Fast Selection of INTRA CTU Partitioning in HEVC Encoders using Artificial Neural Networks

Mateusz Lorkiewicz
Poznań University of Technology
Institute of Multimedia Telecommunications
Poznań, Poland
ORCID: 0000-0002-9020-1927

Olgierd Stankiewicz
Poznań University of Technology
Institute of Multimedia Telecommunications
Poznań, Poland
ORCID: 0000-0001-9691-9094

Marek Domański
Poznań University of Technology
Institute of Multimedia Telecommunications
Poznań, Poland
ORCID: 0000-0002-9381-0293

Hsueh-Ming Hang
National Chiao Tung University,
Hsin Chu, Taiwan
ORCID: 0000-0001-8965-2619

Wen-Hsiao Peng
National Chiao Tung University,
Hsin Chu, Taiwan
ORCID: 0000-0002-4421-8031

Abstract— In the intra-frame video coding, an image is divided into small blocks, and the actual coding is performed individually in these blocks. In this paper, the process is considered in the context of the widely used HEVC compression, where the optimum choice of the division is crucial for the rate-distortion performance. Unfortunately, the search for such optimum division needs very many operations, and is done on the basis of "try and check" approach in the classic implementations. The idea of the paper is to replace this complex part of the encoder by a neural network, and some variants of the potential neural networks are studied and compared in the paper. For the chosen network, the complexity of the encoder is vastly reduced at the cost of negligible loss in the rate-distortion performance. These features are demonstrated using an extensive set of frames from many test video sequences.

Keywords— Video coding, compression, encoder control, HEVC, fast mode selection, CTU partitioning, neural network

I. INTRODUCTION

The High Efficiency Video Coding (HEVC) [1,2] technology is currently widely applicable in many new systems, especially in ultra-high definition television (UHDTV) systems [3,4], where HEVC serves most of 4K and 8K television channels. Many television operators, like in Poland, Taiwan and many other areas, have decided to employ version 2 of the DVB digital television system for terrestrial services (DVB-T2) [3] that provides higher spectral efficiency than the currently employed DVB-T1 system. This change is often also related to the a shift from the AVC (Advanced Video Coding) [5] to the HEVC technology. Such proliferation of HEVC inevitably increases the needs for efficient and cheap HEVC encoders as their applications in television will induce more applications in all internet-based multimedia services like video over-the-top (OTT).

Well-designed HEVC encoders provide bitrates that are halved as compared to the bitrates produced by AVC encoders that ensure the same quality of the decoded video. The difficulty with HEVC encoders is related to their complexity that is much higher than that of the AVC encoder currently mostly employed for TV and OTT. High complexity of HEVC encoders is challenging also because of the requirement of provision of limited latency when even for demanding content in 4K or 8K formats.

High efficiency of HEVC encoders is provided thanks to efficient choices among various coding modes that are available in huge numbers. The decisions on those choices are made in the process of rate-distortion optimization that yields high complexity of the HEVC encoder implementations.

In order to facilitate development of HEVC codecs, the MPEG (of International Organization for Standardization ISO) and VCEG (of International Telecommunication Union ITU) expert groups have provided freely available reference software [6] and its description [7]. The reference software implements the HEVC encoders including rate-optimization procedures. Therefore, the reference software provides the near-optimum efficiency of compression thanks to near-optimum decisions made during encoding of video.

Here, in the paper, we focus on an important part of the decision process, i.e. on the partitioning of the coding tree units (CTUs) in the intra-frame mode. The goal is to reduce the computational effort of those decisions by compromising the rate-distortion performance, by increasing the bitrate below 5% by multi-fold reduction of the processing time needed for the decisions.

Our idea is to use a pre-trained artificial neural network (ANN) that mimics the decisions of the classic encoder control algorithms developed in the HEVC reference software. Therefore, the neural network is trained by the decisions taken from the HEVC reference software, using a huge data set of possible CTUs. In that way, the processing time for the decisions on CTU partitioning is multi-fold reduced as the effort on multiple CTU encoding cycles is spared.

The goal of this process is to find the division of a given CTU that is a block of luma samples of the size mostly 32×32 or 64×64 luma samples. The CTU has to be divided into coding units (CUs) that can be as small as a block of 8×8 luma samples. In the CUs, different coding modes can be applied in order to improve the rate-distortion performance as the mode can be adapted to the local properties of the prediction error.

The rate-distortion algorithm, as used in the HEVC reference software, can be described as a greedy approach. For CTU division, it checks possible prediction modes and transform block divisions and estimates the number of bits for current size of the coding unit (CU) using a simplified model of the CABAC encoder [6,7]. Next, a division into four blocks can be performed and calculations are run for smaller blocks. The sum of the bits required for divided block is calculated and compared. If a division yields better encoding efficiency,

2021 Signal Processing Symposium (SPSympo) | 978-1-6654-1274-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/SPSYMPO51155.2020.9593483

the computations for the next level of division are conducted. In this process, with the increase of the CTU size, the number of sub-variants increases exponentially. This is very important for the intra-frame mode of encoder, where the full-scan approach is often used and the output bitstream is the largest.

As the technology of CTU partition used for the next generation of video encoders (VVC –Versatile Video Coding) [8,9] is in many aspects quite similar to that from HEVC, the authors demonstrate that the approach from this paper is also adaptable for this newest generation of video encoders where the complexity issues are even more critical than for HEVC.

II. RELATED WORKS

In intra-frame coding, complexity reduction solutions can are mainly in two categories. The first group are heuristic approaches based on estimation of specific features that are used feature extraction, which are then used to make the decision on early termination of the partitioning process in HEVC [10-12] or VVC [13].

The second category of solutions is learning-based CU partitioning, mostly using artificial neural networks (ANNs). In many papers, an ANN is used for early termination of divion process. Feng et al. [15] proposed ANN-based algorithm, which estimates thee depth ranges of the currently processed CTU. Other researchers, e.g.: Xu [14], Li [16], described methods that use ANNs for split decisions on each division level. In this approach, one can train a separate ANN for each division level (e.g. Chen [23]) or one ANN with multiple outputs (e.g. Li [18]). Additionally, for VP9, Paul [21] applied a network with multiple outputs, and early termination for level divisions outputs in order to achieve better performance. The time saving for presented methods vary from 20 to 70 percent with ΔBD_{RATE} from 1.5 to 3 percent. Liu [17] presented application of such approach in hardware encoder.

A yet another ANN approach is to estimate whole partitioning pattern. Katayama [20] created a network with multiple inputs to estimate partitioning pattern for currently processed block. Other approach was presented by Ren [24], which used IPB-CNN network which uses CTU samples.

As an input for the ANN, most of the methods use luma part of currently processed CTU, as in default Brute Force approach in HEVC. Katayama [20] used neighboring preprocessed samples with good results (ΔBD_{RATE} 1.8 %). Other approach was introduced by Amer [22], which used features from Laplacian Transparent Composite Model. As training data authors used images from two sources: few first frames from JCT-VC test set [26] (which was then used for network evaluation) or separate dataset (e.g. RAISE [27]). Moreover the ANN used in most approaches are relatively big (~ 1M of weights [14,16]), but some authors were able to achieve good result with multiple models of size ~40k weight [20].

In this paper, we use ANN with novel architecture for whole division matrix estimation, which mimics quaternary tree architecture. Additionally we have performed a series of experiments to check impact of context of CABAC encoder on CTU division decisions. Then we present architecture which utilize additional context data in order to improve ANN accuracy. We compare our results with Xu [14] and Ren [24]. Approach [14] applies hierarchical model with relatively big networks (~1M weights, learned with RAISE dataset [25]) for each division level. Ren [24] uses same whole division matrix estimation approach with convolutional network, but with shallower and wider ANN (which is bigger than proposed in this paper), has worse learning performance and was trained using JCT-VC dataset [29]

III. NEURAL NETWORK TRAINING FRAMEWORK

The neural network learning process is conducted using the supervised approach. The first step is the choice of the training and verification datasets. As input to a neural network, we choose the luma component from the currently processed CTU of size 64 by 64. For this purpose, we use DIV2K dataset [28], which consists of 800 images in the training subset and 100 images in the validation subset. The images have at least 2000 samples in at least one dimension, and are available in a non-compressed format (.png). Therefore, 522,939 and 66,650 CTUs are available for the training and the validation subsets, respectively. For training and verification, the luma sample values are modified by subtraction of a constant of 128 (8-bit samples) in order to reduce the average values in the individual CTUs.

The reference decisions for training and verification are obtained using the HEVC reference software version 16.23 [6,7].

A partitioning pattern is described by a division matrix [7] that consists of integers from the range from 0 to 3. Each number corresponds to an 8×8 block of samples (smallest possible CU size) and describes the division depth of a block. The size of the division matrix is 8×8, but we can reduce its size to 4×4 without any information loss as shown in Fig. 1. This smaller representation yields a smaller number of outputs from the neural network, which is beneficial for the training process.

The dataset is then used for training of the neural network. Keras software from Tensorflow API is used.

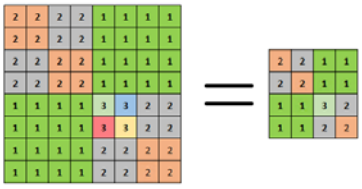

Fig. 1. Reduction of redundancy in the division matrix.

IV. STRAIGHTFORWARD APPROACH

In the straightforward approach, the CTUs are classified according to their content into classes corresponding to the partitions of CTUs. There are 83,522 possible division patterns in HEVC. The pattern is represented as a matrix 4×4×4 – division matrix in format 4×4 which every digit is represented in hot-one format. In this form, every division matrix value (division level) can be represented as single 1 digit in corresponding position.

The architecture of the artificial neural network is depicted in Fig. 2.

The neural network consists of two sections – the convolutional Subnetwork A and the quaternary tree imitating convolutions Subnetwork B. Subnetwork A consists of 4 convolutional layers. Each layer is composed of 2D convolution (kernel: (3,3) , stride: (1,1), padding: “same”), batch normalization, PReLu activation function (separate parameter for each channel) and max-pooling (pool size: (2,2), stride: (2,2) and padding: ‘valid’). The inputs of the consecutive layers are reduced in size 2 times for two dimensions, but the number of convolution channels

increases. Luma samples are normalized to range <0;1> before ANN processing.

Subnetwork B consists of 3 layers and it follows the top-to-bottom approach of the division process. Each layer is composed of convolution, batch normalization and PReLu activation.The first layer applies (3,3) convolution and corresponds to the first level of division. The product of this layer is then split into four 2×2×64 tensors. This corresponds to block splitting in HM. Then, all smaller tensors are processed in separate convolutional layers with (3,3) kernel and 16 channels. Those convolution layers do not share weights. To mimic the last level we should perform one more split. Instead, we concatenate output tensors from second layer and use convolution with (1,1) kernel. The stride is (1,1) in all layers. The output is processed by the softmax layer.

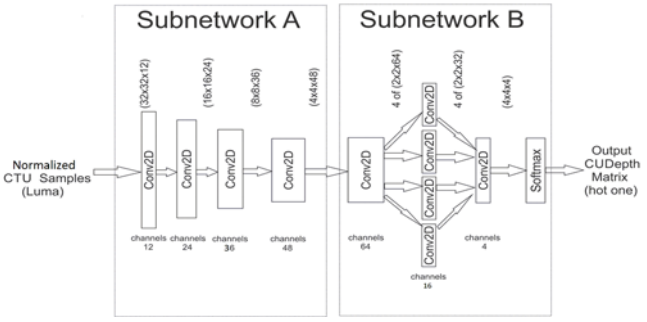

Fig. 2. Simple ANN architecture for CTU partitioning.

The above described neural network is relatively small and has 207,256 parameters. The training process was conducted using the categorical cross entropy as the loss function and ADAM optimizer with learning rate 0.001 was used. Learning of the neural network was performed with minibatch (64 training examples) with data shuffle approach for 150 learning epochs using data encoded with same *QP* = 27. Typical learning curves are presented in Fig. 3.

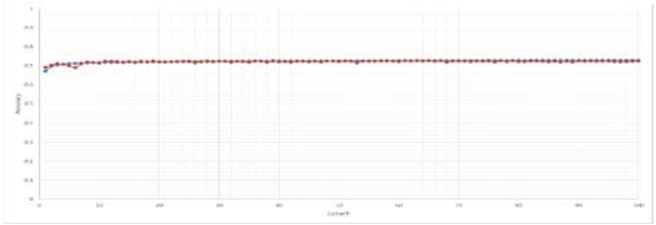

Fig. 3. Straightforward model learning curves for training (blue) and validation (red) data. No overfitting visible.

The learning process ends mostly with accuracy of about 73% on both the training and the validation dataset. We achieved similar learning curves for models learned with QP 22, 32 and 37 with accuracy of 74.5%, 72.5% and 70%. In case of *QP*=37 slight overfitting can be seen starting from epoch 70. Additionally, the output from the neural network must be checked and corrected due to possibility of generation of an invalid partitioning pattern, i.e. one that corresponds to an invalid description of the CTU partitioning. In such cases, CTU partitioning pattern has to be corrected (see: section VII).

From the neural network training perspective, results presented in Fig. 3 are not satisfactory. The results are also affected by the "long tail" distribution of the dataset, i.e. there exist few division patterns that appear much more frequently than others. Therefore, in order to make learning examples more unique we decided to add additional data to the network input to increase its performance.

V. Analysis of the Contextual Effects

Basing on the shape of attained learning curve, we came to a hypothesis that the aforementioned straightforward approach suffers from the insufficient information delivered to the neural network that should mimic the operation of the reference encoder. In fact, the decisions taken by the reference encoder for a given CTU heavily depend on the decisions taken for the previous CTUs.

It may be demonstrated by the example where small noise was added to the input images (cf. Table 0). The results are obtained for the whole DIV2K dataset [28].

TABLE I. Effect of Introduction of Small Input Image Changes on the Decisions in the Reference Encoder.

Changed samples		**Percentage of CUs with changed encoding-tree depths**
Random. *RMS* of the noise:	0.01	63,43%
	0.05	71,73%
	0.10	71,11%
	0.50	72,53%
	1.00	73,23%
Single pixel ±1 change	Top	10,80%
	Middle	6,70%
	Bottom	0,01%

As it can be seen, even for very low *RMS* of the noise, like (0.01), the amount of changed CUs is significant and reaches 63% for training subset and 65% for validation subset. Noise with *RMS* amplitude of 0.01 means that about very few image samples are modified by +/-1, while the others remain unchanged.

The example demonstrates that the decisions made by the reference encoder are strongly affected by the decisions made for the previous CTUs.

The experimental study demonstrates also significant modifications of the partitioning decisions for many CTUs. These experiments show that in the HEVC reference encoder, the selection of the partitioning modes is highly contextual. Moreover it has been shown, that even small changes in the input image can lead to vast changes of partitioning mode selected.

Therefore, the attained results support hypothesis that these effects have to be considered when training neural networks for fast selection of encoding-tree depths. This has inspired our further research presented below.

VI. Context-aware Neural Network Architecture

Based on the hypothesis presented above, the model created by training artificial neural network should use wider context of data. The context–aware neural network is presented in Fig. 4. Simple holistic approach is extended with three additional convolutional subnetworks, whose inputs are neighboring samples from the reconstructed image, the division matrices from the adjacent CTUs and the division matrixes from *N* previously encoded CTUs. Additional context data is collected as shown in Fig. 5.

During the encoding of an image in HEVC, the CTUs are processed in raster-scan order, i.e. line by line (from the top to the bottom) and from the left to the right. Therefore, for the decisions in a currently processed CTU, the decisions made for the blocks located above the current one and to the left of the current CTU should be taken into account.

The subnetwork for neighboring data consist is similar to subnetwork for CTU samples. It consists of 3 convolutional

layers defined as for samples subnetwork, but skips the max pooling in the last layer. The outputs of consecutive convolutional layers have 2, 3 and 4 channels respectively. The neighboring samples are prepared same as CTU samples and written to input 2D array. We consider the neighboring samples as continuous vector which starts from bottom sample of left neighboring CTU. The first 64 samples are placed in bottom left part of array, then, the values from 32th to 95th are written to top left section, the values from 65th to 129th are inserted to top right section and the last 64 samples goes to bottom right section. This approach allows for usage of 2D convolutions. Samples are also placed in the areas, where they may have most impact. If the neighboring samples from certain direction are not available for the currently processed CTU, the missing values are set to zero.

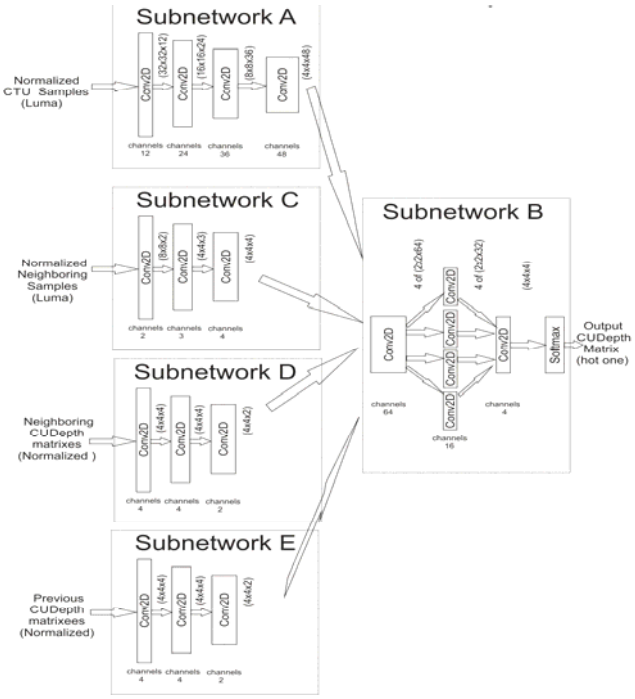

Fig. 4. Proposed context-aware neural netfork for CTU partitioning. CNN – convolutional neural network.

Subnetworks for the neighboring division matrices and the previous CTUs division matrices share the same architecture consisting of three convolution layers. One layer is composed of 2D convolution (stride: (1,1), padding: "same"), batch normalization and PRelu (different parameter for each channel). The first and the last convolutions apply (1,1) kernels, and have 4 and 2 output channels, respectively. The second convolution uses (3,3) kernels with 4 output channels. The input tensor for those subnetworks has size 4×4×M, where M is the number of input division matrixes (4 for neighboring and 8 for previous). If some data is unavailable, the missing division matrices are filled with the value -1. The input tensors (v) are preprocessed according to formula: $v \rightarrow (v+1)/4$. As the elements of the division matrix should be integers in range <0;3> after the preprocessing, we need to keep all the values in range <0;1>.

After adding additional subnetworks, the whole model is still relatively small and has 97,410 parameters. The training process was performed in the same conditions as for the straightforward approach. Typical learning curves are presented in Fig. 6. We trained the models with data encoded with constant quantization parameters: 22, 27, 32, 37, and achieved 75%, 74%, 73% and 71.5% of accuracy respectively. Again, we observed slight overfitting for QP=37. In comparison with the straightforward approach, a small accuracy gain is observed, so better performance of this model is expected.

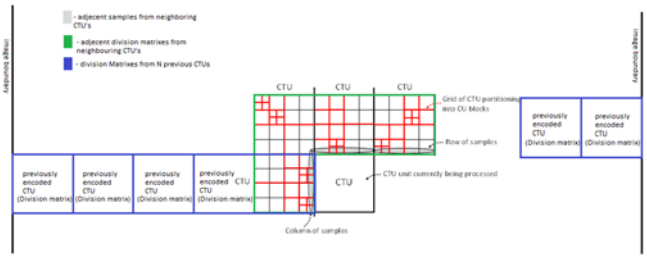

Fig. 5. Description of the context information used for the division matrix estimation by the artificial neural network.

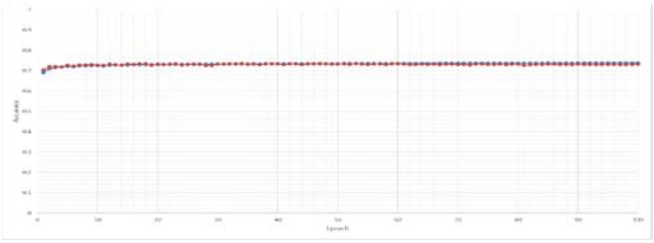

Fig. 6. Contextual model learning curves for training (blue) and validation (red) data. No overfitting visible.

VII. Application in HEVC encoder

Learning performance of the proposed neural network architectures inspired us to implement them inside HEVC encoder in order to assess them in real-world conditions. For this sake we have selected MPEG HEVC Reference Software HM version 16.23. It has been used both as a reference and as a basis for modifications for the application of the proposals.

In the original HM software, CTU partitioning is selected through means of rate-distortion optimization. Possible partitioning patterns are considered and compared, then the best candidate is selected. In our proposals this mechanism is bypassed and overridden with a neural network, which instantly selects a single CTU partitioning pattern. Therefore, much computational power is saved.

Apart from bypassing the CTU partitioning selection, the remaining rate-distortion optimization steps remain unchanged. Therefore, the steps, like selection of prediction method, transform size and small QP variations, are still optimized by the encoder, as in the HM model.

As mentioned earlier, the output from the neural network must be checked and enforced to be conformant with the HEVC syntax. For this, we have introduced a correction algorithm. For CTU partitioning pattern generated by the neural network, it takes into account the fact, that the sum of all elements from output is always equal 16 (due to softmax layer). Firstly, the sum of all outputs for division level 0 are calculated. If result is bigger than 8, then all the elements of the division matrix are set to 0. Otherwise the algorithm looks through values, which correspond to the divided areas. The algorithm sums the outputs of the network in smaller areas. It corresponds to division level 1 in smaller areas. Then, it checks, if the result exceeds 2. If so, the values in this are set to 1. Otherwise the outputs are set to 2 or 3 depending on the values corresponding to the division level outputs.

Moreover it is necessary to specially treat partitioning of CTU blocks which are incompletely laying on boundaries of

images which are not divisible by CTU size (64×64), e.g. last row of CTUs in Full-HD images (with 1080 lines). Original HM software works in so called conformance window mode, where splitting of such CTUs is enforced. In order to allow fair performance evaluation of our proposed neural networks, we have also enforced such splitting for the CTU partitioning patterns generated by the proposed neural networks.

VIII. Encoding results

For testing dataset we have used mentioned earlier DIV2K dataset [28] (both training and validation subset) and JCT-VC sequence dataset [29]. These sequences are grouped in five classes (A, B, C, D, E) with varying resolution and frame-rate (Table 0).

TABLE II. Summary of JCT-VC test sequences [29].

Class	Number of sequences	Resolution	Frame rates
A	4	2560×1600	30; 60
B	5	1920×1080	24; 50
C	4	832×480	30; 50; 60
D	4	416×240	30; 50; 60
E	3	1280×720	60

The encoding has been as performed in accordance with the "Common test conditions" (CTC) [29] developed by MPEG/JCT-VC group during their works on HEVC. In particular, the "All Intra" configuration has been used with the following quantization parameters: 22, 27, 32 and 37. The tested cases: (a)-(e) have been listed in Table 0: Results for (d) and (e) are cited for JCT-VC dataset Unfortunately, the authors have not provided data for "Nebula Festival", "SteamLocomotion" sequences nor for DIV2K dataset.

TABLE III. Two approaches compared to other encoders.

Tested case	Description
(a)	The reference (original HM version 16.23)
(b)	Proposal with non-contextual NN (Section IV)
(c)	Proposal with contextual NN (Section VI)
(d)	Technique described in [14]
(e)	Technique described in [24]

Here, the widely used Bjøntegaard metric [31] is used for comparison. We use $\Delta BD_{RATE,}$ that expresses the average increase of the bitrate for the considered encoder with respect to the bitrate for the reference encoder, measured for the constant value of the luma *PSNR*. A negative value of $\Delta BD_{RATE,}$ indicates a reduction in bitrate with respect to the reference. Similarly, positive ΔBD_{PSNR}

The averaged results for DIV2K dataset [28] are presented in Table IV. For JCT-VC dataset [29], the detailed ΔBD_{RATE} results are presented in Table V and averaged results are shown in Table VI, both for ΔBD_{RATE} and ΔBD_{PSNR}. We have also measured the reduction of the encoder complexity attained with usage of the proposed neural networks. This experiment has been performed on Intel Core i7 computer, with the use of a single-thread process. The results for the reduction of the processing time [26] in the intra-frame encoder are presented in Table VII.

TABLE IV. Averaged HEVC encoding results for DIV2K dataset.

DIV2K subset	Proposed Non-contextual Network (b) vs reference (a)		Proposed Contextual Network (c) vs reference (a)	
	ΔBD_{RATE} [%]	ΔBD_{PSNR} [dB]	ΔBD_{RATE} [%]	ΔBD_{PSNR} [dB]
Training	1.51	-0.069	5.97	-0.279
Validation	1.77	-0.075	6.49	-0.268

TABLE V. Detailed HEVC encoding results (see Table 0). for JCT-VC dataset [29].

JCT-VC	Sequence	ΔBD_{RATE} [%]			
		Proposed (b) vs ref. (a)	Proposed (c) vs ref. (a)	[14] (d)vs ref. (a)	[24] (e) vs ref.(a)
A	NebulaFestival	1.31	9.79	-	-
	PeopleOnStreet	2.16	7.70	2.37	2.91
	SteamLocomot.	2.05	14.47	-	-
	Traffic	2.23	6.81	2.55	1.90
B	BQTerrace	1.36	5.58	1.84	1.83
	BasketballDrive	2.97	11.00	4.27	0.60
	Cactus	2.21	8.23	2.27	-0.01
	Kimono1	1.85	12.42	2.59	1.64
	ParkScene	1.69	4.93	1.96	-1.55
C	BasketballDrill	2.56	10.24	2.86	3.26
	BQMall	1.60	8.87	2.09	2.32
	PartyScene	0.49	6.63	0.66	0.94
	RaceHorses	1.56	7.09	1.97	1.63
D	BasketballPass	1.43	9.45	1.84	1.55
	BlowingBubbles	0.44	6.40	0.62	1.05
	BQSquare	0.66	7.54	0.91	0.79
	RaceHorsesLow	1.21	7.82	1.32	1.37
E	FourPeople	2.63	10.55	3.11	1.29
	Johnny	3.10	17.61	3.82	3.48
	KristenAndSara	2.62	14.87	3.46	2.83

TABLE VI. Averaged HEVC encoding results (see Table 0) for JCT-VC dataset [29].

JCT-VC class	Proposed (b) vs ref. (a)		Proposed (c) vs ref. (a)		[14] (d) vs ref. (a)		[24] (e) vs ref. (a)	
	ΔBD_{RATE} [%]	ΔBD_{PSNR} [dB]	ΔBD_{RATE} [%]	ΔBD_{PSNR} [dB]	ΔBD_{RATE} [%]	ΔBD_{PSNR} [dB]	ΔBD_{RATE} [%]	ΔBD_{PSNR} [dB]
A	1.94	-0.099	9.69	-0.462	2.46	-0.126	2.41	-0.210
B	2.02	-0.077	8.43	-0.307	2.58	-0.090	0.50	-0.176
C	1.55	-0.085	8.21	-0.470	1.90	-0.099	2.04	-0.128
D	0.93	-0.061	7.80	-0.526	1.17	-0.072	1.19	-0.070
E	2.78	-0.138	14.35	-0.683	3.46	-0.050	2.29	-0.153
All	1.81	-0.089	9.40	-0.471	2.25	-0.085	1.55	-0.142

TABLE VII. Encoding time reduction (see Table 0) estimated for JCT-VC dataset [29].

JCT-VC class	$\Delta T = 100\% \cdot \left({}^{T_{tested}}/_{T_{reference}} - 1 \right)$			
	Proposed (b) vs ref. (a)	Proposed (c) vs ref. (a)	[14] (d) vs ref. (a)	[24] (e) vs ref. (a)
A	-59.69	-60.37	-65.90	-59.95
B	-61.58	-60.66	-70.61	-68.92
C	-60.04	-61.30	-53.26	-55.07
D	-61.97	-60.67	-49.64	-43.829
E	-59.88	-59.23	-72.28	-65.56
All	-60.63	-60.45	-61.08	-59.07

As it can be seen, encoding with the proposed non-contextual network (b) performs quite well, especially considering its poorer learning performance. Whereas the usage of this network leads to slight increase in bitrate for about 1.75% (for maintaining the same quality - ΔBD_{RATE}) it also significantly reduces computational complexity and thus encoding time, for about 60%. This result is comparable to the one presented in paper [14].

Interestingly, the contextual-network (c), which during learning outperformed the non-contextual-network (b), in this experiment attained less satisfactory results. The average bitrate loss is higher – about 6% for DIV2K dataset [28] and about 9% for JCT-VC dataset [29]. The encoding time reduction is similar to this attained with the non-contextual network – about 60%.

IX. Conclussions

In the paper, it is proposed to replace the classic "try and check" strategy for the choices of the CTU partitioning in the HEVC encoders. We propose to replace this step by an appropriately trained ANN that produces the decisions upon the CTU divisions. The obvious goal is to reduce the huge complexity of the HEVC encoders.

In the paper, the approach to ANN learning is based on mimicking the decisions made by the HM model. We proposed the quaternary-tree-inspired ANN architecture, which have relatively small size compared to similar approaches [14] and performs similarly efficient to more complicated architectures that employ the hierarchical approach [24].

Two proposed approaches are considered in this paper:

- The straightforward approach (Section IV), where the ANN is trained using the CTU luma samples;
- The contextual approach (section VI), where the ANN is trained using both CTU luma samples and the CTU context formed by the luma samples from the neighboring CTUs and the decisions already made for the neighboring CTUs.

The result of the extensive experiments is that the straightforward approach mimics the decisions of HM less accurately than the ANN trained according to the contextual approach. Astonishingly, the performance of the whole intra-frame video encoder is worse for the contextual approach than for the straightforward one.

Generally, the attained results are comparable to those obtainable using the state-of-the-art solutions known from the literature (e.g. [14, 24]). In an HEVC encoder in the intra-frame mode, the proposed network allows for substantial reduction of the computational effort (about 60%) at the cost of small (about 1.7%) average bitrate increase.

It is worth to mention that the recently standardized VVC coding technology employs similar CTU partitioning. Therefore, it is likely that the approach would be applicable to VVC as well.

Acknowledgment

The work was supported by Ministry of Science and Technology (MOST) of Taiwan and National Centre for Research and Development (NCBiR) of Poland under a joint research project.

References

[1] "Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 2: High Efficiency Video Coding," ISO/IEC IS 23008-2, also ITU-T Rec. H.265, 2013.

[2] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Trans. Circuits Systems for Video Techn., Dec. 2012.

[3] K. Hayashi, K. Kumamaru and S. Yokozawa, "Development of new UHD-1 (4K)/UHD-2 (8K) UHDTV satellite broadcasting system in Japan," SMPTE Motion Imaging Jrnl, vol. 129, no. 6, pp. 15-24, 2020.

[4] L. Vangelista et al., "Key technologies for next-generation terrestrial digital television standard DVB-T2," in IEEE Communications Magazine, vol. 47, no. 10, pp. 146-153, October 2009.

[5] "Coding of audio-visual objects — Part 10: Advanced Video Coding," ISO/IEC IS 14496-10, also ITU-T Rec. H.264, 2014.

[6] 2D HEVC reference codec available online https:// hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.18.

[7] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, G. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Encoder Description, "Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document: JCTVC-R1002, 18th Meeting: Sapporo, JP, 2014.

[8] "Information technology — Coded representation of immersive media — Part 3: Versatile video coding", ISO/IEC IS 23090-3, also ITU-T Rec. H.266, 2021.

[9] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, .-K. Wang, "Developments in international video coding standardization after AVC, with an overview of Versatile Video Coding (VVC)" Proceedings of the IEEE (Early Access), 2021, doi: 10.1109/JPROC.2020.3043399.

[10] N. Kim, S. Jeon, et al, , "Adaptive keypoint-based CU depth decision for HEVC intra coding," Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB), pp. 1–3, Jun. 2016.

[11] M. Khan, M. Shafique, J. Henkel, "An adaptive complexity reduction scheme with fast prediction unit decision for HEVC intra encoding," Proc. IEEE Int. Conf. Image Process., Sep. 2013,pp. 1578–1582.

[12] X. Shen, L. Yu, J. Chen, "Fast coding unit size selection for HEVC based on Bayesian decision rule," Proc. Picture Coding Symp., May 2012, pp. 453–456.

[13] Q. Zhang, Y. Wang, L. Huang, B. Jiang, "Fast CU partition and intra mode decision method for H.266/VVC," IEEE Access, vol. 8, pp. 117539-117550, 2020.

[14] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, Z. Guan, "Reducing complexity of HEVC: A deep learning approach", IEEE Trans. Image Proc., vol. 27, pp. 5044 - 5059, Oct. 2018.

[15] Z. Feng, P. Liu, K. Jia1, K. Duan, "HEVC fast intra coding based CTU depth range prediction", 3rd IEEE Int. Conf. on Image, Vision and Computing, Chongqing, China, 2018.

[16] Y. Li, Z. Liu, X. Ji, D. Wang, "CNN based CU partition mode decision algorithm for HEVC inter coding", IEEE Int. Conf. Image Proc. (ICIP), Athens, pp. 993-997, 2018.

[17] Z. Liu, X. Yu, S. Chen, D. Wang, "CNN Oriented Fast HEVC Intra CU Mode Decision", IEEE International Symposium on Circuits and Systems (ISCAS), Montreal 2016, pp. 2270-2273

[18] T. Li, M. Xu, X. Deng, "A deep convolutional neural network approach for complexityreduction on intra-mode HEVC", IEEE Int. Conf. Multimedia & Expo (ICME), pp. 1255-1260, Hong Kong, 2017.

[19] K. Kim, W. W. Ro, "Fast CU Depth Decision for HEVC Using Neural Networks",IEEE Trans. On Circuits And Systems For Video Technology, Vol. 29, pp. 1462 – 147, May 2019.

[20] T. Katayama, K. Kuroda, W. Shi, T. Song, T. Shimamoto , "Low-Complexity Intra Coding Algorithm Based on Convolutional Neural Network for HEVC", International Conference on Information and Computer Technologies (ICICT), pp. 115-118, DeKalb, May 2018.

[21] S. Paul, A. Norkin, A. C. Bovi, "Speeding up VP9 intra encoder with hierarchical deep learning based partition prediction", IEEE Trans. Image Proc.,vol 29, pp. 8134 – 8148, July 2020.

[22] H. Amer, A. Rashwan, E. Yang, "Fully connected network for HEVC CU split decision equipped with Laplacian transparent composite model", Picture Coding Symp. (PCS), San Francisco, June 2018.

[23] Z. Chen, J. Shi, W. Li, "Learned fast HEVC intra coding", IEEE Trans. Image Proc., vol. 29, 2020, pp. 5431-5446.

[24] W. Ren, J. Su, Ch. Sun, Z. Shi, "An IBP-CNN based fast block partition for intra prediction", Picture Coding Symp., Nov.2019.

[25] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, G. Boato, "RAISE:A raw images dataset for digital image forensics," Proc. 6th ACMMultimedia Syst. Conf., pp. 219–224, 2015.

[26] G. Correa, P. Assuncao, L. Agostini, and L. da Silva Cruz, "Performance and computational complexity assessment of high-efficiencyvideo encoders, "IEEE Trans. Circuits Syst. Video Technol., vol. 22,no. 12, pp. 1899–1909, Dec. 2012.

[27] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, G. Boato, "RAISE:A raw images dataset for digital image forensics," Proc. 6th ACMMultimedia Syst. Conf., pp. 219–224, 2015.

[28] https://data.vision.ee.ethz.ch/cvl/DIV2K/.

[29] "Common test conditions and software reference configurations", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 12th Meeting, Document: JCTVC-L1100, WG11 m28412, Geneva, Switzerland, 2013.

[30] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, T. Wiegand, "Comparison of the coding efficiency of video coding standards—Including high efficiency video coding (HEVC) ,"IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1669–1684, 2012.

[31] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves", ITU-T SG16 / Q6, Doc. VCEG-M33, 2001.