Complexity reduction of ANN model for CU size selection in HEVC

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

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

Marek Domański
Institute of Multimedia Telecommunications
Poznan University of Technology
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 HEVC compression is performed in Coding Units (CUs) being pixel blocks of a size adaptively chosen according to the local content within a video frame. Near-optimum selection of the frame partition into CUs is crucial for the coding efficiency. A huge number of partitioning schemes is available and the optimum partitioning scheme is obtained in an iterative computation-heavy procedure in a classic HEVC encoder. In order to reduce the encoding time and the encoding energy, a few approaches have been proposed with the use of neural networks (NNs). These approaches demonstrate a significant reduction of the encoding time and a negligible increase of the bitrate as compared to the traditional iterative approach. Nevertheless, they use very large neural networks whereas it is demonstrated in this paper that much smaller neural networks provide similar results encoding tome reduction with the similar bitrate reduction.**

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

I. Introduction

Applications of Artificial Intelligence (AI) constitute a rapidly growing area of both research and technology. Among various applications, video coding seems to be a promising area for future AI-based technology. The research on AI-based video coding has two major directions: end-to-end artificial networks for compression, and classic structures where some functional blocks are replaced by neural networks both in encoders and decoders. Here, in this paper, we focus on the latter approach.

In the paper, we focus on High Efficiency Video Coding (HEVC) [1,2] which is the premium video coding technology for ultra-high definition (UHD) video for both television and the internet. In many practical applications, HEVC replaces less efficient and less complex AVC (Advanced Video Coding) [3]. The adoption of HEVC is additionally accelerated by the proliferation of version 2 of the DVB digital television system for terrestrial services (DVB T2) [4,5]. Also, HEVC is a widely used video coding technology in consumer devices like smartphones or cameras, where often the encoded video is shared among a limited number of recipients, which breaks the classical one-encoder-many-decoders balance. Therefore, the encoder complexity reduction is of paramount importance for mobile and consumer battery-powered devices where energy consumption is crucial.

The aforementioned factors vastly increase the demand for efficient and low-cost HEVC encoders that feature low consumption of energy fed from the battery of portable devices.

In modern encoders, high performance is ensured with an efficient choice among different coding modes, which, in the case of HEVC, are available in large number. These choices are decided within the rate-distortion optimization (RDO) process which involves iterative encoding and comparison scheme in order to find the best performing partitioning scheme (from a rate-distortion perspective). Such an approach results in a high complexity of the HEVC encoders. To facilitate the development and proliferation of HEVC, the HEVC reference software HM [6] was issued, which is freely available with its description [7]. HM software implements the entire HEVC decoder and encoder, including RDO routines for rate-distortion optimization (RDO). In HM, RDO operates on the level of coding blocks, in HEVC called coding tree units (CTUs). Due to this, HM ensures suboptimal compression efficiency by making near-optimal decisions during video encoding.

In HM software the RDO algorithm follows a greedy approach. For CTU partitioning, it checks all possible block splits and estimates the number of bits for the current coding unit (CU) size using the simplified CABAC binary encoder model [6,7]. Then, further partitioning into four sub-blocks can be performed and calculations can be repeated for smaller blocks. If the deeper partitioning results in higher coding efficiency, calculations for the next level of partitioning are executed. In this process, the total number of sub-options increases exponentially as the base size of the CTU increases. This is very important for an intra-frame encoder, where a full-scan approach is often used and the output bitrate is the highest. During the RDO process, the encoder successively compares the rate-distortion performance of all of the analyzed modes and finally selects the best one. Although RDO allows for the generation of semi-optimal output bitstreams, computationally it is very demanding.

Our research is related to an important part of the decision process which entails most of the complexity of the encoding: the partitioning of CTUs in the intra-frame mode. The purpose of this process is to find the partitioning of a given CTU, which is a block of image samples with a size of usually 32×32

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.

or 64×64. The CTU must be partitioned into CUs, which in HEVC can be as small as 8×8 luma samples. Different coding modes can be applied to the CUs to improve the rate-distortion performance, as the mode can be adapted to local prediction error properties.

In this paper we propose a new version of ANN from our previous work [8], which reduces the computational cost with maintained rate-distortion performance. Our new approach uses the architecture that emphasizes the gradient flow through the model during training, which yields better results of such a model. The novelty of the current approach is related to the significantly reduced size of the neural network. In this paper, we propose a convolutional network that is much smaller than the networks described in [8,9,10]. Despite the significantly reduced number of weights of the NN, the efficiency of the CU partitioning process is even slightly improved as compared to the much larger NNs described in the abovementioned references.

II. General idea

The main idea is to use an artificial neural network (ANN) that is trained to mimic the decisions of the encoding control algorithms developed in HEVC reference software that controls CTU partitioning. Thus, the training is performed on the basis of the decisions made by the HM, using a huge dataset of CTUs. In this way, the processing time for CTU splitting decisions is many times shorter, as the effort of multiple CTU encoding cycles is saved.

In our approach, we reduce the hierarchical partitioning decision problem to a well-known classification problem. We assume the maximum size of the CTU, which is 64×64 samples. For each CTU, we classify components for the lowest level CU blocks (LLCUs) into particular four partitioning depth levels: from 0 (no partitioning, whole CTU intact) to 3 (the deepest partitioning). This is illustrated in Fig. 1. Therefore, the ANN outputs 4 probabilities of 4 depth partitioning levels for each LLCU. Since there are 4×4 LLCUs (each representing 8×8 samples) for the whole CTU there are 64 outputs from our ANN.

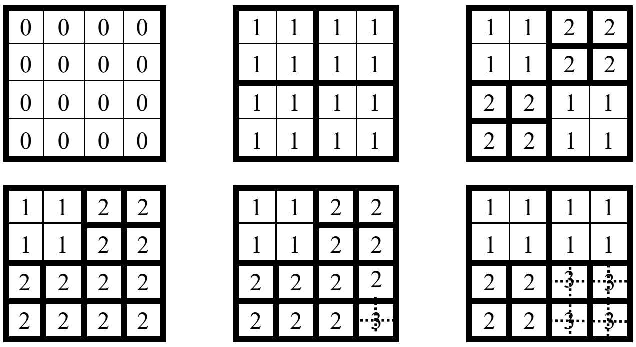

Fig 1. Examples of CTU partitioning schemes based on partitioning depth levels (0..3) in LLCU blocks. Please note that LLCU blocks with level 3 are partitioned even furtherly (deeper – dotted line), than the grid of the LLCUs (solid bold line).

As compared to our previous work [8], in this paper we present the following novelties:

- The improved version of the ANN architecture which is significantly reduced in size (from ~93k weights to ~43k weights) and offers better learning capabilities.
- A more insightful comparison of complexity and number of parameters between proposed and other models known from the literature.

The approach presented in this paper is demonstrated on Intra frames only, but it can also be adapted Inter frames and to the next generation of video coding technologies, where CTU partitioning is very similar (e.g. VVC - Versatile Video Coding) [11,12]. Complexity issues in such encoders are even more critical than in the case of HEVC.

III. Related works

The most modern encoder complexity reduction methods [13] aim at finding the most probable encoding modes without performing full RD optimization. This is typically attained by two means. The first one is heuristic methods. Specific features are identified (typically based on expert knowledge) which are used to decide on an earlier decision-making process in HEVC [14,15,16] or VVC [17]. Thanks to this encoding time is shortened and complexity is reduced.

The second category of solutions is learning-based methods. Here, the features are learned from the training dataset. Nowadays, most often this is done with the use of ANNs. Methods in this category are the most similar to the approach presented in this work, especially in the context of Intra-frame encoding considered.

In many works, artificial neural networks are used for early termination of the partitioning process and selection amongst only the options indicated by ANN. Feng et al. [18] proposed an algorithm that estimates the depth ranges of currently processed CTUs. In other works, including: Xu [9], Li [19], ANNs are used to make splitting decisions at each partitioning level. In such an approach, one can train a separate ANN for each partitioning level (e.g. Chen [20]) or a single ANN with multiple outputs (e.g. Li [21]). Paul [22] focused on VP9 and used a network with multiple outputs and early termination for the outputs of the partitioning levels to achieve better performance. Time savings for the presented methods range from 20 to 70 percent with Bjøntegaard [23] ΔBDRATE of about 1.5 to 3 percent (the bigger the time savings the bigger the bitstream size increase). Liu [24] presented an application of the mentioned method in a hardware encoder.

Yet another approach is to estimate the entire partitioning pattern at once. Katayama [25] created ANN with multiple inputs to estimate the partitioning pattern for the currently processed CTU block. Another approach was presented by Ren [10], who applied an IPB-CNN network using CTU samples. A similar approach is used in this work.

As input for ANNs, most methods employ luma samples from the CTU currently being processed. This is similar to the default brute-force RDO approach in the HEVC reference encoder. Katayama [25] used adjacent preprocessed samples, obtaining good results but trained and evaluated on the same set of data (part of JCT-VC) [26]. Amera [27], on the other hand, used features from the Laplacian Transparent Composite Model. As training data, images from two sources are used: the first few frames from the JCT-VC test set [26] (which was then used for network evaluation) or a separate dataset (e.g. RAISE [28]).

In general, it can be noted, that the ANNs used in most approaches are relatively large (~1M weights [9,19]), but some authors have been able to achieve good results using the hierarchical approach with multiple models of ~40k

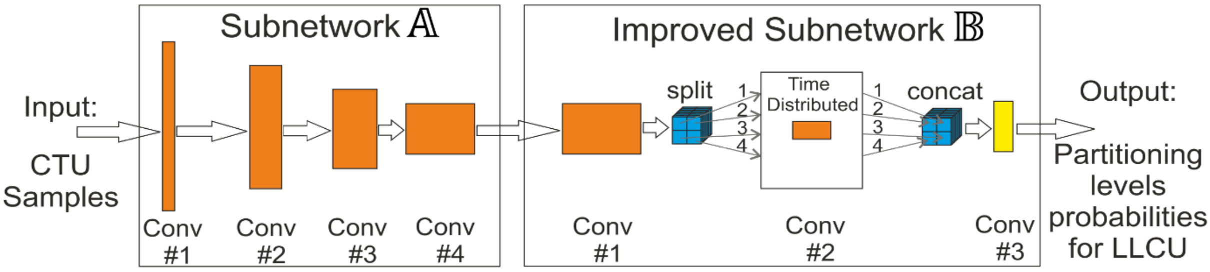

Fig. 2. The proposed architecture of artificial neural network in a straightforward approach, with improved subnetwork 𝔹, as compared to [1].

weights [20]. Ren [10] uses a similar approach to estimate the all-division matrix with a convolutional network, but with a shallower and wider ANN (which is significantly larger than the one proposed in this paper), has poorer learning performance, and was trained using the JCT VC dataset [26].

IV. Modified Straightforward approach

The straightforward approach aims to find a matrix of LLCU that will give the closest to HM encoding performance. The neural network model process CTU samples and return a 4×4×4 matrix of probabilities. The first dimension corresponds to a division layer, remaining two describe the position of LLCU in the image. For each LLCU returned are probabilities of division levels, which describe how likely represented area belongs to a certain division level. As a target (supervised training example) a division pattern is featured as a division matrix, whose every element is represented in one-hot format.

The proposed improved architecture is depicted in Fig 2. It consists of two separate subnetworks: 𝔸 and 𝔹.

The goal of subnetwork 𝔸 is to create a deep latent representation of CTU samples and reduce input data spatial dimensions to 4×4 (division matrix size). This part of the model consists of 4 layers. Each of them performs 2D convolution, batch normalization, processing through PReLU activation, ending with a max-pooling operation. Convolutions, with kernel size 3 by 3, in consecutive layers output 12, 24, 36 and 48 feature maps. The input Luma samples are converted to (0;1) range values format.

Subnetwork 𝔹 is designed to emulate the quaternary tree partitioning algorithm used in HEVC. Each convolution layer corresponds to consecutive divisions. Every layer consists of 2D convolution, batch normalization and activation (PReLU). Most parameters of layers remain unchanged compared to our previous work [8], but a modification was made to 2nd layer. Instead of 4 separate, parallel layers a single one is used, which processes data serially. Output from the first layer is split into four 64×2×2 sections, each of which contains latent representation corresponding to CU blocks of sizes 32×32. Next, data is serialized in order: top-left, top-right, bottom-left and bottom-right and then processed. Next, the outputs are concatenated into single features map in order that puts consecutive outputs into the previous spatial position. Finally, the data is fed to the last layer, which, after convolution, produces output probabilities using the softmax operation.

The aforementioned serialization modification yields with direct reduction of the number of parameters used by the whole network to only 63 760, as compared to 92 600 in our previous work [8]. Also, it allows to use the whole gradient information to update parameters of the modified subnetwork 𝔹 during the training phase [29]. Additionally, the filter weights are shared among split data, so during learning we get more generalizing features extracted from convolution. As the modified layer is one of the last in the model, this improvement will influence the rest of the network in the backward direction. The experiments demonstrated that this modification allows for a decrease of the number of convolution filters in the first and the second layer of Subnetwork 𝔹 to 32 and 8 respectively, which yields even further reduction to only 42 832 model parameters. The more comprehensive performance analysis is presented in Section VI

Of course, serialization of ANN may result in increased computational time, especially on platforms, which can easily parallelize computations. Despite that, multiple devices, which operate with limited hardware resources will benefit from reduced parameters number, e.g. be more energy efficient. More analyses on time complexity and performance are presented in Section VI D.

V. Experimental results

A. Learning dataset

For the purpose of learning the DIV2k [30] test image dataset was used. First, the images were encoded with HM software [6], whose decisions we aim to mimic. The results, e.g. CTU partitioning schemes selected by the HM encoder, were used to generate reference CTUs data. The final training set consisted of 589 589 CTUs, which were divided into two subsets: 522 939 and 66 650 CTUs for training and validation, respectively.

B. ANN models training

In our approach a separate ANN model is used for different quantization parameter (QP) setting of the encoder. As described later, the experiments have been performed with accordance to the "All Intra" scenario in Common Test Conditions (CTC) for HEVC [26] which assumes the evaluation of the encoder with four constant QP parameters (22, 27, 32 and 37). This set of QP parameters corresponds to the use of the encoder in practical applications. Therefore, we learned separate ANN for each of them, as only one QP value can be used in the CTU encoding process in the given scenario. These four networks share the same architecture but were learned with DIV2k-based datasets encoded using different QP.

For the learning of ANN models we have used TensorFlow [31] library. The loss function was categorical cross-entropy [32] with ADAM [33] as an optimization algorithm. The learning process was performed in batches of 64 samples. The training data were shuffled every learning epoch. The optimizer was restarted after every 10th epoch. This setup leads us to achieve learning convergence after 100 learning loop iterations. The achieved accuracy varies around

73, 71, 70 and 69 percent for QP parameter values 22, 27, 32 and 37, respectively. Measured performance was similar for training and validation datasets.

C. Evaluation methodology

For the model performance evaluation, we have used two datasets. The first one is based on the DIV2k image dataset, as described in previous subsections. The second one is a subset of the JCT-VC video dataset. In the latter case the evaluation was conducted in accordance with the "All Intra" scenario from CTC [26] for HEVC. This dataset consists of sequences collected in classes characterized by the same resolution and similar frame rate (Table I).

TABLE I. SEQUENCE CLASSES IN JCT-VC DATASET.

JCT-VC class	A	B	C	D	E
Resolution	2560×1600	1920×1080	832×480	416×240	1280×720
Frame rate	60 or 30	60,50 or 24	60,50 or 30	60,50 or 30	60

The proposed ANN model was embedded into HM version 16.23 software that is used for experimental validation of the technique proposed. Reductions in encoding time were evaluated based on results given by HM software build-in total encoding time mechanism. Implementation of the ANN utilizes PyTorch as a backend [34]. The ANN was limited to use only one thread for the CTU partitioning estimation process as the HM is a single-threaded application.

All experiments were performed on the AMD Ryzen 9 5900X platform with 32 GB of RAM and Windows 11 (build 22000.493) as the operating system. All time-based analysis experiments were executed on NVME solid-state drives.

D. Modified straightforward approach

The first proposed model was evaluated in two ways: encoding efficiency in comparison to other methods and processing time of ANN. In Table II, we present Bjøntegaard metric [23] results for output bitrate for particular sequences and classes.

TABLE II. BITRATE INCREASE (ΔBD RATE [%]) VERSUS HM.

JCT-VC class	Sequence	ΔBD_{RATE} [%]			
		Proposed model	**Previous work [8]**	**[9]**	**[10]**
A	NebulaFestival	1.24	1.31	-	-
	PeopleOnStreet	2.20	2.16	2.37	2.91
	SteamLocomot.	2.10	2.05	-	-
	Traffic	2.26	2.23	2.55	1.90
B	BQTerrace	1.36	1.36	1.84	1.83
	BasketballDrive	2.98	2.97	4.27	0.60
	Cactus	2.11	2.21	2.27	-0.01
	Kimono1	1.82	1.85	2.59	1.64
	ParkScene	1.70	1.69	1.96	-1.55
C	BasketballDrill	2.61	2.56	2.86	3.26
	BQMall	1.63	1.60	2.09	2.32
	PartyScene	0.48	0.49	0.66	0.94
	RaceHorses	1.63	1.56	1.97	1.63
D	BasketballPass	1.51	1.43	1.84	1.55
	BlowingBubbles	0.46	0.44	0.62	1.05
	BQSquare	0.63	0.66	0.91	0.79
	RaceHorsesLow	1.32	1.21	1.32	1.37
E	FourPeople	2.49	2.63	3.11	1.29
	Johnny	3.11	3.10	3.82	3.48
	KristenAndSara	2.52	2.62	3.46	2.83

We set these results in comparison to the previous straightforward model [8] and results from other successful approaches known from the literature [9,10]. As we can observe modified architecture performed very similar to the original architecture, and sometimes mildly surpass the previous version. In Table III, the average results for bitrate [23] are shown. Analogously, Table IV shows results for visual quality. Please note that the average result for class A was calculated only for two sequences (*PeopleOnStreet* and *Traffic*) like in [9,10]. It is clear that the proposed solution gives slightly higher quality as compared to its predecessor. Here we can see that the bitrate increase is smaller for higher resolution classes. Additionally, the overall bitrate increase is smaller by 0.04. Similar observations may be done in regards to PSNR reduction – the proposed solution gives a little better encoded sequence quality.

TABLE III. BITRATE INCREASE (ΔBD RATE [%]) VERSUS HM AVERAGED OVER VIDEO CLASSES.

CT-VC class	Proposed model	Previous work [8]	[9]	[10]
A	2.18	2.20	2.46	2.41
B	2.01	2.02	2.58	0.50
C	1.56	1.55	1.90	2.04
D	0.92	0.93	1.17	1.19
E	2.84	2.78	3.46	2.29
All	**1.90**	1.94	2.25	1.55

TABLE IV. LUMA PSNR REDUCTION (ΔBD PSNR [DB]) VERSUS HM AVERAGED OVER VIDEO CLASSES.

CT-VC class	Proposed model	Previous work [8]	[9]	[10]
A	-0.124	-0.125	-0.126	-0.210
B	-0.076	-0.077	-0.090	-0.176
C	-0.085	-0.085	-0.099	-0.128
D	-0.060	-0.061	-0.072	-0.070
E	-0.140	-0.138	-0.050	-0.153
All	**-0.097**	-0.097	-0.085	-0.142

In order to visualize the results of the proposed method more clearly we also present the rate-distortion (RD) curves for averaged results for JCT-VC sequences. In Fig 3 a) and b) we shod RD curves for the proposed model and HM. One can observe that the curves are almost identical. The difference in compression efficiency is very small.

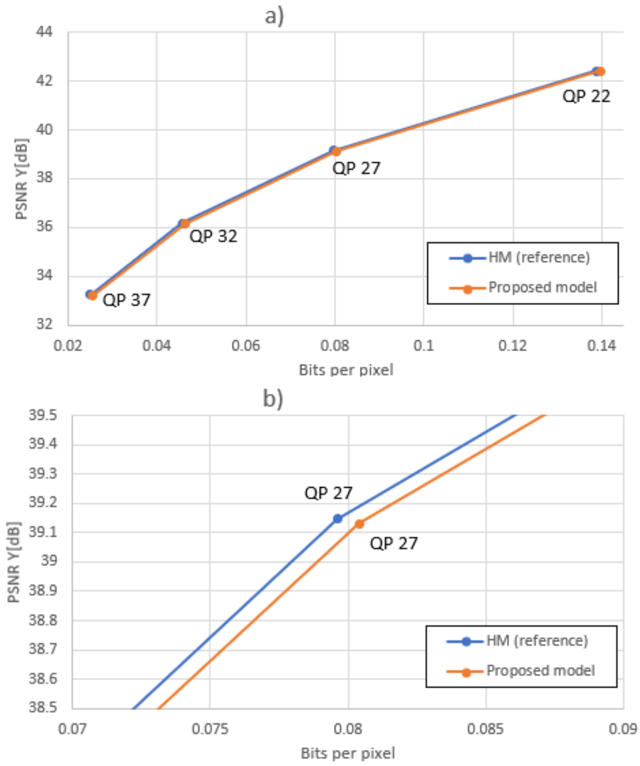

Fig 3. RD-curves for HM (reference) and proposed model. a) RD-curves are almost identical overlaps b) Zoom in on the RD-curves for QP=27 point. A small difference between curves is visible

An analysis of complexity reduction of the encoder was done by means of encoding time, presented in Table V, alongside results from our previous work [8] (on two platforms: new and previous one) and other works [9,10]. The new solution slightly outperforms the previous one for high resolution sequences. For classes A, B and E, the proposed model performs better as compared to our previous work, but for classes C and D time reduction is slightly worse. Note, that the presented model was evaluated on an AMD R9 processor, but solutions in [8] and [9] were used on Intel-i7-based machines. Unfortunately, work [10] does not provide information about the hardware used in the experiments. The operating system – Windows 11 – is quite new so thread management problems may have caused the observed performance decrease. Despite that, time reduction for the presented and the previous model [8] shows similar problems on the new platform.

TABLE V. REDUCTIONS OF THE TOTAL ENCODING TIME VERSUS HM 16.23.

JCT-VC class	$\Delta T = 100\% \cdot (T_{tested}/T_{reference} - 1)$				
	Proposed model	**Previous work [8] – new test platform**	**Previous work [8]**	**[9]**	**[10]**
A	-62.40	-62.11	-59.69	-65.9	-59.95
B	-58.19	-58.65	-61.58	-70.61	-68.92
C	-44.08	-43.60	-60.04	-53.26	-55.07
D	-38.46	-38.14	-61.97	-49.64	-43.83
E	-59.06	-58.74	-59.88	-72.28	-65.56
All	**-52.39**	-52.24	-60.63	-61.08	-59.07

For further comparison, in Table VI we show the numbers of trainable parameters used in networks. Unfortunately, the authors of [10] did not present this number, so we estimated it by creating a network, following the available architecture description, and summarizing the model, using TensorFlow utilities. As shown in Table V, the presented approach uses significantly fewer weights than other methods.

TABLE VI. NUMBER OF WEIGHTS EMPLOYED IN ANN MODELS.

	Proposed Model	**Previous Work [8]**	**[9]**	**[10]**
Number of parameters	**42 832**	91 600	1 287 189	3 852 928

VI. CONCLUSIONS

The presented method can be used to efficiently select the partitioning scheme of CTUs in the encoder and replace the classic RDO algorithm in HEVC, which operates on a computationally expensive “try and check” strategy. This is done by employing ANN that mimics the decisions made by the RDO algorithm like in the HEVC reference software.

The proposed new version of ANN architecture allows for the reduction of the number of parameters used by the whole network to only ~43k weights as compared to ~93k in our previous work [8] and ~1.3M in [9] and ~3,9M in [10]. Since the size of the network is considerably reduced, the complexity of the encoder is also decreased. This is particularly important for applications related to mobile devices, where energy efficiency is significant. As shown in the results, the proposed modification can even provide a slight drop in encoding time as compared to our previous work [8]. Moreover, the results opts, that this way of composing a model allow to train it better and may increase the efficiency of the neural networks in other applications.

This complexity improvement of the proposed ANN is attained while sustaining encoding performance comparable to solutions known from the literature, e.g. as compared to the reference (HM encoder), the proposed approach causes only negligible loss in the rate-distortion performance.

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, Geneva, Switzerland, 2013.

[2] G. J. Sullivan, J. -R. Ohm, W. -J. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012, doi: 10.1109/TCSVT.2012.2221191.

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

[4] K. Hayashi, K. Kumamaru and S. Yokozawa, "Development of New UHD-1 (4K)/UHD-2 (8K) UHDTV Satellite Broadcasting System in Japan," in *SMPTE Motion Imaging Journal*, vol. 129, no. 6, pp. 15-24, July 2020, doi: 10.5594/JMI.2020.2993188.

[5] 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, doi: 10.1109/MCOM.2009.5273822.

[6] 2D HEVC reference codec available online https:// hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.23.

[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] M. Lorkiewicz, O. Stankiewicz, M. Domanski, H. -M. Hang and W. -H. Peng, "Fast Selection of INTRA CTU Partitioning in HEVC Encoders using Artificial Neural Networks," *2021 Signal Processing Symposium (SPSympo)*, LODZ, Poland, 2021, pp. 177-182, doi: 10.1109/SPSympo51155.2020.9593483..

[9] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang and Z. Guan, "Reducing Complexity of HEVC: A Deep Learning Approach," in *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5044-5059, Oct. 2018, doi: 10.1109/TIP.2018.2847035.

[10] W. Ren, J. Su, C. Sun and Z. Shi, "An IBP-CNN Based Fast Block Partition For Intra Prediction," *2019 Picture Coding Symposium (PCS)*, Ningbo, China, 2019, pp. 1-5, doi: 10.1109/PCS48520.2019.8954522.

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

[12]]B. Bross, J. Chen, J. -R. Ohm, G. J. Sullivan and Y. -K. Wang, "Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC)," in *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1463-1493, Sept. 2021, doi: 10.1109/JPROC.2020.3043399.

[13] C. E. Rhee, K. Lee, T. S. Kim and H. -J. Lee, "A survey of fast mode decision algorithms for inter-prediction and their applications to high efficiency video coding," in *IEEE Transactions on Consumer Electronics*, vol. 58, no. 4, pp. 1375-1383, November 2012, doi: 10.1109/TCE.2012.6415009.

[14] N. Kim, S. Jeon, H. J. Shim, B. Jeon, S. -C. Lim and H. Ko, "Adaptive keypoint-based CU depth decision for HEVC intra coding," *2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Nara, Japan, 2016, pp. 1-3, doi: 10.1109/BMSB.2016.7521923.

[15] M. U. K. Khan, M. Shafique and J. Henkel, "An adaptive complexity reduction scheme with fast prediction unit decision for HEVC intra encoding," *2013 IEEE International Conference on Image Processing*, Melbourne, VIC, Australia, 2013, pp. 1578-1582, doi: 10.1109/ICIP.2013.6738325.

[16] X. Shen, L. Yu and J. Chen, "Fast coding unit size selection for HEVC based on Bayesian decision rule," *2012 Picture Coding Symposium*, Krakow, Poland, 2012, pp. 453-456, doi: 10.1109/PCS.2012.6213252.

[17] Q. Zhang, Y. Wang, L. Huang and B. Jiang, "Fast CU Partition and Intra Mode Decision Method for H.266/VVC," in *IEEE Access*, vol. 8, pp. 117539-117550, 2020, doi: 10.1109/ACCESS.2020.3004580.

[18] Z. Feng, P. Liu, K. Jia and K. Duan, "HEVC Fast Intra Coding Based CTU Depth Range Prediction," *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, Chongqing, China, 2018, pp. 551-555, doi: 10.1109/ICIVC.2018.8492898.

[19] Y. Li, Z. Liu, X. Ji and D. Wang, "CNN Based CU Partition Mode Decision Algorithm for HEVC Inter Coding," *2018 25th IEEE*

International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 993-997, doi: 10.1109/ICIP.2018.8451290.

[20] Z. Chen, J. Shi and W. Li, "Learned Fast HEVC Intra Coding," in *IEEE Transactions on Image Processing*, vol. 29, pp. 5431-5446, 2020, doi: 10.1109/TIP.2020.2982832.

[21] T. Li, M. Xu and X. Deng, "A deep convolutional neural network approach for complexity reduction on intra-mode HEVC," *2017 IEEE International Conference on Multimedia and Expo (ICME)*, Hong Kong, China, 2017, pp. 1255-1260, doi: 10.1109/ICME.2017.8019316

[22] S. Paul, A. Norkin and A. C. Bovik, "Speeding Up VP9 Intra Encoder With Hierarchical Deep Learning-Based Partition Prediction," in *IEEE Transactions on Image Processing*, vol. 29, pp. 8134-8148, 2020, doi: 10.1109/TIP.2020.3011270.

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

[24] Z. Liu, X. Yu, S. Chen and D. Wang, "CNN oriented fast HEVC intra CU mode decision," *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, QC, Canada, 2016, pp. 2270-2273, doi: 10.1109/ISCAS.2016.7539036.

[25] T. Katayama, K. Kuroda, W. Shi, T. Song and T. Shimamoto, "Low-complexity intra coding algorithm based on convolutional neural network for HEVC," *2018 International Conference on Information and Computer Technologies (ICICT)*, DeKalb, IL, USA, 2018, pp. 115-118, doi: 10.1109/INFOCT.2018.8356852.

[26] 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.

[27] H. Amer, A. Rashwan and E. -h. Yang, "Fully Connected Network for HEVC CU Split Decision equipped with Laplacian Transparent Composite Model," *2018 Picture Coding Symposium (PCS)*, San Francisco, CA, USA, 2018, pp. 189-193, doi: 10.1109/PCS.2018.8456290.

[28] D. Dang-Nguyen, C. Pasquini, V. Conotter and G. Boato “RAISE: a raw images dataset for digital image forensics”, *in Proc. 6th ACM Multimedia Systems Conference (MMSys '15). Association for Computing Machinery*, New York, NY, USA, 2015, pp. 219–224, doi:https://doi.org/10.1145/2713168.2713194

[29] I. Goodfellowm, Y. Bengio, A. Courville, "Deep Learning", MIT Press, 2016, available at :https://www.deeplearningbook.org/

[30] E. Agustsson and R. Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA, 2017, pp. 1122-1131, doi: 10.1109/CVPRW.2017.150, dataset available at : https://data.vision.ee.ethz.ch/cvl/DIV2K/

[31] Martín Abadi et Al. 'TensorFlow: Large-scale machine learning on heterogeneous systems", https://www.tensorflow.org, 2015. M. Abadi et. Al. ”TensorFlow: a system for large-scale machine learning” In Proc. 12th USENIX conference on Operating Systems Design and Implementation, Nov 2016, USA, pp. 265–283.

[32] Good, I. J. “Rational Decisions.” Journal of the Royal Statistical Society. Series B (Methodological), vol. 14, no. 1, 1952, pp. 107–114. JSTOR, www.jstor.org/stable/2984087

[33] P. Diederik, J.B. Kingma, "Adam: A Method for Stochastic Optimization" 3rd International Conference for Learning Representations, San Diego, 2015.

[34] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: Advances in Neural Information Processing Systems 32 [Internet]. Curran Associates, Inc.,, Red Hook, NY, USA, 2019, Article 721, pp. 8026–8037. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf