22	40.4	20.4	20.0	19.2	36.2
	27	43.1	22.0	20.6	14.3	38.7
416 × 240	32	46.4	25.7	19.5	8.5	41.8
	37	55.7	26.2	14.1	3.9	49.5
BQMall	22	23.3	27.4	30.0	19.2	23.4
	27	39.2	25.5	22.7	12.6	36.0
832 × 480	32	49.3	24.5	18.5	7.80	44.0
	37	58.6	23.4	14.0	4.00	51.4
FourPeople	22	60.8	23.4	12.5	3.40	53.2
	27	76.2	15.5	6.60	1.70	64.6
1280 × 720	22	83.4	11.6	4.20	0.80	69.9
	27	89.0	8.20	2.40	0.30	74.0
Tennis	22	12.2	48.5	31.2	8.20	17.2
	27	25.5	48.4	21.8	4.30	28.0
1920 × 1080	32	39.4	44.9	13.9	1.80	38.7
	37	55.0	36.6	7.70	0.60	50.2
PeopleOnStreet	22	7.50	23.4	39.8	29.3	10.1
	27	14.5	24.4	38.6	22.5	15.9
2560 × 1600	32	19.2	30.6	35.9	14.3	20.5
	37	25.9	36.4	29.8	7.80	26.8
Average		43.2	27.4	20.2	9.20	**39.5**

The up-bound of *TS*, noted as $\hat{TS}$ is estimated as the time saving potential through early terminating the rest of RD cost comparisons after reaching the optimal CU at depths D0, D1 and D2, as shown in Table 1. Derived from the ratio of CU in the optimal CU and the estimated encoding complexity of CU at D0, D1 and D2, $\hat{TS}(N)$ of applying early termination for low complexity CU decision at different depths are calculated as

$$\hat{TS}(N) = \sum_{d=1}^{\log_2 N/4-1} 2^{2d} \cdot T_{N/2^d}, \tag{2}$$

where $N > 4$ and CU size is $N \times N$, and $d \in \{0, 1, 2, 3\}$ denotes the depth in {D0, D1, D2, D3}. The overall $\hat{TS}(64)$ of applying early termination to CU decision at {D0, D1, D2} is estimated as 39.5% in Table 1. Encoding complexity T of CU decision at depth in {D0, D1, D2, D3} are estimated from selected video sequences, which are scaled as $T_{64\times64} : T_{32\times32} : T_{16\times16} : T_{8\times8} = 45.32{:}16.16{:}4.70{:}1$. Video sequences covering different motions, texture information and resolutions are coded with Quantization Parameters (QPs) in $\{22, 27, 32, 37\}$ by HM version 16.5, including {"BasketballPass", "BQMall", "Fourpeople", "Tennis", "PeopleOnStreet"}. Low delay B main configuration is adopted in the coding experiments.

We propose to develop an early termination algorithm to achieve *TS*. The early termination of CU at lower depths is essential for time saving of CU decision. In Table 1, overall 70.6% of frames are coded with CU at D0 and D1. The efficiency up-bound of CU early termination algorithm in terms of *TS* is supposed to be close to 39.5% with negligible coding performance degradation, according to Table 1. CU early termination algorithms with computational complexity larger than 39.5% is assumed to lose more coding performance.

3. Framework of the proposed RL based video encoder

In the original HM encoder, videos are compressed into bitstream through brute-force searching of the optimal CU combination. To reduce the computational complexity of the CU decision, we propose to learn a CU early termination classifier independent of depths with RL algorithm. RL algorithm is adopted to solve CU decision problem as a MDP. Almost all decision processes with Markov property can be formulated as a MDP. CU decisions satisfy the Markov property [25] that the future is independent of the past given the present. Thus, we model the CU decision problem as a MDP.

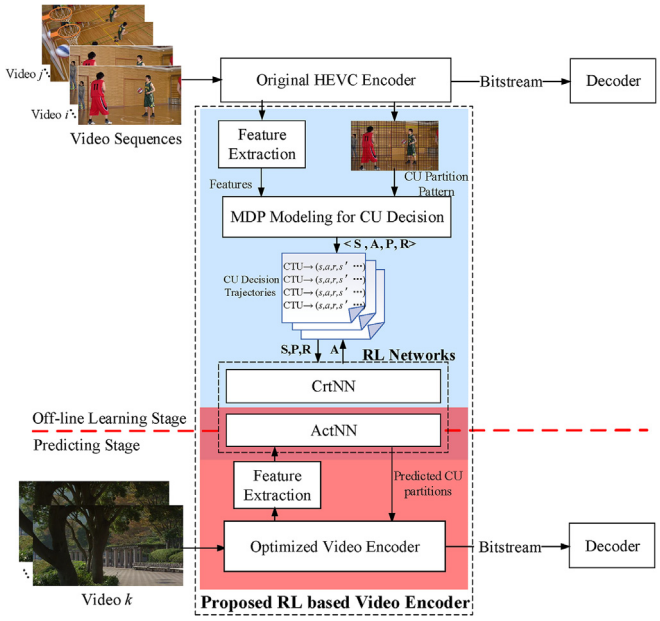

Fig. 2. Framework of the proposed RL based video encoder.

A framework of optimizing the video encoder with the proposed CU early termination classifier is presented in Fig. 2, which comprises of two stages, i.e., off-line learning stage and predicting stage. At the stage of off-line learning, we propose to learn a CU early termination classifier with RL algorithm from trajectories of CU decision. Optimal CU partition patterns and features are extracted from coding process based on the MDP modeling of CU decision, which is defined as tuples $\langle \mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R} \rangle$. CU is the state *s*. The state space **S** is represented with the vector of features for CUs, which are extracted from the original encoding process. The CU decision is the action *a* taken at the current state, including "split" and "unsplit". **A** indicates the optimal CU decision. The accuracy of taking action *a* at state *s* is the reward *r*. Many algorithms are developed for optimizing CU decisions to determine whether the rest of RD cost comparison can be avoided or not for current CU decision. CU decision classifier referred to policy **P** is learned as ActNN from trajectories of $\langle s_i, a_i, r_i \rangle$ with RL algorithm. Neural Network is adopted to approximate both reward **R** and policy **P**, noted as Critic Neural Network (CrtNN) and Actor Neural Network (ActNN) respectively. An end-to-end actor-critic RL algorithm is introduced to learn CrtNN and ActNN.

At the stage of prediction, ActNN is to reduce the complexity of RD cost comparison. A CU early termination algorithm is constructed to utilize the ActNN to predict CU decision. Features of CUs for the ActNN are extracted from the original video encoder before triggering the ActNN for CU early termination. The weight of ActNN is selected with metrics on CU decision classification performance at the stage of off-line learning.

4. Problem formulation and reinforcement learning for CU decision

4.1. MDP modeling for CU decision

The problem of CU early termination satisfies the Markov property [25] that the future is independent of the past given the pre-

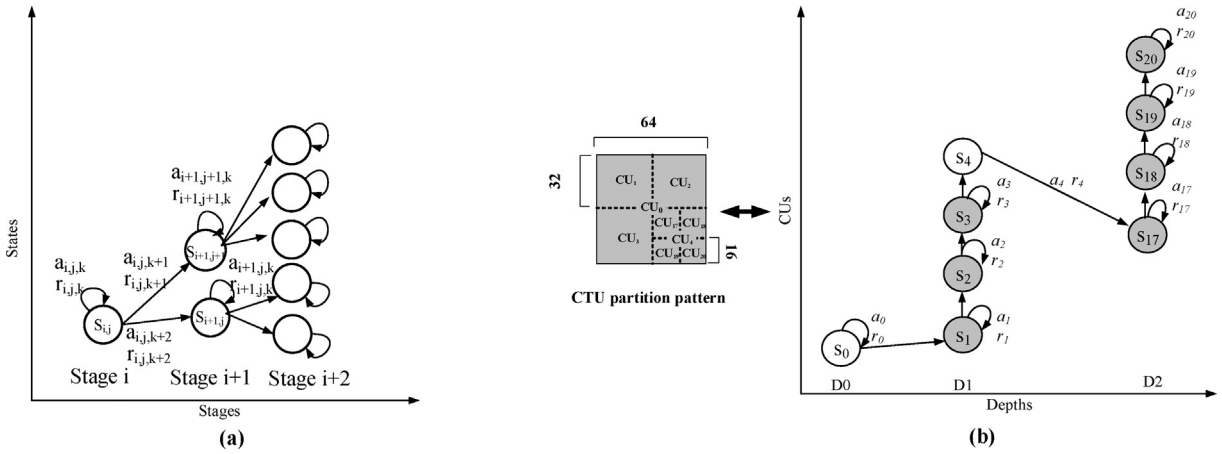

Fig. 3. MDP modeling for RD cost comparison of CU decision. (a) General MDP for CU decision; (b) MDP for CU early termination.

sent. CU early termination is modeled as a MDP, which is derived from MDP of CU decision. The video coding can be achieved through solving the constrained optimization problem [26]

$$\min \sum_{i=1}^{N} D_{i,j}, \quad s.t. \sum_{i=1}^{N} R_{i,j} < R_c, \tag{3}$$

where CU decision indexed with j is determined with the RD cost comparison on N steps of CU decision and $j = L_{ID}$. Each CU decision associates with one pair of bit and distortion $\{R_{i,j},\ D_{i,j}\}$. R_c is the target bit. Lagrange Multiplier (LM) [27,28] and Dynamic Programming [29] are applied to the RD cost comparison process. However, Lagrange algorithm fails to select $\{R_{i,j},\ D_{i,j}\}$ within convex envelop which is produced by introducing λ. Programming is more capable of producing better decisions than Lagrange algorithm without the constraints on the convex envelop. Programming can get the optimal solution $\{R^*,\ D^*\}$ of Eq. (3) through producing a grid of all possible pairs $\{R_{i,j},\ D_{i,j}\}$. The complexity of programming increases with the increase of nodes in the grid, which limits the application of programming in video coding.

We introduce RL to learn classifier to early terminate RD cost comparison for CU decision. RD cost comparisons over the whole grid of $\{R_{i,j},\ D_{i,j}\}$ are pruned through selecting a set of $\{R_{i,j},\ D_{i,j}\}$ to maximize the reward, so as to generate optimal CTU partition patterns. The optimal CTU partition pattern generated from RD cost comparison is associated with a trajectory of $\{R_{i,j},\ D_{i,j}\}$. A MDP is a tuple $\langle \mathbf{S},\ \mathbf{A},\ \mathbf{P},\ \mathbf{R}\rangle$, where $\mathbf{S}$ denotes the state space, $\mathbf{A}$ the action space, $\mathbf{T}: \mathbf{S}\times\mathbf{A}\times\mathbf{S}\to[0,\infty)$ the density function of state transition probability and $\mathbf{R}: \mathbf{S}\times\mathbf{A}\times\mathbf{S}\to\mathbb{R}$ the reward function. A general MDP of RD cost comparison is shown in Fig. 3(a). $S_{i,j}$ is the corresponding states of $\langle R_{i,j}, D_{i,j}\rangle$ categorized as CU decision at different depths. Transition from $S_{i,j}$ to $S_{i+1,j}$ stands for the transition from $\langle R_{i,j}, D_{i,j}\rangle$ to $\langle R_{i+1,j}, D_{i+1,j}\rangle$.

The MDP of CU early termination is derived from the general MDP through specifying the next state a_{t+1} and the reward r_t of taking actions a_t at states s_t. State transitions of CU decision are associated with the transition between nodes in the grid. States associated to CU decision for one CTU partition pattern are shaded with gray. The corresponded trajectory of tuples $\langle s_t, a_t, r_t\rangle$ is derived to represent the process of CU decision shown in Fig. 3(b) as

$$\{s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, s_3, a_3, r_3, s_4, a_4, r_4, s_{17}, a_{17}, r_{17}, s_{18}, a_{18}, r_{18}, s_{19}, a_{19}, r_{19}, s_{20}, a_{20}, r_{20}\},$$

where t is the integer index of states. Transition from current state s_t to the next state s'_t is triggered by executing action a_t according to $s'_t \sim \mathrm{T}(\cdot|s_t, a_t)$. The transition between states of CU decisions derived from the same parent CU at lower depth is triggered according to the standard quad-tree traversing order without indicating action a_t nor reward r_t. The MDP of CU early termination explored in this paper is specified, where action $a_t \in \{split, unsplit\}$ and the state transition is constrained under the quad-tree traversing order. In Fig. 3(b), only actions a_0 and a_4 of states s_0 and s_4 are *split*. Early termination decision of CU at D3 is unnecessary regarding to the negligible *TS* potential at D3. States of CU at different depths are grouped into different stages. Reward r_t of taking action *split* or *unsplit* at state s_t is assigned with function on the optimal CU decision and the CU decision prediction. Reward r_t is collected according to long-delay return after checking lower depths of current CU, quantity factorizations for which is the binary classification accuracy of CU decision. RL can be adopted to learn a CU early termination classifier independent of depths.

4.2. RL networks

A CU early termination classifier independent of depths is learned as the policy in the MDP. The schematic diagram of applying the actor-critic RL to learn CU decision classifier is shown in Fig. 4, which is inherited from [30].

The proposed actor-critic RL scheme is adopted to tackle the fast CU decision as a class of decision learning and control problem. The reward in the RL scheme is designed for a step of CU decision. In general MDP, reward is one of the key component that determines the performance of the decision learning and control system.

As it is difficult to learn tuples $\langle s_t, a_t, r_t\rangle$ for CUs individually. We introduce to approximate CU early termination classification and

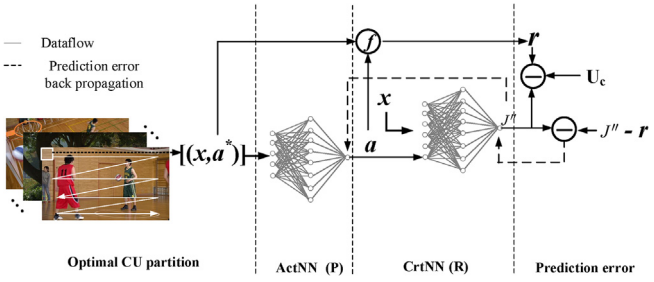

Fig. 4. Schematic diagram for implementing the end-to-end actor-critic RL algorithm.

the corresponding reward with neural network, as it shown in Fig. 4. ActNN is a CU early termination classifier and CrtNN is a reward function. ActNN for CU early termination prediction is updated with prediction performance metrics. Reward r towards critic metrics is noted as reward function between optimal CU decision and predicted CU decision. In this work, the CU early termination classifier ActNN is adjusted according to prediction performance estimation of CrtNN. The CrtNN's estimation of the expected return allows the actor to update with gradients of low variance. ActNN and CrtNN are learned from trajectories of CU decisions in terms of $\langle s_t, a_t, r_t \rangle$ for CU early termination. The diagram in Fig. 4 is introduced by learning ActNN and CrtNN from prediction errors with regard to the CU decision classification performance and long-delay reward of the CU early termination.

We utilize neural network with one hidden layer for CU early termination classification, so as to achieve better prediction accuracy with less classification complexity. Feature representation $x(t)$ of CUs is the input of ActNN. Whereas, the input of CrtNN is produced by appending the action signal $a(t)$ to the feature vector x of current CU. Function approximation of ActNN noted as function $a_t = F(\mu(s|\theta^{\mu}))$ of the state s_t for CU_t is learned by the proposed end-to-end actor-critic RL algorithm.

Hidden units for ActNN and CrtNN are the same. The number of hidden neurons W is initialized based on the empirical equation

$$W = \sqrt{p+q} + \alpha, \tag{4}$$

where p is the number of units at the input layer, q the number of units at the output layer, $\alpha \in \{1, \ldots, 10\}$. Generally, the increasing number of hidden neurons will enhance the representation capability of neural networks. The gain of larger number of neurons is associated with the feature selection in the case of the proposed ActNN and CrtNN. The rising of neuron numbers can bring up extra complexity of feature extraction and neural network computation to the proposed CU early termination algorithm. Therefore, different numbers of neurons play different roles in balancing the CU decision precision and the computation complexity. Under the constrain of the overhead of computational complexity, we experimentally evaluated ActNN and CrtNN with different number of neurons in Section 6. In the future, the neuron number is supposed to be adjusted considering feature selections.

Generally, in Table 2, we further compare the proposed single-layer Neural Network (NN) with Deep Neural Network (DNN) structure for CU decision prediction from various aspects. Handcraft features are required by single-layer NN. Features in the DNN network structure can be handcraft or data driven. The features difficult to be defined by humans can be learned from large data. However, the data driven features can hardly represent the spatial and temporal correlation among CUs when only pixels of the current CU was used as input. Besides, the architecture DNN is much more complex than the single-layer NN, which enhances the capabilities of learning and representation power for highly complicated decision problem. However, it also leads to higher training complexity and requires large amount of training samples. Given that the number of hidden neurons at layer j is W_j, the dimensions of input and convolution kernel are p and C_j, the prediction complexity of the single-layer NN is $p \times W_0 + W_0$. Whereas, the prediction complexity of DNN is $p \times C_0 \times C_0 \times W_0 + \sum_j W_{j-1} \times C_j \times C_j \times W_j$, which is relatively higher than that of the single-layer NN. Overall, the DNN is suitable for complex decision problem with higher computational cost while single-layer NN is simpler and with much lower complexity. In order to reduce the complexity overhead of the proposed RL based CU decision algorithm, we adopt the single-layer NN in the CU early termination decision.

4.2.1. End-to-end actor-critic RL algorithm

Algorithm 1 is proposed to learn the ActNN from trajectories of CU decision extracted from limited CTU partition patterns, which can be friendly extended for live communication applications. The proposed RL based CU early termination is presented according to the diagram in Fig. 4. The input is a set of pair on feature vectors and corresponding optimal CU decisions extracted from HM encoder. The return of the proposed actor-critic RL algorithm is ActNN function approximation noted as $\mu(s|\theta^{\mu})$ and the CrtNN function approximation $Q(s, a|\theta^{Q})$. θ^{μ} is the weight of ActNN. $Q(s, a|\theta^{Q})$ is the state-action value function. We address the training of CU early termination classifier in terms of optimizing an efficiency measure J, which is the function approximation noted as CrtNN [30].

The weight of CrtNN is updated by minimizing the prediction errors as follows

$$L_c = \frac{1}{2}(J - [J'' - r])^2, \tag{5}$$

where return prediction history J'' is expected to be close to the current J regarding the return value of each CU decision. The best return value of the CU decision can be set as $r^* = 0$. In this work, the end-to-end actor-critic RL algorithm is designed. The weight of ArtNN is updated towards the optimal critic U_c, say $U_c = 0$. The objective function to be minimized for updating the weight of ActNN is

$$L_a = \frac{1}{2}(J - U_c)^2, \tag{6}$$

where J is the long-delay return value of CU decision reward r. A general future accumulated efficiency-to-go return at step t is given by

$$v_t = r_{t+1} + r_{t+2} + \ldots = \sum_{k=1}^{\infty} r_{t+k}, \tag{7}$$

where the reward r_t provides a method to evaluate and guide the learning of ActNN towards to the optimal CU decision combination. Reward function $r = f(a^*, a)$ regarding to the optimal CU decision a^* and a of the CU decision prediction is controlled according to MDP for CU decision. For the convenience of discussion, reward of the ActNN is assigned as simple as "−1" or "0" in this work

$$r_t = f(a, a^*) = \begin{cases} -1 & a_t \neq a_t^* \\ 0 & otherwise \end{cases}, \tag{8}$$

Table 2
Property comparison between the proposed single-layer Neural Network and Deep Neural Network.

Item	Single-layer Neural Network	Deep Neural Network
Feature	Handcraft	Data driven
Architecture	Single layer	Multiple layer
Training complexity	Lower	Much higher
Training samples	Small	Large
Prediction complexity	$p \times W_0 + W_0$	$p \times C_0 \times C_0 \times W_0 + \sum_j W_{j-1} \times C_j \times C_j \times W_j$

where the predicted CU decision a is the same as the optimal CU decision a^*, *i.e.*, with minimum classification error of CU decision, the CU decision classification is assigned with a reward $r = 0$, otherwise $r = -1$, as shown in Eq. (8).

Based on the initialization of the number of hidden neurons computed as empirical equation, we experimentally adjust the scale of hidden units for ActNN and CrtNN. Both ActNN and CrtNN are configured as the nonlinear multilayer feed-forward network with one hidden layer that comprises the same number of neurons. Min-max normalization is applied for preprocessing the input vector at stages of off-line learning and predicting. The ActNN and CrtNN are adapted according to the chain rule in [30]. The discount factor is set as 0.99. Learning rate l_a increases from 0.001 to 0.25 with the step size 0.01 for every 10 steps. Learning rate l_c increases from 0.0001 to 0.25 with step size 0.01 for every 10 steps. The number of times that all CTUs in $\mathbb{S}$ are utilized to update the weights is noted by the epoch, which is under constraint of Z. We achieve a moderate CU early termination classifier with limited samples of CTU partition, which can be friendly extended to live communication applications. The number of CUs utilized at the off-line learning stage of the ActNN is relatively small.

4.3. Trajectory sampling and training strategies

The CU decision trajectories are derived from the HM 16.5 and extended with exploitation. For training, trajectories of CU decision are filtered out from sequences with different resolutions, texture and motions. In order to cover sequences with different bit rates, top 50 frames of four selected test video sequences are encoded with QPs in {22, 27, 32, 37}. CTU are selected from sequences "BasketballDrive", "FourPeople", "BQMall", "BQSquare" with resolution "1920 × 1080", "1280 × 720", "832 × 480", "416 × 240". Frames are selected from sequences with ratio of 1:4, 1:3, 1:2 and 1:1 individually, so as to balance the sampling among different sequences. A training set with CU decision trajectories of 3188 CTUs is produced, where trajectories of CU decision for each CTU is of size 83,522. By adopting the exploration and exploitation mechanism of RL, 3188 × 83,522 trajectories of CU decision utilized to train the networks are self-adapted generated, which are collected to provide sufficient trajectory samples for training. Samples of trajectories in terms of $\langle s_t, a_t, r_t \rangle$ are collected regarding to MDP of CU early termination.

The feature vector of current CU in this paper is extracted statistically from the HM 16.5 which is related to SKIP, INTER mode, INTRA mode, including the depth, RD cost of neighboring CUs, the CU distortion and SKIP flag of current CU, as well as the proposed L_{ID} of current CU. We drop half of CTUs partitioned with CU 64 × 64 to balance the CTU partition pattern distribution. Trajectories of CU decision in the training set are shuffled regarding to CTUs before training, so as to increase content structure invariance among CU decision trajectories, which is proved to be beneficial for learning the ActNN efficiently.

At the stage of validation, sequences "BasketballDrill", "ParkScene", with resolution "832 × 480", "1920 × 1080" are encoded for collecting CU decision trajectories under the same configuration of training.

4.4. Parameter setting and validation

The partition pattern of CTU is recognized as a combination of CUs across different depths. Based on MDP, reward sampling for trajectories of CU decision is taken in bottom to top order. Given the state s_t and action a_t, reward r_t of taking state transition from s_t to s_t' is collected from bottom to top along the quad-tree with Eq. (8). Afterwards, the CU early termination classifier ActNN is learned from trajectories of CU decision.

The ActNN is selected according to its classification performance on the validation set. Performance of applying the CU early termination classifier ActNN to predict the CU decision is evaluated with two metrics at the level of CTU and CU respectively. Let $\mathbf{I}$ be the partition matrix for one CTU, each element of which indicates the CU of size 4 × 4. Towards the number of CTUs that are partitioned fully correct with the proposed CU early termination algorithm, the classification precision H of the proposed CU early termination algorithm is defined as

$$H = \frac{M_{\mathbf{I}^*,\mathbf{I}}}{n}, \tag{9}$$

where $M_{\mathbf{I}^*,\mathbf{I}}$ is the number of CTU partitioned by the proposed CU early termination algorithm with the same partition matrix $\mathbf{I}$ output as the optimal CTU partition matrix $\mathbf{I}^*$. n is the number of CTU. O is the overlap between the CTU partition matrix output of the proposed CU early termination algorithm and the optimal CTU partition matrix, which is defined as

$$O = \frac{\sum_n \sum \mathbf{I}_i^* \cap \mathbf{I}_i}{n \cdot K}, \tag{10}$$

where $K = 256$ is the number of elements in the partition matrix $\mathbf{I}$, each of which corresponds to a CU of size 4 × 4 for the CTU.

Figs. 5 and 6 present H and O of applying ActNN to video sequences for validation. In Fig. 5, Q of ActNN can reach stability around 0.65 through exploiting and exploring 3110 steps of CU decision. After 3110 steps, the update of weights for the ActNN can rarely and hardly improve the H of the ActNN. O for ActNN with different numbers of CU decision steps is shown in Fig. 6. O of the selected ActNN achieves the relatively best value as 0.88. The fluctuation of H and O illustrates the weight adjustment towards maximum return of the ActNN. The experimental analyses indicate the selected ActNN can achieve relatively high performance on H and O with 3110 steps of CU decision. We set the ActNN with H and O as 0.65 and 0.88 on the validation set. The complexity reduction performance of the selected ActNN in HM 16.5 is illustrated in comparison to the state-of-the-art CU decision algorithms.

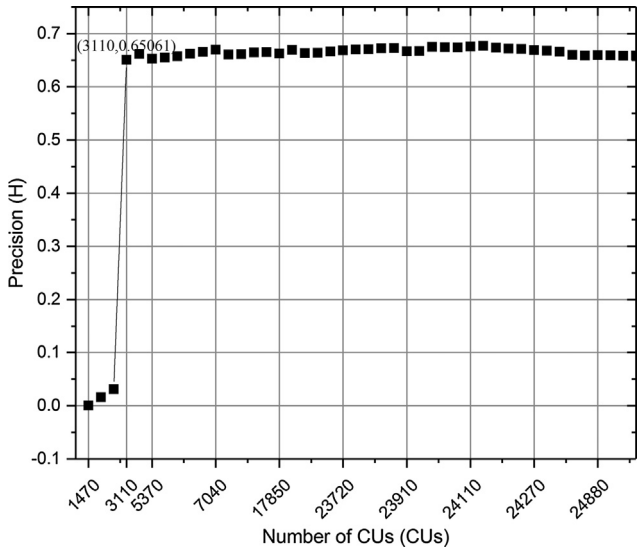

Fig. 5. Precision of CTU partition prediction in terms of CTU.

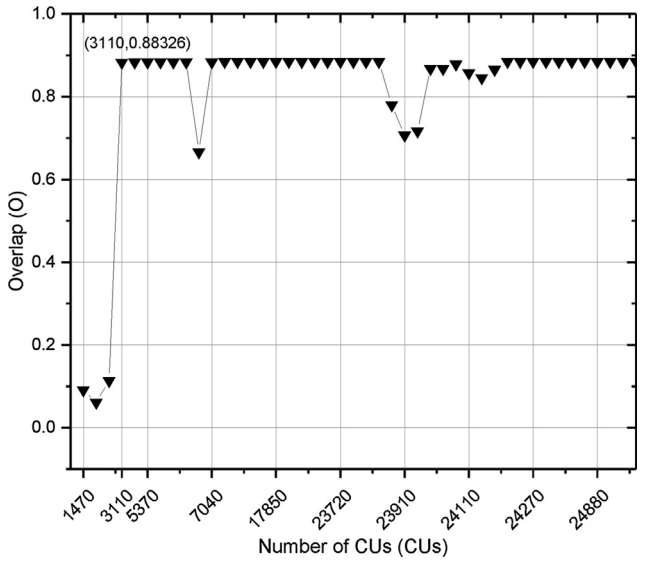

Fig. 6. Overlap between CTU partition prediction and the best CTU partition.

5. RL based video coding optimization

A CU early termination classifier ActNN learned from trajectories of CU decision across depths is utilized to reduce the computational complexity of CU decision independent of depths.

5.1. Feature selection

In the field of low complexity CU decision, features are selected regarding to both feature extraction overhead and classification accuracy [31]. Zhu et al. [31] presented 24 features for CU decision through surveying the state-of-the-art studies.

To the RL based CU early termination, features of states noted as $\mathbf{x}_t = \{x_i\}$ are introduced by combining coding information and CU index L_{ID}.

(1) $x_{NB\_CU\_Depth}$ is the average depth of CU at the left and above side of the current CU. For instance, the average depth of the current CU with size 16×16 is 2 when 16 of 4×4 units labeled depth as 2.
(2) $x_{NB\_CU\_RDcost}$ is the average RD cost of the left and above CUs of the current CU.
(3) $x_{CU\_SKIPFlag}$ is the output of SKIP mode checking to indicate the flag of skipping.
(4) $x_{CU\_Distortion}$ is the bypass total distortion after checking the SKIP mode, which indicates the video texture.
(5) $x_{CU\_Location}$ is the CU index L_{ID} at the quad-tree.

Features are selected for ActNN with different numbers of neurons, according to prediction accuracy and feature extraction complexity. In general, CU decision discrimination prefers features of high dimension to computational consumption.

The increasing number of features is supposed to bring more complexity overhead for not only feature extraction, but also the application of the CU early termination classifier. For neural network based CU early termination classifiers, the structure complexity of the neural network increases with the dimension of the feature vector. According to the empirical relationship between the input and the number of hidden neurons, the input dimension are selected for ActNN with different scales of neurons. Features from (2) to (4) is selected based on the feature analysis in [31] in consideration of Pearson Correlation Coefficient and Cross Validation accuracy.

Algorithm 1. Actor-critic RL for CU early termination

Input: $\mathbb{S}$= {< s_t, a_t, r_t >}
Output: Weights θ^{Q} and θ^{μ} for ActNN and CrtNN.
Init l_a = 0.001 for ActNN and l_c = 0.0001 for CrtNN
Init CrtNN $Q(s,a|\theta^{Q})$ with random weights θ^{Q}
Init ActNN $\mu(s|\theta^{\mu})$ with random weights θ^{μ}
Init observation state s_0, J_0 = 0, J_{pre} = J, epoch=0
Shuffle $\mathbb{S}$
while *epoch* ≤ Z **do**
  epoch = epoch +1
  for *each CTU* **do**
    Initialize CTU partition matrix $I = [-64]_{8\times8}$; load the corresponding target matrix I^*
    Initialize *done*=false and $x_0 = \Phi(s_0)$
    while ***not*** *done* **do**
      −1. Simulate CTU partition with ActNN
      Check the partition matrix I to return *done*
      Select action a_t =ActNN(x_0) , *steps*+ = 1
      Compute action value $u_t = \mu(s_t|\theta^{\mu})$
      a_t = "*split*""*unsplit*" : u_t >= 0
      Execute action a_t at state s_t and update I
      Observe r_t, *done*, s_{t+1}, and $r_t = f(a_t, a_t^*)$
      $x_{t+1} = \Phi(s_{t+1})$
      −2. Update learning rate
      if *steps*%10 == 0 **then**
        l_c+ = 0.01 ; l_a+ = 0.01
      if l_c >0.25 **then**
        l_c = 0.0001
      if l_a >0.25 **then**
        l_a = 0.001
      Buffer observation < s_t, a_t, r_t >
      −3. Update θ^{Q} *for CrtNN*
        min L_c
      Normalize weights of CrtNN for ActNN
      −4. Update θ^{μ} *for ActNN*
        min L_a
      $J'' = J$
return θ^{Q} and θ^{μ};

5.2. The proposed RL based CU early termination algorithm

In the proposed CU early termination algorithm, ActNN is adopted as the CU early termination classifier to predict CU decision for CUs indexed with t, where $t = L_{ID}$ and $t \in [0, 1, \ldots 84]$. The flowchart of applying the CU early termination classifier independent of depths for one CTU partition is shown in Fig. 7.

The procedure of the proposed CU early termination algorithm starts with CU_t, where index t of CU is assigned with L_{ID} and $L_{ID} = 0$. According to the RD cost comparison process for one CTU partition in HEVC, L_{ID} indicates the checking order of CUs. CU_t is located and initialized for compression before checking optimal CTU partition patterns for CU_t with RD cost comparison. ActNN is utilized to predict decisions for CUt independent of depths.

The coding parameter generated from the previous mode checking process are collected as the feature vector of current CU. Features extracted as $\mathbf{x}_t$ from coding information are utilized as the input of ActNN to classify the partition of CU_t, including

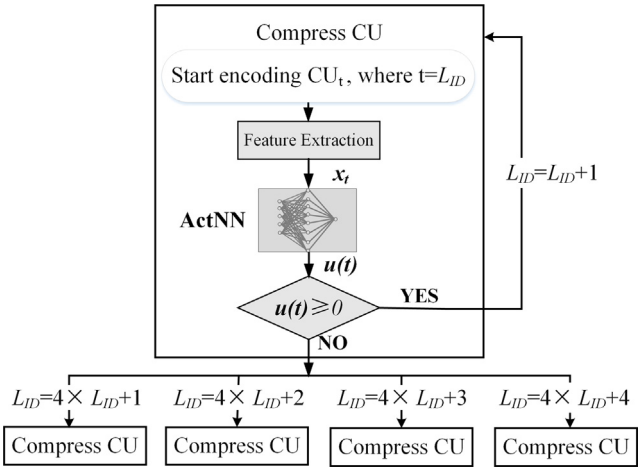

Fig. 7. Flowchart of the proposed CU early termination algorithm.

SKIP/MERGE, INTER and INTRA. According to the feature $\mathbf{x}_t$ extracted from coding information of current CU, ActNN is deployed to determine whether to early terminate the RD cost comparison or split the current CU into sub-CU, according to the probability $u(t)$. $u(t)$ is the output of ActNN. Threshold constrain on $u(t)$ is defined to determine whether split the current CU or not. Classification outputs of AcrtNN are illustrated as two cases in detail.

(i) When $u(t) \geqslant 0$, the prediction of CU early termination is "unsplit". The RD cost comparison process is early terminated for current CU. L_{ID} is updated as $L_{ID}+1$ to indicate the next CU to be checked. $CU_{L_{ID}+1}$ will be prepared for compression if it does not exceed the L_{ID} range of CU in the quadtree. Otherwise, CU at the same depth of the parent node will be checked by updating index t as $L_{ID}+1$.
(ii) When $u(t) < 0$, the prediction of CU early termination is "split". The partition of current CU is determined as "split" and the next CU to be checked with RD cost comparison is CU noted as $4 \times L_{ID}+1, 4 \times L_{ID}+2, 4 \times L_{ID}+3$ and $4 \times L_{ID}+4$, which indicate the sub-CU of current CU.

In the above cases, L_{ID} indicates the CU to be partitioned. CU early termination classifier ActNN is applied to CU indexed by L_{ID} from 0 to 20. To the case when prediction of CU early termination classifier is "unsplit", the CU decision process of current CU is "done". While in the case that the prediction of CU early termination classifier is "split", the Compress CU is called recursively.

6. Experimental results and analyses

We evaluated the proposed RL based CU early termination, which was implemented in HM 16.5 [32] for CU decision prediction. The test sequences were encoded with the low delay B main configuration. The size of CTU and SCU are 64×64 and 4×4, respectively. GOP size is 4. The minimum and maximum Residual Quad Tree transform size is 4 and 32. Motion search range is 64. Other parameters were set as default. The proposed algorithm was evaluated using test sequences recommended by JCT-VC in five resolutions {416×240, 832×480, 1280×720, 1920×1080, 2560×1600}. All experiments were carried out in a PC with 3.4 GHz CPU and 32.0 GB memory, Windows 8 operating system. Experimental comparison between the proposed algorithms and the state-of-the-art algorithms were conducted as the optimal CU decision output from HM16.5. PSNR and bit rate were utilized as the coding efficiency metrics. Bjøntegaard Delta Peak Signal-to-Noise Ratio (BDPSNR), Bjøntegaard Delta Bit Rate (BDBR) [33] were adopted to represent the average PSNR and bit rate differences.

Coding performance of the proposed CU early termination algorithms with different settings of the neural network structure are evaluated in terms of BDBR, BDPSNR and *TS*. Table 3 presents the experimental comparison for the proposed CU early termination classifier ActNN with 6 and 8 neurons, noted as 6NN and 8NN. These two implementation instances of the framework can present different coding performances, which indicate the proposed RL based video encoder can achieve flexible coding performances. The state-of-the-art algorithm, ZupancicTMM [13] and JungTCSVT [4], are utilized to evaluate the proposed implementations of the RL based video encoder as two state-of-the-art works from both the fields of machine learning based method and statistical methods. Average BDBR and BDPSNR between JungTCSVT are 0.53% and −0.019 dB, with *TS* of 32.82%. As to ZupancicTMM, average BDBR and BDPSNR are 3.79% and −0.129 dB, with *TS* of 33.38%. The average BDBR and BDPSNR of the proposed 8NN are 0.85% and −0.033 dB, with *TS* of 34.34%. Whereas, the average BDBR, BDPSNR and *TS* of 6NN are 2.56% and −0.099 dB, with 43.33%. The proposed CU early termination algorithm based on 8NN achieves the similar coding quality with JungTCSVT in terms of BDBR and

Table 3
Evaluation of the proposed fast CU decision prediction algorithm on sequences. [UNIT: %/dB/%].

Sequence	Zupancic et al. [13]			Jung et al. [4]			Proposed (6NN)			Proposed (8NN)		
	BD BR	BD PSNR	*TS*	BD BR	BD PSNR	*TS*	BD BR	BD PSNR	*TS*	BD BR	BD PSNR	*TS*
BQMall 832 × 480	5.14	−0.206	28.02	0.59	−0.025	18.38	5.56	−0.227	37.11	0.41	−0.018	25.31
BQSquare 416 × 240	2.83	−0.118	29.85	0.79	−0.033	26.68	3.38	−0.145	35.72	3.93	−0.167	30.90
BQTerrace 1920 × 1080	3.07	−0.056	40.13	0.54	−0.010	37.11	2.15	−0.038	38.70	0.74	−0.013	45.51
BasketballDrillText 832 × 480	4.93	−0.197	24.04	0.65	−0.027	26.09	4.07	−0.169	35.68	0.19	−0.007	22.23
BasketballDrill 832 × 480	5.12	−0.194	20.65	0.97	−0.038	27.91	3.90	−0.148	32.12	0.25	−0.010	18.32
BasketballDrive 1920 × 1080	5.48	−0.125	39.26	1.60	−0.036	33.88	2.06	−0.046	39.45	0.28	−0.006	26.24
BlowingBubbles 416 × 240	3.38	−0.130	27.06	1.33	−0.051	17.46	3.41	−0.136	24.73	3.05	−0.122	17.44
Cactus 1920 × 1080	4.98	−0.113	23.29	1.36	−0.033	31.04	2.79	−0.065	43.22	1.16	−0.027	32.40
Flowervase 832 × 480	3.39	−0.119	33.10	0.35	−0.015	39.86	1.31	−0.049	60.18	0.35	−0.015	54.92
Flowervase 416 × 240	2.13	−0.111	30.19	−0.51	0.023	40.60	0.53	−0.027	50.06	0.91	−0.045	44.59
FourPeople 1280 × 720	2.59	−0.095	32.53	−0.1	0.004	48.35	1.66	−0.058	65.28	0.40	−0.014	60.33
Johnny 1280 × 720	2.66	−0.058	32.11	−0.62	0.015	47.99	0.90	−0.020	64.05	0.76	−0.010	59.02
Keiba 832 × 480	4.57	−0.171	46.54	0.34	−0.014	20.52	4.56	−0.170	34.60	0.29	−0.010	20.50
Keiba 416 × 240	3.71	−0.190	44.13	0.73	−0.038	14.93	5.57	−0.280	27.20	0.71	−0.090	17.22
Kimono 1920 × 1080	2.62	−0.090	34.71	1.32	−0.045	31.84	0.35	−0.010	34.74	0.23	−0.010	18.72
KristenAndSara 1280 × 720	2.64	−0.080	31.85	−0.54	0.017	46.13	1.58	−0.050	64.67	0.51	−0.020	59.68
Mobisode 832 × 480	4.08	−0.09	29.36	0.00	0.002	36.28	2.30	−0.060	53.62	0.69	−0.020	46.30
Mobisode 416 × 240	5.09	−0.222	33.14	−0.06	0.002	28.29	2.42	−0.107	40.05	0.62	−0.028	34.54
NebutaFestival 2560 × 1600	0.75	−0.029	33.63	1.13	−0.039	34.94	0.05	−0.000	29.67	0.02	−0.000	14.15
ParkScene 1920 × 1080	3.72	−0.116	25.34	0.51	−0.016	30.35	2.84	−0.090	46.42	1.22	−0.040	37.17
PartyScene 832 × 480	2.72	−0.120	36.56	1.72	−0.076	19.26	4.74	−0.210	31.58	2.10	−0.090	21.15
PeopleOnStreet 2560 × 1600	4.81	−0.226	42.09	0.51	−0.023	22.48	5.45	−0.250	29.84	0.37	−0.020	15.06
RaceHorses 832 × 480	4.37	−0.179	37.73	1.37	−0.056	22.42	2.10	−0.180	26.03	0.81	−0.030	12.40
SteamLocomotiveTrain 2560 × 1600	2.05	−0.040	32.20	1.04	−0.022	42.79	0.45	−0.011	48.21	0.20	−0.004	36.64
Tennis 1920 × 1080	7.20	−0.209	42.07	0.62	−0.018	27.66	1.90	−0.057	34.11	0.26	−0.008	19.96
vidyo 720p	3.64	−0.106	32.38	−0.60	0.025	48.96	1.41	−0.043	64.98	0.61	−0.017	60.24
vidyo3 720p	5.48	−0.176	34.57	0.02	−0.01	45.74	2.49	−0.084	61.17	1.27	−0.040	55.12
vidyo4 720p	4.25	−0.114	32.63	0.35	0.012	50.97	1.75	−0.046	60.07	1.45	−0.034	55.59
Average	**3.79**	**−0.129**	**33.38**	**0.53**	**−0.019**	**32.82**	**2.56**	**−0.099**	**43.33**	**0.85**	**−0.033**	**34.34**

BDPSNR, with 10% of *TS* complexity more than JungTCSVT. In ZupancicTMM, CUs in one CTU can be adaptively visited for different optimization steps of low complexity video coding. The proposed CU early termination algorithm with 6NN can achieve more *TS* than ZupancicTMM with much better coding quality. Thus, the effectiveness of the proposed RL based CU early termination is proved. As the CU partition classifier is approximated with one hidden layer neural network over handcraft features, the complexity of the proposed actor-critic RL for CU early termination is lower than the state-of-the-art ML based CU decision classifiers.

Besides the coding performance of the proposed CU early termination algorithm, the classification accuracy of the proposed CU early termination classifier implemented as different encoders is shown in Table 4. The overlap *O* referred to Eq. (10) is adopted to validate the CTU partition accuracy of the proposed CU early termination algorithm. Five sequences with different levels of motion, texture property and resolutions are selected for evaluating the classification performance of the proposed CU early termination algorithm. In Table 4, CU early termination algorithms with two different neural network structures are compared to ZupancicTMM. According to the *TS* in Table 3, ZupancicTMM achieves *TS* of 33.38% which is more than JungTCSVT. Therefore, we compare the CTU partition output of ActNN of different structures with the method developed in [13], as shown in Table 4. Two approximations of ActNN with 6 neurons and 8 neurons are noted as 6NN and 8NN respectively. The classification performance of different algorithms are illustrated with the overlap *O* between the CTU partition matrix output of different algorithms and the optimal CTU partition matrix in Table 4. The classification performance in terms of *O* for 8NN, 6NN and ZupancicTMM are 79.07%, 78.10% and 77.71% on average with variance of 0.840, 0.950 and 0.860 respectively. The proposed 8NN outperforms both 6NN and ZupancicTMM in many sequences coded with different QPs, which is consistent with the coding performance in terms of BDBR, BDPSNR and *TS* in Table 3. Among the proposed 6NN, 8NN and ZupancicTMM, classification performance on values of *O* that are better than the other two in different encoding cases are shown in bold in Table 4.

Table 4
Prediction overlap *O* of the proposed fast CU decision prediction algorithm in comparison with the state-of-the-art methods in [13]. [UNIT: %].

Sequence	Resolution	QP	Prediction Overlap *O*		
			Zupancic et al. [13]	Proposed (6NN)	Proposed (8NN)
Keiba	416 × 240	22	**68.81**	61.66	63.81
		27	**81.00**	67.19	74.97
		32	80.97	81.47	**83.68**
		37	85.27	84.77	**86.38**
BQMall	832 × 480	22	69.34	**72.79**	71.05
		27	75.52	**80.02**	78.70
		32	84.11	83.26	**84.96**
		37	84.66	**87.42**	87.06
FourPeople	1280 × 720	22	78.85	**82.15**	81.71
		27	89.03	89.56	**89.65**
		32	90.87	91.88	**92.58**
		37	94.76	95.24	95.09
Cactus	1920 × 1080	22	60.93	61.99	**62.95**
		27	74.77	77.28	**78.24**
		32	80.08	81.21	**81.82**
		37	83.34	83.94	**84.13**
PeopleOnStreet	2560 × 1600	22	66.51	66.55	**68.20**
		27	67.39	70.04	**71.86**
		32	68.66	71.63	**72.41**
		37	69.33	71.99	**72.16**
Average			77.71	78.10	**79.07**
Variance			0.860	0.950	**0.840**

ActNN with 6 neurons is more variant than ActNN with 8 neurons, which brings more coding efficiency degradation according to the metric value of BDBR and BDPSNR in Table 3 and demonstrates the coding efficiency of the proposed CU early termination algorithm. The CU decision classification overlap indicates that the proposed CU early termination algorithm can save desirable computational complexity with acceptable deteriorating of the encoder performance on BDBR and BDPSNR.

According to the statistical analysis in Section 2, the potential *TS* on applying early termination to the HEVC encoder is 39.5% without losing of coding performance. According to the experimental results, the *TS* of applying CU early termination algorithm with ActNN of 8 neurons can save 34.34% computational complexity which achieves over 86% upper bound of the potential *TS* for early termination, with negligible coding performance degradation. Sequences with different resolutions in {416 × 240, 832 × 480, 1280 × 720, 1920 × 1080, 2560 × 1600} are presented to evaluate the efficiency of the proposed CU early termination algorithm. Although "FourPeople" is encoded in the optimized encoder based on ActNN of 8 neurons with relatively worse encoder efficiency towards *TS* and remarkable degradation in BDBR and BDPSNR as in the optimized encoder of 6 neurons, the CU decision classification accuracy of the proposed optimized encoder for "FourPeople" is over 89.76% on average in terms of the overlap *O*. Therefore, the proposed CU early termination algorithm is proved to achieve desirable coding efficiency, so as to reduce the CU decision complexity.

The computational overhead of ActNN for sequences "BasketballDrill" and "ParkScene" coded with four QPs in {22, 27, 32, 37} are presented in Figs. 8 and 9. The time consumption ratio of applying CU early termination algorithm to CU with size in {64 × 64, 32 × 32, 16 × 16} are evaluated over 50 frames of each sequence, which are no more than 0.016%. The percent of time computation on early termination prediction increases monotonously with the CU early termination ratio for CU at different depths in Fig. 8. The complexity of the proposed CU early termination algorithm is negligible, as each CU decision consumes only 40 floating-point multiplication.

The proposed CU early termination algorithm independent of depths implemented independent of depths is proposed to focus on optimizing the overall CU decision for partitioning one CTU. Computational overhead of applying the CU early termination algorithm is negligible. If we do not perform RD cost comparison

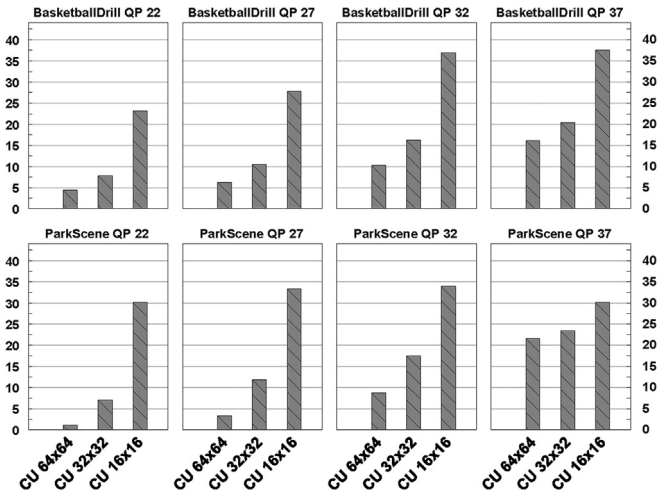

Fig. 8. Percent of CU predicted as early termination at Depth D0, D1 and D2. The top row corresponds to sequence "BasketballDrill" compressed by HM 16.5 with QPs in {22, 27, 32, 37}. The bottom row corresponds to sequence "ParkScene" compressed by HM 16.5 with QPs in {22, 27, 32, 37}. [UNIT: %].

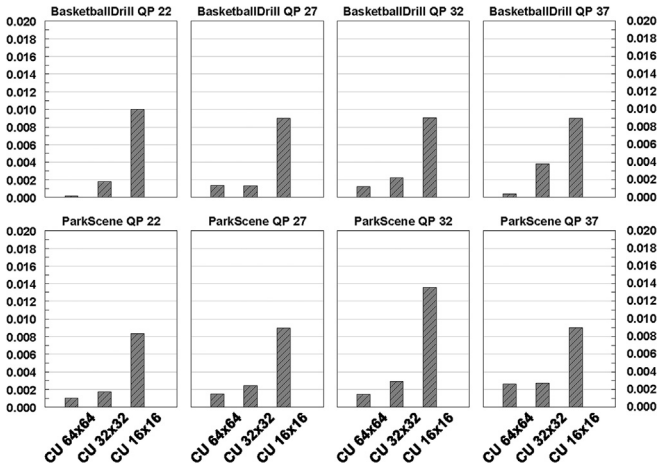

Fig. 9. Percent of time consumption of CU early termination prediction at Depth D0, D1 and D2. The top row corresponds to sequence "BasketballDrill" compressed by HM 16.5 with QPs in {22, 27, 32, 37}. The bottom row corresponds to sequence "ParkScene" compressed by HM 16.5 with QPs in {22, 27, 32, 37}. [UNIT: %].

on the following depths when reaching the optimal depth according to brute-force RD cost comparison, we can assume from Table 1 that the coding efficiency is very close to the optimal one. The experimental results of the proposed CU early termination algorithm with ActNN of 8 neurons verify the assumption. On the other hand, the elimination of RD cost comparison on depths lower than the optimal depth observed by exhaustive RD cost comparison will cause coding efficiency degradation, as illustrated with the coding efficiency results of the proposed CU early termination algorithm with ActNN of 6 neurons.

The coding efficiency is related to the prediction accuracy of the proposed CU early termination algorithm, which is independent of depths. Deliberate feature selection, network structure enhancement and RL algorithm for learning the CU early termination classifier ActNN will improve the prediction accuracy, so as to improve the coding efficiency. Efficiency of the proposed framework can be further improved through deliberate choices on feature selection, increasing the number of neurons in the hidden layer of neural network, and adjusting the hyper-parameter for the proposed actor-critic RL algorithm. The prediction accuracy of the proposed CU early termination algorithm can be further improved through not only selecting features deliberately, but also increasing the number of layers for NN. Besides, the structure of the CU early termination algorithm could be further extended to PU decision for further complexity reduction. As to the development of feature selection algorithms and RL algorithms, it is possible to investigate prosperous combinations of feature selection strategies, the classification approximation function and policy gradient RL algorithm for the proposed framework.

7. Conclusion

In this paper, a framework of RL based video encoder optimized with the CU early termination classifier independent of depths is proposed, where the Rate Distortion (RD) cost comparison process is modeled as the Markov Decision Process (MDP). The CU early termination classifier is learned as Actor Neural Network (ActNN) approximated with one hidden layer neural network by an end-to-end actor-critic RL algorithm. Then, a CU early termination algorithm is developed independent of depths with the CU early termination classifier. The flexible of the proposed CU early termination algorithm is proved with different neural network approximations of ActNN. The video coding performance of the proposed RL based CU early termination is evaluated with the experimental comparison with the state-of-the-art methods. The proposed overall framework of RL based video encoder for CU decision early termination can be extended to other coding parameter decision problems in video coding.

Conflict of interest

The authors declared that there is no conflict of interest.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 61471348, 61871372, 61672443, in part by Guangdong Natural Science Foundation for Distinguished Young Scholar under Grant 2016A030306022, in part by Shenzhen Science and Technology Development Project under Grant JCYJ20170811160212033 and Shenzhen International Collaborative Research Project under Grant GJHZ20170314155404913, in part by the Key Project for Guangdong Provincial Science and Technology Development under Grant 2017B010110014, in part by Free Application Fund of Natural Science Foundation of Guangdong Province under Grant 2018A0303130126, in part by RGC General Research Fund (GRF) 9042322, 9042489 (CityU 11200116, 11206317), in part by Guangdong International Science and Technology Cooperative Research Project under Grant 2018A050506063, in part by Membership of Youth Innovation Promotion Association, Chinese Academy of Sciences under Grant 2018392.

References

[1] G.J. Sullivan, J. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, IEEE Trans. Circ. Syst. Video Technol. 22 (12) (2012) 1649–1668.
[2] Y. Zhang, S. Kwong, G. Zhang, Z. Pan, H. Yuan, G. Jiang, Low complexity HEVC INTRA coding for high quality mobile video communication, IEEE Trans. Industr. Inf. 11 (6) (2015) 1492–1504.
[3] Y. Zhang, S. Kwong, G. Jiang, X. Wang, M. Yu, Statistical early termination model for fast mode decision and reference frame selection in multiview video coding, IEEE Trans. Broadcast. 58 (1) (2012) 10–23.
[4] S. Jung, H.W. Park, A fast mode decision method in HEVC using adaptive ordering of modes, IEEE Trans. Circ. Syst. Video Technol. 26 (10) (2016) 1846–1858.
[5] J. Kim, S. Jeong, S. Cho, J.S. Choi, Adaptive coding unit early termination algorithm for HEVC, in: IEEE International Conference on Consumer Electronics (ICCE), 2012, pp. 261–262.
[6] Y.-J. Ahn, D. Sim, Fast mode decision and early termination based on perceptual visual quality for HEVC encoders, J. Real-Time Image Proc. (2017) 1–16.
[7] L. Shen, Z. Zhang, Z. Liu, Effective CU size decision for HEVC intracoding, IEEE Trans. Image Process. 23 (10) (2014) 4232–4241.
[8] Y. Zhang, S. Kwong, X. Wang, H. Yuan, Z. Pan, L. Xu, Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding, IEEE Trans. Image Process. 24 (7) (2015) 2225–2238.
[9] Y. Zhang, Z. Pan, N. Li, X. Wang, G. Jiang, S. Kwong, Effective data driven coding unit size decision approaches for HEVC INTRA coding, IEEE Trans. Circ. Syst. Video Technol. 28 (11) (2018) 3208–3222.
[10] L. Zhu, Y. Zhang, Z. Pan, R. Wang, S. Kwong, Z. Peng, Binary and multi-class learning based low complexity optimization for HEVC encoding, IEEE Trans. Broadcast. 25 (4) (2017) 547–561.
[11] Y. Zhang, Z. Pan, N. Li, X. Wang, G. Jiang, S. Kwong, Effective data driven CU size decision approaches for HEVC intra coding, IEEE Trans. Circ. Syst. Video Technol. 28 (11) (2017) 3208–3222.
[12] L. Zhu, Y. Zhang, S. Kwong, X. Wang, T. Zhao, Fuzzy SVM-based coding unit decision in HEVC, IEEE Trans. Broadcast. 64 (3) (2018) 681–694.
[13] I. Zupancic, S.G. Blasi, E. Peixoto, E. Izquierdo, Inter-prediction optimizations for video coding using adaptive coding unit visiting order, IEEE Trans. Multimedia 18 (9) (2016) 1677–1690.
[14] X. Shen, L. Yu, J. Chen, Fast coding unit size selection for HEVC based on bayesian decision rule, Picture Coding Symp. 8355 (3) (2012) 453–456.
[15] H. Kim, R. Park, Fast CU partitioning algorithm for HEVC using an online-learning-based bayesian decision rule, IEEE Trans. Circ. Syst. Video Technol. 26 (1) (2016) 130–138.

[16] J. Xiong, H. Li, F. Meng, S. Zhu, Q. Wu, B. Zeng, MRF-Based fast HEVC inter CU decision with the variance of absolute differences, IEEE Trans. Multimedia 16 (8) (2014) 2141–2153.
[17] H. Chen, T. Zhang, M. Sun, A. Saxena, M. Budagavi, Improving intra prediction in high-efficiency video coding, IEEE Trans. Image Process. 25 (8) (2016) 3671–3682.
[18] G. Correa, P.A. Assuncao, L.V. Agostini, L.A. da Silva Cruz, Fast HEVC encoding decisions using data mining, IEEE Trans. Circ. Syst. Video Technol. 25 (4) (2015) 660–673.
[19] F. Duanmu, Z. Ma, Y. Wang, Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension, IEEE J. Emerg. Sel. Top. Circ. Syst. 6 (4) (2016) 517–531.
[20] F. Duanmu, Z. Ma, Y. Wang, Fast CU partition decision using machine learning for screen content compression, in: IEEE International Conference on Image Processing, 2015, pp. 4972–4976.
[21] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, D. Wang, CU partition mode decision for HEVC hardwired intra encoder using convolution neural network, IEEE Trans. Image Process. 25 (11) (2016) 5088–5103.
[22] M. Xu, T. Li, Z. Wang, X. Deng, Z. Guan, Reducing complexity of HEVC: a deep learning approach, IEEE Trans. Image Process. 27 (10) (2018) 5044–5059.
[23] P. Helle, H. Schwarz, T. Wiegand, K.R. Müller, Reinforcement learning for video encoder control in HEVC, in: IEEE International Conference on Systems, Signals and Image Processing, 2017, pp. 1–5.
[24] C. Chung, W. Peng, J. Hu, HEVC/H.265 coding unit split decision using deep reinforcement learning, in: International Symposium on Intelligent Signal Processing and Communication Systems, 2017, pp. 570–575.
[25] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, 2016.
[26] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G. Sullivan, Rate-constrained coder control and comparison of video coding standards, IEEE Trans. Circ. Syst. Video Technol. 13 (7) (2003) 688–703.
[27] H. Everett, Generalized Lagrange multiplier method for solving problems of optimum allocation of resources, Oper. Res. 11 (3) (1963) 399–417.
[28] A. Ortega, K. Ramchandran, M. Vetterli, Optimal trellis-based buffered compression and fast approximations, IEEE Trans. Image Process. 3 (1) (1994) 26–40.
[29] D. Bertsekas, Dynamic Programming and Optimal Control, fourth ed., Athena Scientific, 2012.
[30] J. Si, Y. Wang, Online learning control by association and reinforcement, IEEE Trans. Neural Networks 12 (2) (2001) 264–276.
[31] L. Zhu, Y. Zhang, N. Li, G. Jiang, S. Kwong, Machine learning based fast H.264/AVC to HEVC transcoding exploiting block partition similarity, J. Vis. Commun. Image Represent. 38 (C) (2016) 824–837.
[32] JCT-VC, HM software, 2014 (Online). Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.5/ (accessed 23 Dec. 2017).
[33] G. Bjøntegaard, Calculation of average PSNR differences between RD-curves, in: ITU-T Q. 6/SG16 VCEG, 15th Meeting, Austin, Texas, USA, April, 2001.