IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 June 2024, accepted 13 July 2024, date of publication 25 July 2024, date of current version 2 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3433424

RESEARCH ARTICLE

Complexity-Efficiency Control With ANN-Based CTU Partitioning for Video Encoding

MATEUSZ LORKIEWICZ[1], OLGIERD STANKIEWICZ[1], (Member, IEEE), MAREK DOMAŃSKI[1], (Life Senior Member, IEEE), HSUEH-MING HANG[2], (Life Fellow, IEEE), AND WEN-HSIAO PENG[2], (Senior Member, IEEE)

[1]Institute of Multimedia Telecommunications, Poznań University of Technology, 60-965 Poznań, Poland
[2]Institute of Data Science, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan

Corresponding author: Mateusz Lorkiewicz (mateusz.lorkiewicz@put.poznan.pl)

The work was partially supported by Ministry of Science and Higher Education of Poland. The achievements described in the paper partially result from a former joint project of Ministry of Science and Technology (MOST) of Taiwan and National Centre for Research and Development (NCBR) of Poland.

ABSTRACT The application of machine learning to video coding is generally studied in two main approaches: end-to-end video coding using deep neural networks and classic hybrid codecs with individual tools implemented using such networks. This work exploits the latter approach, where a trained Artificial Neural Network (ANN) is used for fast implementation of the search for the partitioning of Coding Tree Units (CTU) into Coding Units (CUs) and Prediction Units (PUs). The proposed approach differs from the previous ones, among other factors, by the application of an ANN with probabilistic soft outputs, which allows for assessing the probability of particular division patterns. This is followed by a decision algorithm that selects more than one candidate division pattern from the most probable patterns provided by ANN. Two variants of the ANN model are considered, out of which the extended one involves deep partitioning at the PU-level. The number of selected division pattern candidates is controlled by a single parameter – uncertainty range – that allows for choosing a trade-off between the complexity and the loss of coding efficiency. This feature is particularly important for broadcasting applications, where processing must be adjusted to the current computational load and resource availability. The experiments demonstrate that the proposed approach yields good results using much smaller ANNs than those described in the references. This makes the proposed approach well-suited for using on edge servers where GPU may not be available. The source code for the respective HEVC codec software implementation is provided.

INDEX TERMS CTU partitioning, encoder control, HEVC, neural network, video compression.

I. INTRODUCTION

The High Efficiency Video Coding (HEVC) [1] is currently the most widely used MPEG video coding standard, especially for television [2]. HEVC support is mandatory in TV sets in several regions of the world, e.g. in Europe. As compared to its predecessor, Advanced Video Coding (AVC) [3], [4], HEVC achieves substantial compression gains at the cost of higher encoding complexity. The envisioned successor to HEVC, Versatile Video Coding (VVC) [5], is even more complex [6], impeding its widespread adoption. Currently, HEVC is still most prevalent in hardware implementations, especially in mobile devices, where power consumption is a challenge [7]. Consequently, research focused on the optimization and simplification of HEVC is still of high practical significance [8].

Modern video codecs, and HEVC in particular, search for the optimum encoding mode among a huge variety of modes for each block of pixels in each frame. In the classical approach, the (nearly) optimal encoding is estimated using the rate-distortion optimization (RDO), which performs an extensive search by speculative encoding with different modes. This approach significantly contributes to the complexity of an encoder. A significant share of this complexity

The associate editor coordinating the review of this manuscript and approving it for publication was Xueqin Jiang.

is related to the estimation of the division pattern for pixel blocks, known as Coding Tree Unit (CTU). In HEVC, the partitioning is described by quaternary tree of Coding Units (CUs) [9]. Methods for efficient encoder control, and therefore reduction of this complexity constitute a research problem addressed in this paper.

The abovementioned problem of CTU partitioning complexity is the subject of this paper. The problem is considered for Intra frames, whose encoding requires numbers of bits exceeding other frame types. An original solution, based on an artificial neural network (ANN) and a probabilistic soft decision algorithm, is proposed. Such a method is very desirable, especially in modern mobile devices where dedicated hardware for neural-network processing is available (e.g. Apple smartphones) [10]. In combination with the proposed method, this hardware could be used to reduce the complexity and power consumption of HEVC encoding. Along with the experimental results, the respective software for comparative research is provided.

The organization of the paper is as follows. Section II presents the current state of the art, starting with a detailed description of the problem of CTU partitioning. This is followed by a survey of related research works and a comparison of these approaches with the one in our work. Next, Section III overviews the main idea of the work along with its novelties. Sections IV and V present the core of the proposed solution, which includes a neural network and a decision algorithm, respectively. Sections VI, VII, and VIII provide a description of the methodology of the experiments and the respective results, demonstrating the efficiency and practicality of the proposed methods. Subjects for future works are highlighted in Section IX, and the paper is summarized in Section X.

II. STATE OF THE ART

A. CTU PARTITIONING IN THE ENCODER

In HEVC, video frames consist of Coding Tree Units (CTUs) [1], which are square blocks of samples, of a size of, for example, 64 × 64 luma samples. Each CTU is further split into square Coding Units (CUs). The actual compression is implemented in these CUs, thus the choice of splits influences the coding efficiency. In this paper, we consider the task of dividing 64 × 64 CTUs; for smaller CTUs, the problem is correspondingly simpler. Each split in the CTU division tree (Fig. 1) always divides an area into four same-sized blocks.

To determine which splits should be made, i.e., which partitioning pattern should be applied, a rate-distortion optimization (RDO) algorithm is usually used in the encoder. This process involves multiple CU encodings to find the modes and encoding tools that yield the best result under given constraints. A typical RDO algorithm follows a top-down search pattern: it starts at the top CU level (Fig. 1 – depth level 0) and steps down to smaller blocks, until all possible divisions are checked. At each level, a CU block of the considered size is encoded to estimate the resulting

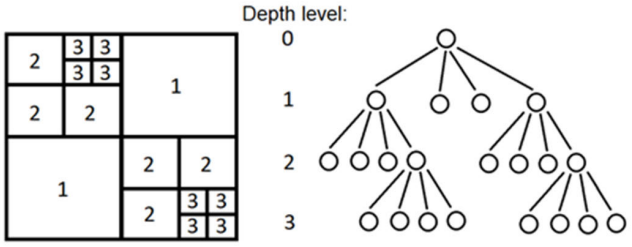

FIGURE 1. An example of the CTU block division pattern with the corresponding splits in a quaternary tree.

image quality and bitstream size. The same procedure is performed for divided blocks, and then the results are compared. If the results for smaller CUs are better, the algorithm repeats the process further for even smaller blocks. It can be noted that for certain content that benefits from smaller block sizes, a significant amount of processing time is wasted on redundant checking of larger block sizes, and vice versa.

For HEVC, the size of CU may vary, from the maximum size of CTU (64 × 64, Fig 2 - depth level 0) to the smallest CU of 8 × 8 luma samples (Fig 2 - depth level 3). A CU block can further be divided into Prediction Units (PUs), for which prediction modes are selected [1], [9]. In Intra mode (which is the focus of this paper), a PU has the same size as its parent CU, except for CUs of the smallest size (8 × 8), where PUs may be either 8 × 8 or 4 × 4 [1] (Fig. 2 – depth 4). Therefore, there are five possible depth levels of partitioning. In total, there are 83522 different division patterns conforming to the HEVC syntax of CU blocks. When the subdivision of 8 × 8 CUs into PU blocks is considered, the number of unique division patterns rises to millions. Therefore, an algorithm for the efficient selection of the division pattern is of high practical importance.

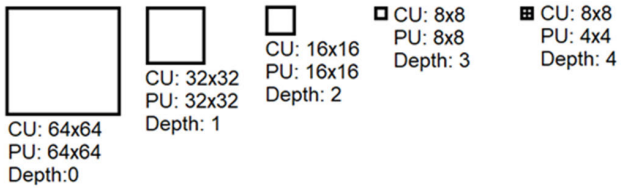

FIGURE 2. Block sizes (in luma samples) in HEVC Intra mode.

B. RELATED WORKS

Optimization of video encoders is an important problem that has prompted extensive research efforts [14]. Several methods utilize specific features identified in the content of the encoded video for decision pruning in HEVC or VVC, thereby reducing encoding complexity. These features may include dense feature maps extracted from the image [15], human perceptual quality [16], sparse keypoints found in a picture [17], encoding statistical models [18], [19], [20], or analysis of the currently encoded block context [21],

[22], [23]. Among the most recent approaches, worth noting is [24], where SVM is used for estimation of prediction modes. Unfortunately the paper focuses on a different part of RDO and was evaluated under conditions (test sequences, training methodology) that make comparison with other related works and with our proposal impossible.

Regarding the problem of mode selection in encoder optimization, several authors have considered the use of machine learning. Erabadda et al. [25] and Zhang et al. [26] proposed the usage of Support Vector Machines (SVM) to accelerate the CU partitioning process and prediction mode estimation. Recently, the use of artificial neural networks (ANNs) has been proposed to replace the RDO or its parts. The key scientific challenges include: 1) the choice of the neural network architecture appropriate to the formats of input and output data, and 2) the analysis (interpretation) of the network outputs. In many works, artificial neural networks are used for pruning the partitioning process by omitting some of the branches in the entire tree of CTU partitioning (Fig.1), e.g. by early termination of the algorithm stopping the processing below a given tree level, or by processing only the most probable branches/levels.

Feng et al. [27] proposed an algorithm that estimates the depth ranges of currently processed CTUs. In particular, an ANN decides to omit some of the highest and lowest level branches in the division pattern tree.

In other works, e.g. Huang et al. [12], Xu et al. [28], and Li et al. [29], ANNs are used to make explicit partitioning decisions at each partition level. The process starts from the top (the largest CU) and at each node, ANN is employed to decide whether to traverse to nodes below or not. The advantage of such a scheme is that it resembles the syntax of HEVC, where partitioning flags are signaled hierarchically. The ANNs used for decisions at various tree nodes can be homogenous or heterogeneous, e.g. Chen et al. [30] and Zhao et al. [31] trained separate ANNs for each partitioning level, whereas Li [32] used a single ANN with multiple outputs. Paul et al. [33] focused on VP9 and used a network with multiple outputs and early termination for the outputs of partitioning levels for better performance. A similar idea of ANN architecture was used by Huang et al. [12] and Xu et al. [28]. For the abovementioned methods, the reductions of encoding times are between 20% and 70%, whereas the bitrate increases are around 1.5% to 3% [34]. Liu [35] presented an application of the aforementioned method to a hardware encoder.

Another approach found in the literature [11], [36], and [37] is to estimate the entire partition pattern at once, before starting the RDO process (e.g. for a given CU). For example, Katayama et al. [36] created an ANN with multiple inputs to estimate the partition pattern for the CTU block currently being processed. In work [37], Ren applied IPB-CNN using CTU samples, employing an approach where the entire partitioning pattern is estimated at once. In our work, we further extend this approach of estimating entire partitioning pattern at once by the usage of Division Tensors as described in Section IV.

Most ANN-based methods focus on estimating partitions for CU blocks. Some researchers additionally address PU partitioning by adding additional algorithms, such as Huang et al. [12] using a Naïve-Bias classifier. Another approach is to include PU block size estimation into the ANN, as seen in Feng et al. [11]. In contrast to this, our approach integrates block size estimation for both CU and PU together as a unified solution. Moreover, we enhance decision flexibility by utilizing decision algorithms on the ANN output.

Apart from using samples from the currently processed CTU only, there are works in which a broader context of image data is used. Katayama et al. [36] used adjacent preprocessed samples and obtained improved results (bitrate increase of 1.8% only). Amera et al. [38], on the other hand, used features from the Laplacian Transparent Composite model. Images from two sources were used as training data: the first few frames from the JCT-VC test set [39] (as in [36]), which was then used for network evaluation) or a separate dataset [11] (e.g. RAISE [40]). Zhao et al. [41] proposed a method for All-Intra scenario that utilizes ANN and probability model to estimate division depth of 8×8 block in VVC based on spatial-temporal coherence. Unfortunately, this approach does not apply to the first encoded frame in sequence and therefore images.

Almost all of the abovementioned methods focus on the Intra coding mode. The Intra mode constitutes a large part of encoder complexity (e.g. 16% [6], [9]), whereas Intra frames comprise a much smaller percentage of all frames (e.g. 2% [6], [9]). Also, still image coding in HEIF (High Efficiency Image File Format) depends on HEVC Intra coding. Therefore, the optimization of the Intra coding mode is a very efficient research direction, also chosen in this work. However, there exist some works that propose solutions also for the Inter mode. For example, the authors of [28] employ recurrent ANN for CUs in P frames. In general, it can be noted that the ANNs employed in most of the works found in the literature are relatively large (e.g. $\sim$1M of parameters [28], [29]). Some authors have been able to achieve promising results with the use of multiple smaller models, each consisting of $\sim$40k parameters [36]. Ren et al. [37] uses the same approach to estimate the matrix of all divisions using a convolutional network but with a shallower and wider ANN, which is still larger than the one proposed in this paper. Also, it shows worse learning results compared to the one proposed in this work and was trained using the JCT VC dataset [39], making the approach incomparable. In our work, we present a solution tailored for a much smaller size of the ANN model.

Most of the methods found in the literature do not provide functionality to control the complexity-efficiency trade-off, with a few notable exceptions. For example, Deng et al. [42] presents an algorithm where CTU divisions are restricted based on complexity modeling, to precisely control encoding

time. In the context of the Intra mode, Li et al. [43] proposed ETH-CNN, which enables control over encoding time reduction through ANN weight pruning. Huang et al. [12] proposed yet another approach that enables encoding time reduction control in a wide range (up to 75%). This however, requires a complex process of adjusting multiple thresholds, which may pose a disadvantage in practical applications.

III. MAIN IDEA OF THE WORK

The goal of this work is to reduce the complexity of the search for the optimum division pattern, thereby reducing encoding time and power consumption associated with rate-distortion optimization (RDO), cf. Fig. 3a, b. This reduction is requested under the condition that the rate-distortion performance of compression would be deteriorated only negligibly.

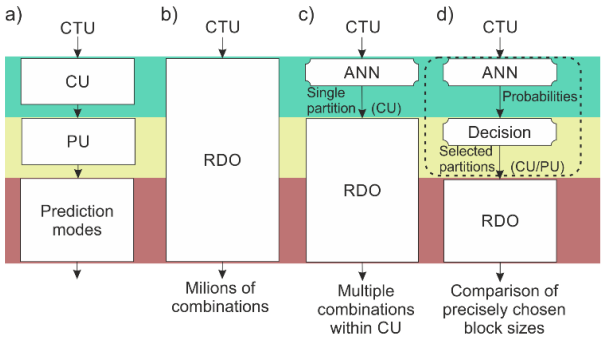

FIGURE 3. Encoder control levels (a) and respective typical solutions: b) classical RDO, c) ANN for CU partitioning, and d) proposed ANN with the soft-decisive algorithm.

As shown in Section II, a promising approach is to use an artificial neural network (ANN) to select a single division pattern to encode a block of pixels (a single CTU in HEVC). In such approach, which we follow in this paper, an ANN is trained to mimic the operation of the RDO algorithm (cf. Fig. 3c). Such solutions offer a specific trade-off between rate-distortion performance (sub optimal mode is selected) and low complexity (only a single division pattern is used/checked in the encoding), which is fixed for a given method/network, e.g. complexity reduction of the encoder of about 60% at the cost of about 3% bitrate increase [11], [12], [13]. Often, only CU-level partitioning is considered.

The idea of this paper (Fig. 3d) is to use an ANN, with specially defined probabilistic soft outputs, that is used for the estimation of the probability of particular division patterns. The usage of an additional decision algorithm (two proposals, with variants, are presented in the paper) enables the selection of more than one candidate division pattern from the most probable patterns. As we will show, such an approach features the following novelties:

- It improves encoding efficiency (in the sense of the rate-distortion characteristics) at only a slight cost of computational complexity, outperforming methods found in the literature.
- It allows flexible control of **the complexity – efficiency trade-off** by a single parameter in the encoder, e.g. reducing the encoder complexity at the cost of lower encoding efficiency (rate-distortion), i.e. lower image quality or increased bitrate. The usage of only a single parameter constitutes an advantage over other methods. Such a feature can be used to adapt the encoder to the current computational load of the system.
- ANN is not directly used to select a partitioning pattern (selection is done by the decision algorithm). Thus even if this ANN is less complex or less trained, the performance may be not degraded as compared to the solution from Fig. 3c (the most common in the literature).

In this paper we demonstrate our solution in the context of HEVC but it can be generalized to other techniques that utilize quaternary tree partitioning. Intra coding is considered due to reasons discussed in Section II.

IV. NEURAL NETWORK ARCHITECTURE

We propose two ANN z 4) which can be used alternatively, depending on the application:

- The Basic model with a division pattern limited to CU-level partitioning.
- The Extended model with deep partitioning (joint for PU and CU).

Both models have the same input format. However, depending on the specific model, a different ANN architecture is used with a different output format.

A. INPUT TO ANN MODEL

Luma samples are widely used to solely control decisions in video encoders (e.g. MPEG HEVC Test Model Software (HM) [44]), whereas chroma is ignored because it was demonstrated that this component provides a negligible amount of distinctive information for CTU partitioning [9]. Such a scheme is also used in this paper in the CTU partitioning procedure. Therefore, the input of ANN is an array of 64×64 luma samples with the values normalized to the range $[0; 1]$.

B. OUTPUT OF ANN MODEL

The output of ANN is designed to convey information about the probabilities of partitioning of the whole CTU.

In HEVC, the partitioning structure can be represented by a depth level of partitioning in particular regions, i.e. an integer from 0 to 4 (Fig. 2), which corresponds to the descending CU/PU block sizes: "0" for 64×64, "1" for 32×32, "2" for 16×16, "3" for 8×8, and "4" for 4×4. In HEVC, 4×4 block is attained by the usage of 8×8 block subdivided into four 4×4 PU blocks. A particular partitioning pattern of the whole CTU can be thus described by arranging depth level values into a matrix with elements corresponding to the smallest possible blocks. We can denote this

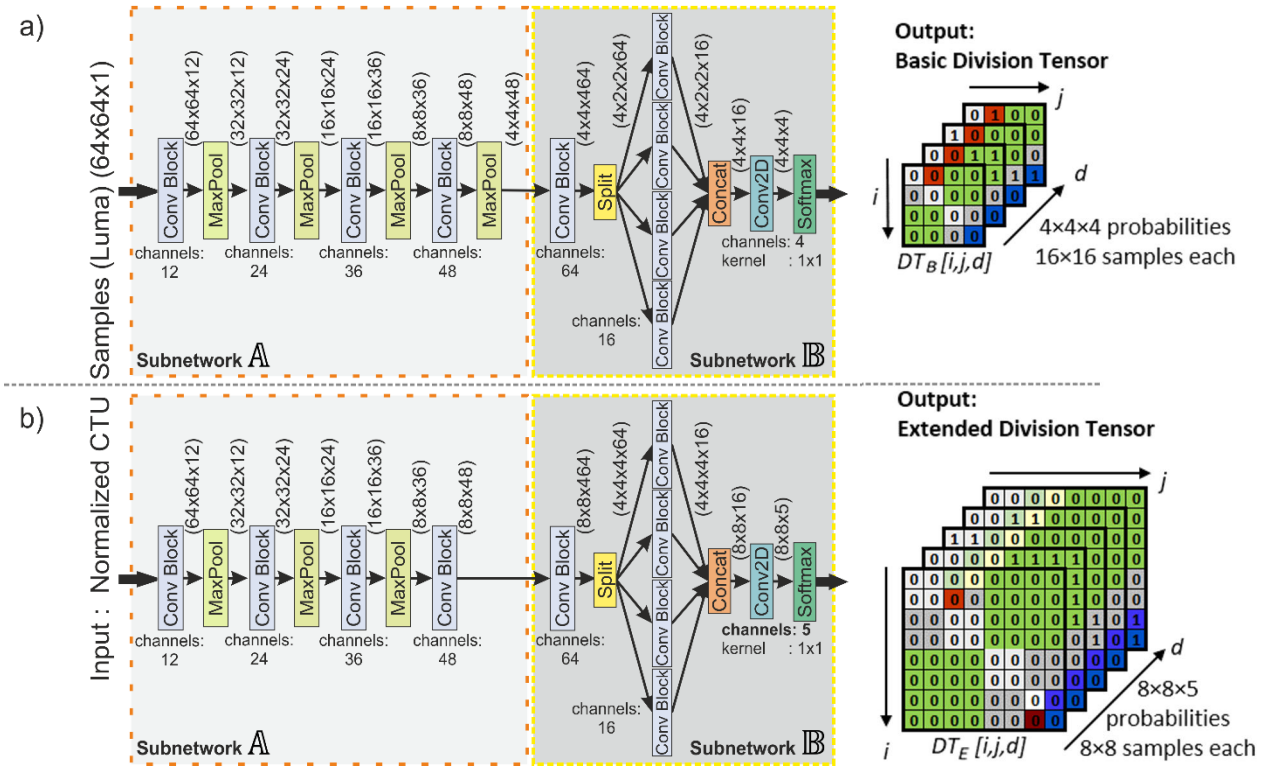

FIGURE 4. Architecture of the ANN: a) Basic model, and b) Extended model.

pattern as $DM\,[i,j]$:

$$DM\,[i,j] = d \text{ where } i,j \in [0;\,15]\,,\, d \in [0;\,4]\,, \qquad (1)$$

where d is the depth of partitioning (cf. Section II).

In the case of the Basic model, where partitioning into only CU blocks is considered, depth level values have a range limited to [0; 3], and therefore the representation by a 16×16 array $DM\,[i,j]$ is redundant. The depth level can be attributed to an area of 16×16 samples, resulting in a Basic Division Matrix $DM_B\,[i,j]$:

$$DM_B\,[i,j] = d \text{ where } i,j \in [0;\,3]\,,\, d \in [0;\,3]\,. \qquad (2)$$

In the case of the Extended model, where CU and PU splits are considered jointly, the depth level value can be attributed to an area of 8×8 samples, yielding an Extended Division Matrix, $DM_E\,[i,j]$:

$$DM_E\,[i,j] = d \text{ where } i,j \in [0;\,7]\,,\, d \in [0;\,4]\,. \qquad (3)$$

The aforementioned simplification of $DM\,[i,j]$ to $DM_B\,[i,j]$ and $DM_E\,[i,j]$, respectively, allows more efficient training of the ANN model due to lesser redundancy and more compact representation (fewer network parameters).

The Division Matrices defined above can be used to characterize a single partitioning pattern. To represent the probabilities of division levels for a corresponding CTU area, we employ division tensors with an additional dimension. This dimension uses the depth level as an index pointing to the probability $prob_{i,j,d}$. This value reflects the likelihood that the corresponding area should be in a particular block size. In the general case of $DM\,[i,j]$, we define Division Tensor $DT\,[i,j,d]$ as follows:

$$DT\,[i,j,d] = prob_{i,j,d} \text{ where } i,j \in [0;\,15]\,,\, d \in [0;\,4]\,. \quad (4)$$

In the case of the Basic model (CU-level partitioning only), we define Basic Division Tensor, $DT_B\,[i,j,d]$ as follows:

$$DT_B\,[i,j,d] = prob_{i,j,d} \text{ where } i,j \in [0;\,3]\,,\, d \in [0;\,3]\,. \quad (5)$$

For the Extended model (joint CU and PU partitioning), we define Extended Division Tensor, $DT_E\,[i,j,d]$:

$$DT_E\,[i,j,d] = prob_{i,j,d} \text{ where } i,j \in [0;\,7]\,,\, d \in [0;\,4]\,. \quad (6)$$

For a better understanding of the presented tensors, we depicted them in Fig. 4.

C. ANN ARCHITECTURE FOR THE BASIC MODEL

For the Basic Model, we adapted the ANN architecture from the paper [11], as it requires relatively low computing power and provides good rate-distortion performance. The architecture has been modified to output $BD_T\,[i,j,d]$ with probabilities instead of $DM_B\,[i,j]$, which is a significant modification, as this allows the application of decision algorithms. In this paper we refer to this modified model as the Basic model.

The Basic model (Fig 4a) is composed of two subnetworks $\mathbb{A}$ and $\mathbb{B}$. The first one ($\mathbb{A}$) contains Convolution Blocks

(consisting of convolution with batch normalization and ReLU activation) and Max Pooling operations (used to reduce the size of the feature maps to 4 × 4 in the first two dimensions). Subnetwork $\mathbb{B}$ mimics the quaternary tree with blocks that are composed of Convolution Blocks. The second convolution is divided into 4 parallel separate Convolution Blocks to ensure that filters estimate features based on data that correspond to 32 × 32 blocks. Similarly. the kernel size for the last convolution was set to 1 × 1 to estimate features based on 16 × 16 area. Additionally, the number of filters is set to be the same as the number of output probabilities for the smallest considered area, which is 4 in this case, to get a features map shape of 4 × 4 ×4. At the end, the Soft-max function is applied over the last dimension. This ensures that the output can be interpreted as the probabilities of particular division depths for corresponding CTU areas.

The network is designed in such a way that all its outputs estimate partitioning depth for the corresponding CTU areas. Therefore, convolutions are good option to find internal dependencies that should be similar between samples in particular regions in CTU. Just like in the original ANN architecture [11], the number of filters is set to 12, 24, 26, and 48 in the consecutive Convolution Blocks in Subnetwork $\mathbb{A}$ and 64, 16, and 4 in Subnetwork $\mathbb{B}$. The kernel size was set to 3×3 except for the last convolution where it was 1×1. Such a set of filters provides good training results (∼72% accuracy on average) without overfitting. The presented network is relatively small: 91600 weights, 6.76M sums and additions, and relatively shallow. The detailed results are presented in Section VII.

D. ANN ARCHITECTURE FOR THE EXTENDED MODEL

The Extended model aims to additionally consider deep partitioning (Section II). In HEVC, deep partitioning is attained through PU partitioning.

To adjust the architecture of the Basic model (Fig. 4a) to allow joint CU and PU partitioning in the Extended model (Fig. 4b), it was necessary to remove one of the MaxPool operations and increase the number of filters in the last convolution of Subnetwork $\mathbb{B}$ to 5. The last MaxPool in Subnetwork $\mathbb{A}$ was chosen to ensure the smallest computational complexity increase. The model after modification has 91617 weights and requires 8.54M multiplication-addition operations to calculate the output.

Regarding training of the ANN for Extended model, the accuracy dropped on average to 62%, both for training and validation subsets. Considering that the problem is much more complex (8 × 8 × 5 output instead of 4 × 4 × 4), this result is deemed entirely satisfactory.

The Extended model (for joint CU and PU block size estimation) requires 1.78M more multiplications-addition operations (as compared to the Basic model), which slightly impacts the processing time (see details in Section VII).

V. DECISION ALGORITHMS

Each of the described network architectures enables estimation of the Division Tensor with probabilities of individual block sizes (depths) for a given CTU area. The network itself does not constrain its output to conform with the bitstream syntax. Therefore a decision algorithm is required to determine the most efficient and syntax-conformant partitioning pattern. For this purpose, in this paper, we consider two alternative algorithms: index-based (denoted as Alg-Idx) and probability-based (denoted as Alg-Prb), described in detail in the subsections below. In both of them, a CTU is partitioned in a quaternary tree manner, from top to bottom (Fig. 1). At each node of the partitioning tree, the algorithm decides, whether a split should be made or not (always in a syntax-conformant manner).

A standalone ANN with a simple common output decision function (Softmax in our case) does not always give results that allow an unambiguous choice of a particular split in a quaternary tree. Here, we propose a more sophisticated output decision algorithm that process the ANN output. In the hard-decisive version of the decision algorithms, only one partitioning scheme is chosen, which may be suboptimal. To address this deficiency, we propose soft-decisive variants of these algorithms. Instead of a hard decision in the case of an uncertain split, we allow the decision algorithm to check two options of block size selection: 1) block size at the currently considered depth level and 2) block size at one level below (twice smaller block). Such sizes of blocks are called as adjacent (symbol ↔ in Table 1) in quaternary tree division hierarchy, where CU and PU blocks are considered jointly. Only two adjacent sizes of blocks are considered because checking more options would be too time-consuming.

TABLE 1. Adjacent (symbol ↔) sizes of blocks used in soft-decisive variants of the decision algorithms.

Basic Model	Extended Model
CU 64×64 ↔ CU 32×32	CU 64×64 ↔ CU 32×32
CU 32×32 ↔ CU 16×16	CU 32×32 ↔ CU 16×16
CU 16×16 ↔ CU 8×8 (PU 8×8)	CU 16×16 ↔ CU 8×8 (PU 8×8)
	CU 8×8 (PU 8×8 ↔ PU 4×4)

The proposed soft-decisive variants of the decision algorithms are controlled by only one parameter, called uncertainty range, described in the following sections. Here we underline that despite the same value range, this parameter is defined differently for each decision algorithm. It means that the given value of the coefficient affects each decision algorithm differently, e.g. how often it will decide to consider two block sizes, instead of one, for the considered CTU.

A. INDEX-BASED DECISION ALGORITHM

The first decision algorithm, referred to as Index-based (Alg-Idx) one, finds an index d of the maximum value in a Division Tensor (DT_B or DT_E, for Basic and Extended

models respectively) for each i, j within appropriate ranges:

$$DM\,[i,j] = ArgMax_d\,\{DT_K\,[i,j,d]\}\,, \tag{7}$$

where $ArgMax_d F\,(d)$ is an argument d for maximal value of $F\,(d)$.

The resultant division matrix $DM\,[i,j]$ is traversed in a top-bottom fashion: starting from the biggest block size (depth level $d = 0$), at each level, making a decision whether to stop at the current level or go to the deeper level $d + 1$, and finally to the smallest block size (e.g. $d = 4$).

At each step of the algorithm, at depth level d, a block of size $N \times N$ is considered, which starts at indices i, j in the Division Tensor. The next step is to decide whether to split or not. For this C (8) is calculated: all values inside the considered block that correspond to the considered depth d are counted. The result is normalized to the range [0; 1] by dividing by N^2:

$$C = {}^{1}\!\Big/\!_{N^2} \cdot \sum\nolimits_{m\in[i,i+N-1]} \sum\nolimits_{n\in[j,j+N-1]} Iv(A\,[m,n] = d), \tag{8}$$

where $Iv\,(\cdot)$ is the Iverson function, such that $Iv\,(true) = 1$ and $Iv\,(false) = 0$.

Value of $C > 0.5$ means that most of the areas inside the considered block voted for the considered depth level d (and not greater depth levels). Therefore depth level d is used and greater depth levels are not considered.

Otherwise, if $C \leq 0.5$, the area of the considered block is divided as in the quaternary tree division, and the procedure is repeated for smaller blocks. When the algorithm reaches the smallest possible block size, it uses the appropriate highest depth level value.

The aforementioned hard-decisive variant of Alg-Idx employs a hard-decisive comparison of C with value 0.5. This approach is very similar to the down-sampling method presented in [11], where the network estimates indices of block sizes. Also, a similar idea was used as an ANN correction algorithm in work [13]. In this paper, however, we extend it to a soft-decisive variant, as described below.

In the soft-decisive variant of the Alg-Idx algorithm, additionally, an uncertain split is considered, instead of implying a hard decision on whether to split the block or not. For this, we introduce the uncertainty range parameter $\alpha \in \langle 0; 0.5)$. If C is within the range: $(0.5{-}\alpha; 0.5{+}\alpha)$, it implies that the decision is uncertain and the RDO algorithm is scheduled to try both block sizes and choose the better performing one.

B. PROBABILITY-BASED DECISION ALGORITHM

The second algorithm, referred to as probability-based (Alg-Prb), processes the data in a quaternary manner, in a top-to-bottom fashion just like in the case of the Alg-Idx algorithm.

At each step of the algorithm, at depth level d, a block of size $N \times N$ is considered, which starts at indices i, j in Division Tensor. In order to decide whether to split or not, the probability of selection of each possible depth level L ($L \geq d$) is calculated as a sum of probabilities in the Division Tensor:

$$S_L = \sum\nolimits_{m\in[i,i+N-1]} \sum\nolimits_{n\in[j,j+N-1]} DT_K\,[m,n,L], \tag{9}$$

where DT_K represents output tensor: DT_B or DT_E, for Basic and Extended models, respectively.

If S_d is the maximum among all values S_L, depth level d is used, and greater depth levels are not considered. Otherwise, the area of the considered block is divided as in the quaternary tree division, and the procedure is repeated for smaller blocks until it reaches the smallest possible block.

For the soft-decisive variant of the Alg-Prb algorithm, the sums of probabilities for the current and next smaller block are inspected. We assume that the RDO algorithm should try two adjacent sizes of blocks if conditional probabilities for depth levels d and $d + 1$ have similar values:

$$\frac{|S_d - S_{d+1}|}{S_d + S_{d+1}} \leq \beta, \tag{10}$$

where β is the uncertainty range parameter from the range $\langle 0; 0.5)$; the values close to 0.5 prohibit the algorithm from checking two adjacent sizes, and the results degenerate to such of index-based decisive algorithm.

VI. TESTING CONDITIONS

A. FRAMEWORK FOR NEURAL NETWORK TRAINING

From the dataset DIV2k [45], 800 images are used for training and 100 images constitute a validation subset. In both subsets, the original images were cropped to dimensions divisible by 64 to avoid CTUs exceeding image boundaries.

The images were encoded using the reference HEVC encoder [44] and then decoded to extract Division Matrices in the formats used in the presented models. The training and the validation subsets were prepared separately for every quantization parameter (QP) from the HEVC Common Test Conditions (CTC) [39] configuration: $QP \in \{22, 27, 32, 37\}$. For each QP, the dataset is composed of 522939 CTUs (training) and 66650 CTUs (validation).

The training procedure employed Early Stopping (ES) as a regularization method. The patience of the ES algorithm was set to 4 epochs. After every 10th epoch, the optimizer algorithm was restarted. Depending on the QP value the training procedure lasted from 14 to ∼30 epochs.

The ADAM algorithm [46] was used as an optimizer for training. The learning rate was set to 0.001. As the loss function, Categorical Cross-Entropy [47] was used. The estimation of the starting weight values was performed using Glorot Uniform distribution [48]. The training process of the model was performed using the TensorFlow framework (version 2.5) [49]. During the learning procedure, we used batches of size 64 training samples. All samples within the given batch had the same Division Matrix. The batches were shuffled each learning epoch. In general, this can lead to overfitting the model, but experiments have shown that it leads to better model efficiency without this effect.

B. HARDWARE USED IN EXPERIMENTS

The training process of the model was performed using the Tensorflow framework [49] with cuDNN support. The platform ran on Ubuntu 20.04 OS with Python 3.8.

The computers were equipped with Intel i7 CPUs, 32 GB of RAM, and GPU (nVidia series 16, 20, and 30). The GPUs were used solely for ANN training.

The encoder used in the evaluation was HM software in version 16.23, with an added proposed tool. The implementation of ANN uses PyTorch as a backend [50]. As HM uses only one CPU thread for timing experiments, the ANN was restricted to using only one logical core.

Timing measurements were performed on a single platform consisting of the i7-8700K processor with 32 GB of DDR4 RAM and NVME SSD for data storage, under the control of Ubuntu in version 22.04. Measurements were performed using the std::chrono library. The initialization of ANN was incorporated into encoding time measurement.

C. EVALUATION PROCEDURE AND REFERENCES

The ANN was evaluated on the JCT-VC [39] dataset. Each dataset was encoded with four QP parameter values: 22, 27, 32, and 37, following he CTC "All Intra" experiment scenario [39]. For each QP value, a dedicated model was used. Then, a comparison was made against the original unmodified HEVC test model software (HM-16.23) in terms of rate-distortion and encoding time, as described below. The HM software implements the reference RDO algorithm for HEVC, making it logical choice for reference. Since our method is tailored for HEVC, we compare it with other methods for encoder complexity optimization specific to this standard.

For rate-distortion comparison, the Bjøntegaard metric [34] was used. This yielded an estimation of the average bitrate increase ($BD-RATE[\%]$) and average PSNR increase ($BD-PSNR$[dB]) for the luma component. Values of $BD-RATE$ above zero mean an increase of bitrate (compared to HM). A negative value of BD-PSNR indicates image quality deterioration.

In terms of encoding time comparison, the time saving $TS[\%]$ (11) was used, where the initialization of ANN was included in the encoding time measurements:

$$TS = \left(1 - {}^{T_{tested}}/_{T_{reference}}\right) \cdot 100\%. \tag{11}$$

Positive value of TS indicates reduction of encoding time, whereas negative TS indicates increase of encoding time.

We have also calculated the share of ANN model processing time T_{model} in the total encoding time T_{ANN}:

$$T_{ANN} = \left({}^{T_{model}}/_{T_{encoder}}\right) \cdot 100\%, \tag{12}$$

where T_{model} is the total time of processing data by the ANN model and T_{encoder} is the total encoding time. The decision algorithm execution is negligibly shorter as compared to the encoding process and thus it has not been presented separately in the results.

In the experiments, we also have measured the decoding time. We observed that it did not change noticeably compared to the HM anchor. Additionally, since the state-of-the-art methods do not report decoding times, we also have chosen not to include them in this paper.

The overall performance of the encoder control optimization and simplification methods, such as the one proposed in this paper, depends on two figures of merit: complexity reduction (expressed by TS as defined in (11)) and rate-distortion performance (expressed by $BD-RATE$ or $BD-PSNR$, as mentioned in Subsection VI-C). For instance, a method may achieve significant complexity reduction at the cost of substantial rate-distortion loss, or vice versa: it may experience negligible loss but yield low complexity reduction. Since, there are two figures of merit involved, a straight-forward approach for methods comparison is using the FoM (Figure of Merit) metric [51], [52]:

$$FoM = \left|\frac{BD-RATE}{TS}\right| \cdot 100\%. \tag{13}$$

Conceptually, a small FoM [%] value indicates better overall performance, e.g. a minimal loss of $BD-RATE$ and a high a complexity reduction, expressed by time savings TS.

VII. EXPERIMENTAL RESULTS

This section is organized in the following manner: first (Subsection A) we present the complexity-efficiency control over the encoding process using our approach. We compare our method with other state-of-the-art methods. Secondly (Subsection B) we provide a precise comparison of our method with other state-of-the-art algorithms in terms of bitrate and time. Finally (Subsections C, D, and E) we provide detailed results regarding the performance of the algorithms, e.g. an explanation for the surprising performance of the Basic Model with Alg-Prb when soft-decisiveness is utilized.

A. COMPLEXITY-EFFICIENCY CONTROL OVER THE ENCODING PROCESS

The goal of this paper was to find a method that would allow control over the trade-off between encoding complexity (e.g. expressed as time saving TS) and encoder rate-distortion efficiency. Encoder efficiency can be expressed either in terms of image quality (e.g. Bjøntegaard $BD-PSNR$) or bitrate (Bjøntegaard $BD-RATE$). For the sake of visualization, in this subsection, we focus on the latter.

In Fig. 5, we compare the reduction TS of the processing time with Bjøntegaard bitrate increase $BD-RATE$ for our algorithms and the state–of–the–art solutions. The results have been unified to $BD-RATE$ vs. TS format. The results for our algorithms are attained with various variants of the considered ANN models and decision algorithms. The operating points corresponding to the hard-decisive variants of the algorithms ($\alpha = 0, \beta = 0$) have been marked with black dots, and the rest of the points correspond to soft-decisive variants.

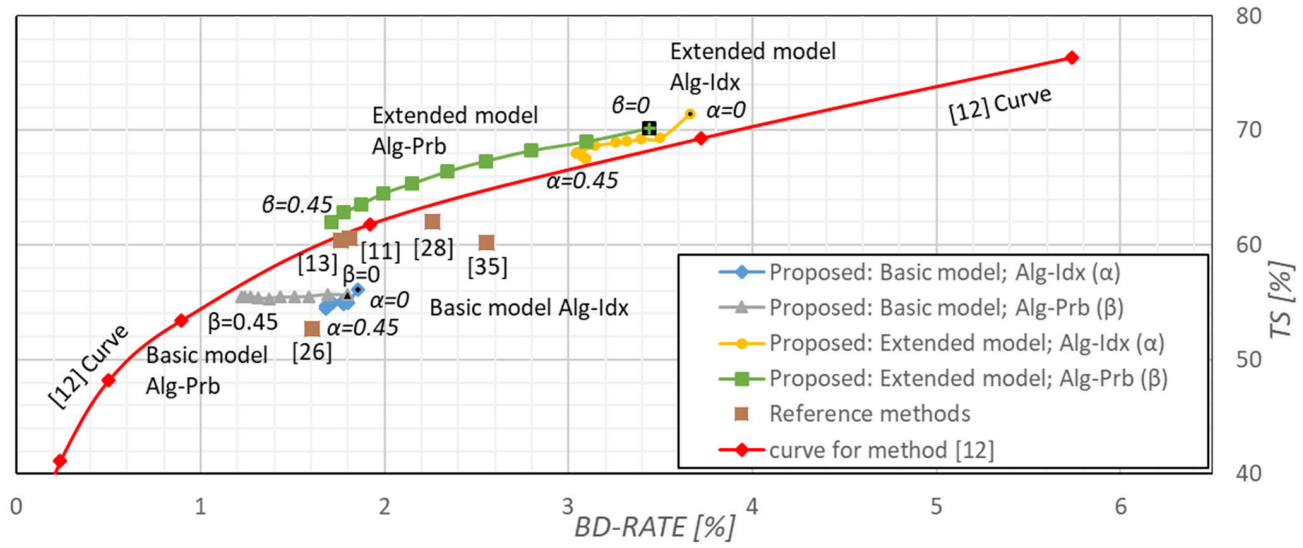

FIGURE 5. Bitrate increase vs. encoder complexity reduction. Comparison of state-of-the-art solutions with the proposed methods: Basic/Extended model, and Index-based (Alg-Idx) and Probability-based (Alg-Prb) algorithms.

The results for the Basic model are competitive with the state-of-the-art solutions. However, the results for this model indicate that decision algorithms do not allow for wide complexity vs. efficiency control. The curve for Alg-Prb (almost flat line) implies that by adding an extension to the decision algorithm the overall bitrate efficiency improves, but the time efficiency remains the same. Also, the higher potential of Alg-Prb over Alg-Idx was shown.

The application of the Extended model provides proper complexity-efficiency control. Adding more precise control over the most costly divisions (PU in the smallest possible CU block) paired with an uncertainty extension of Alg-Prb leads to achieving this. The curve for Alg-Idx (Fig. 5) lies below the curve for the state-of-the-art result from [12], but this algorithm's bitrate efficiency is unsatisfactory.

The results for Alg-Idx prove that basing the decision on probabilities is a much more effective option than [12]. This setup not only provides a reduction (by at least ∼55%) of the encoding time but also grants a ∼10% encoding time range where a user can adjust the compute duration according to their demands.

The Extended model with Alg-Prb showed better results in comparison with the method [12] – the curve for our proposal is below. Here it should be underlined that the wider range of control in [12] is occupied by a much more complicated mechanism that requires the adjustment of multiple parameters. Our method is controlled in a much simpler manner with only a single parameter.

B. COMPARISON WITH THE STATE-OF-THE-ART SOLUTIONS

In Table 2, we present detailed values for bitrate and image quality and in Table 3 - for time performance. We chose [11], [12] ("algorithm #3"), and [13] as the reference solutions and compared their mean results for certain video classes and the whole JCT-VC dataset [39]. One of the chosen references [12] does not present results for two sequences (NebulaFestival and SteamLocomotive). Because of this, we calculated the mean values for two cases: data available in this paper and for the whole dataset. Additionally, as the D subset is no longer used in modern experiments (e.g. VVC test conditions), we delivered results that exclude this data.

The Basic model with Alg-Idx showed similar performance as [13] which is the predecessor of this method. Differences in time performance may come from different experimental platform or model trained anew. Changing the decision method to Alg-Prb in the soft-decisive variant allows to decrease the bitrate loss to 1.22%, outperforming the best reference method [11] by ∼0.5% without compromising encoding time reduction. Here it should be underlined that the Basic model controls only CU divisions, while other referenced methods impact PU block sizes.

The BD-RATE of the Extended model with Alg-Prb in the hard-decisive variant is much worse as compared to the Basic Model. However, it stands out with a substantial encoding time reduction of 70%. This approach outperforms other solutions for joint CU and PU block division control by ∼10%. The soft-decisive variant of Alg-Prb proved that it can enhance the results to a state-of-the-art level and still offer a ∼2% better time reduction. It shows that together with complexity-efficiency control, the proposed method provides top performance even for a not ideally trained model.

Table 4 presents Figures of Merit (FoM) (13) metric values. It can be noticed that the proposed AlgPrb decision algorithms attain the best FoM values, as low as about 2.2. Therefore it outperforms other state-of-the-art techniques.

TABLE 2. Mean bitrate increase (*BD – RATE* [%]) and PSNR increase (*BD – PSNR* [dB]) of the proposed methods, compared with the state-of-the-art averaged over the classes of test sequences.

Sequence class	Basic Model				Extended Model				[11]		[12]		[13]**	
	Alg-Idx $\alpha = 0$		Alg-Prb $\beta = 0.45$		Alg-Prb $\beta = 0$		Alg-Prb $\beta = 0.45$							
	BD-RATE [%]	*BD-PSNR* [dB]	*BD-RATE* [%]	*BD-PSNR* [dB]	*BD-RATE* [%]	*BD-PSNR* [dB]	*BD-RATE* [%]	*BD-PSNR* [dB]	*BD-RATE* [%]	*BD-PSNR* [dB]	*BD-RATE* [%]	*BD-PSNR* [dB]	*BD-RATE* [%]	*BD-PSNR* [dB]
A* (2560×1600)	2.22	-0.123	1.51	-0.084	4.05	-0.224	1.96	-0.109	2.36		1.82		2.20	-0.122
A (2560×1600)	1.99	-0.102	1.19	-0.062	2.95	-0.155	1.40	-0.073	2.06				1.94	-0.099
B (1920×1080)	2.09	-0.079	1.30	-0.051	3.44	-0.137	1.64	-0.064	1.90		1.51		2.02	-0.077
C (832×480)	1.57	-0.086	1.15	-0.063	3.67	-0.204	1.74	-0.097	1.52		2.22		1.55	-0.085
D (416×240)	0.93	-0.061	0.69	-0.045	2.68	-0.176	1.24	-0.083	0.68		1.82		0.93	-0.061
E (1280×720)	2.93	-0.144	1.95	-0.097	4.81	-0.238	2.86	-0.142	2.85		2.26		2.78	-0.138
A*,B,C,D,E ***	**1.87**	**-0.092**	**1.26**	**-0.064**	**3.62**	**-0.187**	**1.81**	**-0.094**	**1.75**		**1.90**		**1.82**	**-0.090**
A*,B,C,--,E ***	2.14	-0.101	1.43	-0.069	3.88	-0.190	1.97	-0.097	2.06		1.92		2.07	-0.099
A ,B,C,--,E ***	2.09	-0.099	1.36	-0.065	3.63	-0.177	1.83	-0.089	2.02				2.02	-0.096
A ,B,C,D,E ***	**1.86**	**-0.091**	**1.23**	**-0.061**	**3.44**	**-0.177**	**1.71**	**-0.088**	**1.76**				**1.81**	**-0.089**

* – only part of class A: PeopleOnStreet and Traffic video test sequences
** – ANN architecture from this work has been adapted in this paper
*** – mean over all sequences in enlisted classes
– not available

TABLE 3. Processing time saving (*TS*) of the proposed methods, compared with state-of-the-art solutions, averaged over the classes of test sequences.

Sequence class	Basic model		Extended model		[11]	[12]	[13]**
	Alg-Idx α=0	Alg-Prb β=0.45	Alg-Prb β =0	Alg-Prb β =0.45			
A* (2560×1600)	56.69	57.46	72.23	63.86	74.25	62.20	
A (2560×1600)	65.94	63.72	74.92	67.26	74.60		55.69
B (1920×1080)	61.38	59.90	72.70	64.08	66.79	64.34	61.38
C (832×480)	47.85	48.85	66.93	58.30	51.24	61.13	60.04
D (416×240)	42.99	43.50	63.40	55.55	39.53	56.13	61.97
E (1280×720)	62.54	62.05	72.92	65.19	70.50	65.40	59.88
A*,B,C,D,E ***	**53.96**	**53.89**	**69.34**	**61.06**	**58.72**	**61.74**	
A*,B,C,--,E ***	57.09	56.85	71.03	62.63	64.21	63.30	
A ,B,C,--,E ***	59.35	58.50	71.86	63.64	65.55		65.55
A ,B,C,D,E ***	**56.08**	**55.50**	**70.16**	**62.02**	**60.35**		60.63

Legend – please refer to Table II

TABLE 4. Figure of merit (FoM) metric for the proposed methods, as compared with state-of-the-art solutions.

Sequence class	Basic model		Extended model		[11]	[12]	[13]**
	Alg-Idx α=0	Alg-Prb β=0.45	Alg-Prb β =0	Alg-Prb β =0.45			
A*,B,C,D,E ***	3.47	**2.34**	5.22	**2.96**	**2.98**	3.08	
A*,B,C,--,E ***	3.75	**2.52**	5.46	**3.15**	3.21	3.03	
A ,B,C,--,E ***	3.52	**2.32**	5.05	**2.88**	3.08		3.08
A ,B,C,D,E ***	3.32	**2.22**	4.90	**2.76**	**2.91**		2.98

* – only part of class A: PeopleOnStreet and Traffic video test seqs.
** – ANN architecture from this work has been adapted in this paper
*** – mean over all sequences in enlisted classes
– not available

C. DETAILED RESULTS – INDEX-BASED DECISION ALGORITHM

In this section, we present and discuss in detail the results for the models and decision algorithms. First, we present the results for Alg-Idx (both models), which can be found in Table 5 A) and B) for rate-distortion and time. Both tables contain results for hard-decisive and soft-decisive variants. The evaluation was made with the use of the JCT-VC datasets [39].

The Hard-decisive variant of Alg-Idx with the Basic model delivers state-of-the-art performance. Unfortunately, the Extended model for the same case yields a twofold bitrate increase. Earlier, Section IV-D, points out worse training accuracy of the Extended model compared to the Basic Model. Despite that, the usage of this model gives ∼15% better time reduction. This means that adding control over PU offers significant speed-up of the encoding process but a wrong decision will result in bitrate increase.

The results show that for this approach using a soft-decisive variant does not have much potential to improve efficiency. A bigger improvement may be seen for the Extended model case, but even for the biggest α parameter value used, this approach does not reduce BD-RATE to a value under 3%. It is caused by a huge information loss due to using only indexes of the most probable division level (ArgMax). ArgMax output format, as in the approach presented in e.g. [11] and [13], lose some output interpretation flexibility disallowing the potential performance gain.

Apart from minor bitrate improvements, ΔT did not change much for the Basic Model. In the Extended model case, the ΔT increases as we increase the uncertainty parameter but it does not exceed 2.5%. A similar observation can be made about ANN processing time – the results are similar as in the case of Basic Model, and are below 30% of total encoding time. It can be concluded that the Alg-Idx does not use the uncertainty mechanism very often. What is more, it does not affect the Basic Model in terms of encoding time.

D. DETAILED RESULTS – PROBABILITY BASED DECISION ALGORITHM

Similarly to the previous section, we show the results for Alg-Prb. Table 6 A) and B) contains results for rate-quality and time successively for both (hard- and soft-decisive) algorithm

TABLE 5. Detailed results for the Index-based decision algorithm.

A) Mean bitrate increase (*BD-RATE*) and PSNR increase (*BD-PSNR*)

Algorithm variant	α Value	Basic model		Extended model	
		BD-RATE [%]	*BD-PSNR* [dB]	*BD-RATE* [%]	*BD-PSNR* [dB]
hard-	-	1.86	-0.091	3.66	-0.187
soft-decisive	0.05	1.80	-0.089	3.50	-0.179
	0.10	1.79	-0.088	3.40	-0.175
	0.15	1.78	-0.088	3.32	-0.171
	0.20	1.77	-0.087	3.26	-0.168
	0.25	1.72	-0.085	3.15	-0.163
	0.30	1.68	-0.083	3.05	-0.158
	0.35	1.68	-0.083	3.04	-0.158
	0.40	1.68	-0.083	3.08	-0.159
	0.45	1.68	-0.083	3.10	-0.160

B) Mean time saving (*TS*) and ANN processing time (T_{ANN})

Algorithm variant	α Value	Basic model		Extended model	
		TS [%]	T_{ANN} [%]	*TS* [%]	T_{ANN} [%]
hard-	-	54.67	21.27	-70.50	28.55
soft-decisive	0.05	54.97	19.36	69.29	27.94
	0.10	55.10	19.32	69.22	27.76
	0.15	54.84	19.28	68.99	27.62
	0.20	55.00	19.25	68.89	27.46
	0.25	54.97	19.11	68.63	27.17
	0.30	54.51	18.99	68.02	26.70
	0.35	54.64	18.97	67.98	26.54
	0.40	54.45	18.91	67.72	26.32
	0.45	54.66	18.84	67.46	26.02

TABLE 6. Detailed results for the Probability-based decision algorithm.

A) Mean bitrate increase (*BD-RATE*) and PSNR increase (*BD-PSNR*)

Algorithm variant	β Value	Basic model		Extended model	
		BD-RATE [%]	*BD-PSNR* [dB]	*BD-RATE* [%]	*BD-PSNR* [dB]
hard-	-	1.80	-0.089	3.44	-0.177
soft-decisive	0.05	1.69	-0.084	3.10	-0.160
	0.10	1.59	-0.079	2.80	-0.145
	0.15	1.51	-0.075	2.55	-0.132
	0.20	1.43	-0.071	2.34	-0.121
	0.25	1.37	-0.068	2.15	-0.111
	0.30	1.32	-0.066	2.00	-0.103
	0.35	1.27	-0.063	1.87	-0.097
	0.40	1.24	-0.062	1.78	-0.091
	0.45	1.23	-0.061	1.71	-0.088

B) Mean time saving (*TS*) and ANN processing share (T_{ANN})

Algorithm variant	β Value	Basic model		Extended model	
		TS [%]	T_{ANN} [%]	*TS* [%]	T_{ANN} [%]
hard-	-	54.20	19.41	69.22	28.44
soft-decisive	0.05	55.72	19.29	69.01	27.38
	0.10	55.51	19.21	68.23	26.53
	0.15	55.45	19.14	67.31	25.81
	0.20	55.44	19.06	66.40	25.14
	0.25	55.30	19.08	65.36	24.60
	0.30	55.36	19.02	64.52	23.99
	0.35	55.52	18.94	63.56	23.40
	0.40	55.51	18.88	62.82	22.87
	0.45	55.50	18.83	62.02	22.33

variants. The value range of the β parameter was [0.05; 0.45] with a step of 0.05.

The values of bitrate increase and time reduction for Hard-decisive variants are similar for Alg-Idx. The difference here is a slightly smaller bitrate increase with a slightly smaller time reduction.

The Basic model performed ∼0.6 % better in terms of bitrate loss without sacrificing the processing time. The percentage of ANN processing time in the encoding process shows that the share is relatively constant. It means that by using a high β factor value, we can significantly improve the performance of the model without compromising the computing time.

For the Extended model, the results are more substantial than for the previously examined decision algorithm. Here one can spot that as the β coefficient value increases, the bitrate loss improves to the point that it matches the results for the Basic model. Additionally, in parallel with state-of-the-art encoding performance, we achieved better computing time results by ∼7% in comparison to the second considered architecture. The ANN processing time to encoding time ratio falls as the β parameter increases.

E. ANALYSIS OF THE SOFT-DECISIVE IMPACT ON DECISION ALGORITHM PERFORMANCE

To explain how the soft-decisive approach impacts the encoding process, the following experiment was performed: the HM was modified, so that the RD optimization considers only two adjacent sizes of blocks (Table 1). During a single experiment, only one option was used. Then we encoded the JCT-VC dataset [39] using HEVC CTC for the "All Intra" experiment scenario [39] and measured the encoding time, bitrate, and image quality. The evaluation results are presented in Table 7.

TABLE 7. Encoding with only two block size options allowed: mean results compared to encoding with HM.

CU Block Size	*TS* [%]	*BD-RATE* [%]	*BD -PSNR* [dB]
64×64 and 32×32	74.29	12.97	-0.665
32×32 and 16×16	73.64	8.25	-0.442
16×16 and 8×8 (PU 8×8)	66.33	5.20	-0.271
8×8 (PU 8×8 or 4×4)	36.68	9.37	-0.415

It can be seen that in a scenario, where only two block sizes are considered, the encoding time reduction is still above state-of-the-art solutions in most cases. The exception here is the smallest CU with two possible PU sizes. One can see that in this case, exploring more than one prediction pattern is relatively time-costly. Regarding soft-decisive variants results presented in previous subsections, we can point out that the examination of the two-block size variants is used relatively rarely so the increase of computational time needed for soft-decisive variants is relatively small. However, it still benefits the bitrate and image quality.

Moreover, the soft-decisive selection between 16 × 16 and 8 × 8 (PU 8 × 8) is more beneficial in terms of encoding time reduction than checking all possibilities for CU 8 × 8. It means the soft-decisiveness of the decision algorithm

leads to avoiding the consideration of a time-costly CU block with 4 × 4 PUs. This explains decreasing bitrate while maintaining the same time reduction for the Basic Model with soft-decisive algorithm variants.

Moreover, the experiment showed that a combination of block sizes: CU 8 × 8 with PU 4 × 4 is the most time-costly encoding option and, if used when not needed, may lead to worse encoding in terms of bitrate. This suggests that the Extended model bitrate results for hard-decisive variants of the algorithm may be caused by poor decisions for the smallest blocks sizes. Following that, the proper selection for smaller blocks may lower the bitrate while increasing time reduction, so it should be overseen during training a model (e.g. more attention than for bigger blocks) or choosing the ANN architecture.

F. EXPLORATION EXPERIMENTS – ADDITIONALL CONTENT AND ENCODING SCENARIO

The proposed models were prepared for the use with HEVC technology, and for standard content, represented by sequences described in Common Test Conditions (CTC) [39], for All Intra configuration. The results for such conditions have already been presented in the previous subsection. Here we present the results of the exploration experiment, where the proposed methods are applied in other sequences and encoding scenarios.

At first, we show results for different types of content. For this purpose, we have chosen sequences from classes A1 and A2 from VVC CTC [53] and class F from HEVC CTC [39]. Classes A1 and A2 are composed of sequences with 4k resolution. The F class from HEVC CTC is dedicated for Screen Content Coding [39]. In Table 8 we present averaged result of evaluation in terms of BD-RATE and *TS*. The results clearly show similar behavior of the proposed method when changing values of the α and β coefficients. It can also be noticed, that the Basic model performs slightly worse on 4k sequences, but maintains good performance on F class. Similarly, the Extended model reports similar results for A1 and A2 sequences as in HEVC-CTC-based evaluation presented earlier. Unfortunately, in the case of HEVC F class, the proposed methods perform noticeably worse in terms of bitrate increase. This may be caused by the poor decision of ANN models on the smallest blocks (4 × 4) level, which may be optimal for CTUs that contain text, which was not subject for training of the presented ANNs, as mentioned in Section VI-A.

To further explore the performance of the proposed method we also run evaluation in Random Access (RA) and Low Delay (LD) scenarios from HEVC CTC [39]. Results are presented in Table 9 A) and B)

The evaluation clearly shows that models trained with data for Intra pictures do not hold their performance when used in Inter slices. The results for RA scenario are slightly better than LD due to wider use of Intra slices. Still, the usage of ANN models specifically trained for RA and LD scenarios would be required. Despite of that, the use of ANN still provides noticeable benefits in terms of reduction of computation time. Additionally, the soft-decisive decision algorithms prove to still allow complexity-efficiency control similarly as in AI scenario.

TABLE 8. Mean bitrate increase (*BD – RATE* [%]) and time saving (*TS* [%]) of the proposed methods, compared with state-of-the-art averaged over the classes A1, A2 and F of test sequences.

	Class	Basic model		Extended model	
		Alg-Idx $\alpha = 0$	Alg-Prb $\beta = 0.45$	Alg-Prb $\beta = 0$	Alg-Prb $\beta = 0.45$
BD-Rate [%]	A1	3.08	2.66	3.77	2.01
	A2	2.8	2.52	3.83	2.05
	HEVC F	1.59	1.49	4.08	3.65
TS [%]	A1	59.65	47.33	68.29	69.23
	A2	55.78	47.27	69.34	69.55
	HEVC F	46.52	65.13	66.18	63.06

TABLE 9. Mean bitrate increase (*BD – RATE* [%]) and time saving (*TS* [%]) of the proposed methods, compared with state-of-the-art.

A) Low Delay scenario

	Class	Basic model		Extended model	
		Alg-Idx $\alpha = 0$	Alg-Prb $\beta = 0.45$	Alg-Prb $\beta = 0$	Alg-Prb $\beta = 0.45$
BD-Rate [%]	A (2560×1600)	20.47	18.96	18.98	11.37
	B (1920×1080)	26.66	25.37	24.86	16.05
	C (832×480)	22.17	21.74	21.95	15.89
	D (416×240)	15.21	14.99	15.41	12.27
	E (1280×720)	66.05	63.66	63.42	44.50
	A ,B,C,D,E	**28.14**	**27.03**	**26.99**	**18.59**
TS [%]	A (2560×1600)	56.57	51.98	55.78	35.21
	B (1920×1080)	52.28	48.65	51.94	28.30
	C (832×480)	52.78	51.20	52.86	34.44
	D (416×240)	47.55	46.90	47.39	34.56
	E (1280×720)	47.27	43.07	46.19	19.37
	A ,B,C,D,E	**51.54**	**48.64**	**51.12**	**30.82**

B) Random Access scenario

	Class	Basic model		Extended model	
		Alg-Idx $\alpha = 0$	Alg-Prb $\beta = 0.45$	Alg-Prb $\beta = 0$	Alg-Prb $\beta = 0.45$
BD-Rate [%]	A (2560×1600)	20.84	19.26	19.70	12.06
	B (1920×1080)	26.64	25.31	25.56	17.07
	C (832×480)	21.71	21.35	22.20	16.43
	D (416×240)	14.52	14.38	15.29	12.13
	E (1280×720)	47.34	45.56	46.93	33.75
	A ,B,C,D,E	**25.18**	**24.16**	**24.87**	**17.45**
TS [%]	A (2560×1600)	55.45	50.93	54.61	34.90
	B (1920×1080)	51.13	47.45	50.69	28.67
	C (832×480)	53.38	51.39	52.99	36.41
	D (416×240)	49.98	48.36	49.34	36.17
	E (1280×720)	45.30	42.01	44.20	19.30
	A ,B,C,D,E	**51.34**	**48.30**	**50.69**	**31.56**

VIII. BROADCASTING APPLICATION SCENARIO

For the sake of assessment of the proposal in practical broadcasting applications, we consider a common problem of encoding/transcoding Full-HD videos (25 frames per second) to HEVC format. We consider the usage of a computer cluster, e.g. equipped with an Intel XEON CPU (32 logical cores). Notably, we do not assume clusters equipped with dedicated GPU, which would drastically impact costs.

Encoding with the original HM software encompasses about 13-18 seconds (depending on QP value) per single frame, on a single i7 CPU core (Table 10). Therefore, real-time processing (25 Frames Per Second) of a single sequence can be attained with the use of about 309 - 447 cores, respectively.

TABLE 10. Original HM encoding time of Full-HD sequences (JVET sequence class B) and considered processing capabilities depending on quantization parameter.

QP	PSNR [dB]	Encoding time/frame [s/frame]	Real-time factor (25-FPS)	Number of 25-FPS real-time streams on 2048 logical cores
22	17.90	17.89	447.25	4.58
27	17.28	17.28	432.00	4.74
32	14.99	14.98	374.50	5.47
37	12.37	12.37	309.25	6.62

Let's assume a computing cluster composed of $N = 64$ CPUs, so there are $C = 64 \cdot 32 = 2048$ logical cores. For high-quality ($QP = 22$, $PSNR = 17.90dB$), it is possible asymptotically to broadcast about 4.6 video encoded, e.g. to serve 4 users. To enable more users to connect and to serve them with dedicated streams (while using the same computational power), possible solution is to decrease the quality of transcoded videos (QP increase), resulting in a reduction of encoding time. E.g. usage of $QP = 32$ would allow encoding of 5.47 streams, serving 5 users, at a cost of quality degradation of about 3 dB.

The usage of the proposed complexity-efficiency control allows much more efficient utilization of computational power and dynamic load control without quality degradation. As shown in Table 6 B), it is possible to attain 62-69% complexity reduction. Therefore in the initial conditions (high quality ($QP = 22$) with the setting of Extended Model, β =0.45, it is possible to process 12.06 (mathematically) streams and thus serve 12 users. When there is a need to serve more users, the cluster can switch to the usage of β =0.05, thus being able to process 14.78 streams and thus serve 14 users, with negligible loss in quality/performance (1 percent point difference of BD-RATE).

IX. FUTURE WORKS

The presented research has been conducted on the basis of HEVC, which is currently the most widespread video coding technology [8]. However, it is foreseeable that VVC, and subsequently the forthcoming technology (tentatively called ECM [54], expected about 2029), will become increasingly significant in the upcoming years. Therefore we intend to focus our future research on adaptation of our technique to VVC. Furthermore, the extension of the proposed method for Inter frames is important. We underline, that presented models and decision algorithms are the first stage of our work. Establishing this basis is necessary in order to start working on newer video standards. A crucial step in such research will involve addressing the differences in the division structure of CTUs. In HEVC this structure is described by a quaternary tree of CUs, as outlined in Section II. In VVC, partitioning follows the same quaternary pattern but with the addition of the multi-type nested tree [6]. Accordingly, the VVC does not utilize PU blocks as separate units. Instead, it transfers prediction mode functionality to CU blocks. Additionally, the maximum size of the CU block is increased, allowing sizes ranging from 128×128 to 4×4, including rectangular shapes [55].

To address the abovementioned differences and adapt the proposed approach to VVC, three modifications are necessary. Firstly, the range of possible depth values must be expanded. The Extended model already considers blocks of size 4×4 as an additional depth level, so to include CU block of size, 128×128, the considered depth level range must be extended from 5 to 6. Secondly, to properly attribute depth level values to an area of 8×8 samples, as described in Section IV-A, the division matrix size should be doubled in each dimension. Thirdly, the decision algorithm must be adjusted to meet the syntax requirements of VVC.

The abovementioned modifications necessitate estimation of Division Tensor similar to (6), but with $i, j \in [0; 15]$ and $d \in [0; 5]$. This can be achieved by changing the ANN architecture similarly to the Extended Model. Therefore, although the application of the proposed approach to VVC is analogous to the case of HEVC, the experimental evaluation would require individual implementation in the VVC codec, which is a considerable workload beyond the scope of this paper.

X. CONCLUSION

This work describes an original approach to CTU Partitioning for Video Encoding, where the ANN does not produce a unique partitioning pattern but rather provides soft outputs that allow assessment of the probability of particular CTU partitioning patterns. This enables the proposed decision algorithms to select more than one candidate division pattern from the most probable ones provided by the ANN. The process is controlled by a single parameter, which allows the complexity-efficiency control of the encoder and does not interfere with the video coding standard syntax.

Among the several approaches presented in the paper, the newly introduced Extended Model of ANN (which considers deep PU-level partitioning) gave the most promising results. When used together with a soft-decisive variant of the proposed Probability-based decision algorithm (Alg-Prb), the model allows wide complexity-efficiency control.

Altogether, it has been shown that with the proposed approach it is possible to adjust the operating point of the encoder between 55% complexity reduction with a 1.5% of bitrate increase, to as much as 70% complexity reduction with a 3.5% of bitrate increase. Such a range of control allows practical applications, where the limited computational power of workstations (e.g. cloud-based) is

utilized to serve dynamically changing numbers of clients for video encoding/transcoding while maintaining the highest attainable encoding efficiency.

Another feature of the proposed method is that the utilization of decision algorithms together with ANNs enables the usage of competitively small ANN models. The main goal of the method, suited also for video surveillance, is the reduction of encoder complexity while maintaining the compression rate. In such application, the proposed method stands out as the best performing and easiest to control.

It is also worth mentioning that the proposed approach can be prospectively applied in VVC, which is considered in Section IX and will be the subject of our future works.

For reproductivity and better comparability of our work, we share our software that has been used to produce the results presented in this paper. The source code with models of presented ANNs can be downloaded from the following website: http://multimedia.edu.pl/HEVC-EVICAD.

ACKNOWLEDGMENT

The achievements described in the article partially result from a former joint project of Ministry of Science and Technology (MOST) of Taiwan and National Centre for Research and Development (NCBR) of Poland.

REFERENCES

[1] *Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 2: High Efficiency Video Coding*, Standard ISO/IEC IS 23008-2, also ITU-T Rec. H.265, Geneva, Switzerland, 2013.

[2] Market Research Intellect. (Jul. 2023). *Global High Efficiency Video Coding (HEVC) Market Size and Forecast*. [Online]. Available: https://www.marketresearchintellect.com/oduct/global-high-efficiency-video-coding-hevc-market-size-and-forecastutm_source=Pulse&utm_medium=019

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[4] *Generic Coding of Audio-Visual Objects, Part10: Advanced Video Coding*, Standard ISO/IEC 14496-10, Mar. 2006.

[5] *Information Technology—Coded Representation of Immersive Media—Part 3: Versatile Video Coding*, Standard ISO/IEC 23090-3, Retrieved 16, International Organization for Standardization, Feb. 2022.

[6] M. Saldanha, G. Sanchez, C. Marcon, and L. Agostini, "Complexity analysis of VVC intra coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 3119–3123, doi: 10.1109/ICIP40778.2020.9190970.

[7] Apple. *Supported Media Formats in Motion*. Accessed: Jun. 29, 2024. [Online]. Available: https://support.apple.com/en-au/guide/motion/motn1252ada3/mac

[8] J. Ozer. (2023). *The State of Video Codecs 2023*. Streaming Media. [Online]. Available: https://www.streamingmedia.com/Articles/ReadArticle.as?ArticleID=158116

[9] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012, doi: 10.1109/TCSVT.2012.2221255.

[10] A. Orhon, A. Wadhwa, Y. Kim, F. Rossi, and V. Jagadeesh. (Jun. 2022). *Deploying Transformers on the Apple Neural Engine*. Apple Machine Learning Reseach, Computer Vision, research area Speech and Natural Language Processing, Highlight. [Online]. Available: https://machinelearning.apple.com/research/neural-engineransformers

[11] A. Feng, C. Gao, L. Li, D. Liu, and F. Wu, "CNN-based depth map prediction for fast block partitioning in HEVC intra coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shenzhen, China, Jul. 2021, pp. 1–6, doi: 10.1109/ICME51207.2021.9428069.

[12] Y. Huang, L. Song, R. Xie, E. Izquierdo, and W. Zhang, "Modeling acceleration properties for flexible INTRA HEVC complexity control," in *Proc. IEEE Trans. Circuits Syst. Video Technol.*, Nov. 2021, vol. 31, no. 11, pp. 4454–4469, doi: 10.1109/TCSVT.2021.3053635.

[13] 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," in *Proc. Signal Process. Symp. (SPSympo)*, LODZ, Poland, Sep. 2021, pp. 177–182, doi: 10.1109/SPSympo51155.2020.9593483.

[14] 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," *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1375–1383, Nov. 2012, doi: 10.1109/TCE.2012.6415009.

[15] Q. Zhang, Y. Wang, L. Huang, and B. Jiang, "Fast CU partition and intra mode decision method for H.266/VVC," *IEEE Access*, vol. 8, pp. 117539–117550, 2020, doi: 10.1109/ACCESS.2020.3004580.

[16] J. Zhao, T. Cui, and Q. Zhang, "Fast CU partition decision strategy based on human visual system perceptual quality," *IEEE Access*, vol. 9, pp. 123635–123647, 2021, doi: 10.1109/ACCESS.2021.3110292.

[17] 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," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Nara, Japan, Jun. 2016, pp. 1–3, doi: 10.1109/BMSB.2016.7521923.

[18] K. Duan, P. Liu, K. Jia, and Z. Feng, "An adaptive quad-tree depth range prediction mechanism for HEVC," *IEEE Access*, vol. 6, pp. 54195–54206, 2018, doi: 10.1109/ACCESS.2018.2871558.

[19] Y. Lu, X. Huang, H. Liu, Y. Zhou, H. Yin, and L. Shen, "Hierarchical classification for complexity reduction in HEVC inter coding," *IEEE Access*, vol. 8, pp. 41690–41704, 2020, doi: 10.1109/ACCESS.2020.2977422.

[20] E. E. Tun, S. Aramvith, and Y. Miyanaga, "Fast coding unit encoding scheme for HEVC using genetic algorithm," *IEEE Access*, vol. 7, pp. 68010–68021, 2019, doi: 10.1109/ACCESS.2019.2918508.

[21] C.-T. Ni, S.-H. Lin, P.-Y. Chen, and Y.-T. Chu, "High efficiency intra CU partition and mode decision method for VVC," *IEEE Access*, vol. 10, pp. 77759–77771, 2022, doi: 10.1109/ACCESS.2022.3193401.

[22] Q. Zhang, Y. Zhao, B. Jiang, L. Huang, and T. Wei, "Fast CU partition decision method based on texture characteristics for H.266/VVC," *IEEE Access*, vol. 8, pp. 203516–203524, 2020, doi: 10.1109/ACCESS.2020.3036858.

[23] H. Li, P. Zhang, B. Jin, and Q. Zhang, "Fast CU decision algorithm based on texture complexity and CNN for VVC," *IEEE Access*, vol. 11, pp. 35808–35817, 2023, doi: 10.1109/ACCESS.2023.3266002.

[24] P. S. Nair and M. S. Nair, "KSVM-based fast intra mode prediction in HEVC using statistical features and sparse autoencoder," *IEEE Access*, vol. 12, pp. 48846–48852, 2024, doi: 10.1109/ACCESS.2024.3382570.

[25] B. Erabadda, T. Mallikarachchi, G. Kulupana, and A. Fernando, "ICUS: Intelligent CU size selection for HEVC inter prediction," *IEEE Access*, vol. 8, pp. 141143–141158, 2020, doi: 10.1109/ACCESS.2020.3013804.

[26] Y. Zhang, Z. Pan, N. Li, X. Wang, G. Jiang, and S. Kwong, "Effective data driven coding unit size decision approaches for HEVC INTRA coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3208–3222, Nov. 2018, doi: 10.1109/TCSVT.2017.2747659.

[27] Z. Feng, P. Liu, K. Jia, and K. Duan, "Fast intra CTU depth decision for HEVC," *IEEE Access*, vol. 6, pp. 45262–45269, 2018, doi: 10.1109/ACCESS.2018.2864881.

[28] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, "Reducing complexity of HEVC: A deep learning approach," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5044–5059, Oct. 2018, doi: 10.1109/TIP.2018.2847035.

[29] Y. Li, Z. Liu, X. Ji, and D. Wang, "CNN based CU partition mode decision algorithm for HEVC inter coding," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 993–997, doi: 10.1109/ICIP.2018.8451290.

[30] Z. Chen, J. Shi, and W. Li, "Learned fast HEVC intra coding," *IEEE Trans. Image Process.*, vol. 29, pp. 5431–5446, 2020, doi: 10.1109/TIP.2020.2982832.

[31] J. Zhao, A. Wu, B. Jiang, and Q. Zhang, "ResNet-based fast CU partition decision algorithm for VVC," *IEEE Access*, vol. 10, pp. 100337–100347, 2022, doi: 10.1109/ACCESS.2022.3208135.

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

[33] S. Paul, A. Norkin, and A. C. Bovik, "Speeding up VP9 intra encoder with hierarchical deep learning-based partition prediction," *IEEE Trans. Image Process.*, vol. 29, pp. 8134–8148, 2020, doi: 10.1109/TIP.2020.3011270.
[34] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD Curves*, document ITU-T SG16/Q6, VCEG-M33, 2001.
[35] Z. Liu, X. Yu, S. Chen, and D. Wang, "CNN oriented fast HEVC intra CU mode decision," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Montreal, QC, Canada, May 2016, pp. 2270–2273, doi: 10.1109/ISCAS.2016.7539036.
[36] T. Katayama, K. Kuroda, W. Shi, T. Song, and T. Shimamoto, "Low-complexity intra coding algorithm based on convolutional neural network for HEVC," in *Proc. Int. Conf. Inf. Comput. Technol. (ICICT)*, DeKalb, IL, USA, Mar. 2018, pp. 115–118, doi: 10.1109/INFOCT.2018.8356852.
[37] W. Ren, J. Su, C. Sun, and Z. Shi, "An IBP-CNN based fast block partition for intra prediction," in *Proc. Picture Coding Symp. (PCS)*, Ningbo, China, Nov. 2019, pp. 1–5, doi: 10.1109/PCS48520.2019.8954522.
[38] H. Amer, A. Rashwan, and E.-H. Yang, "Fully connected network for HEVC CU split decision equipped with Laplacian transparent composite model," in *Proc. Picture Coding Symp. (PCS)*, San Francisco, CA, USA, Jun. 2018, pp. 189–193, doi: 10.1109/PCS.2018.8456290.
[39] *Common Test Conditions and Software Reference Configurations*, document JCTVC-L1100, 12th Meeting, WG11 m28412, Joint Collaborative Team Video Coding (JCT-VC) ITU-T SG16 WP3 ISO/IEC JTC1/SC29/WG11, WG11 m28412, Geneva, Switzerland, 2013.
[40] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *Proc. 6th ACM Multimedia Syst. Conf.*, New York, NY, USA, Mar. 2015, pp. 219–224, doi: 10.1145/2713168.2713194.
[41] T. Zhao, Y. Huang, W. Feng, Y. Xu, and S. Kwong, "Efficient VVC intra prediction based on deep feature fusion and probability estimation," *IEEE Trans. Multimedia*, vol. 25, pp. 6411–6421, 2023, doi: 10.1109/TMM.2022.3208516.
[42] X. Deng, M. Xu, and C. Li, "Hierarchical complexity control of HEVC for live video encoding," *IEEE Access*, vol. 4, pp. 7014–7027, 2016, doi: 10.1109/ACCESS.2016.2612691.
[43] T. Li, M. Xu, X. Deng, and L. Shen, "Accelerate CTU partition to real time for HEVC encoding with complexity control," *IEEE Trans. Image Process.*, vol. 29, pp. 7482–7496, 2020, doi: 10.1109/TIP.2020.3003730.
[44] *2D HEVC Reference Codec*. Accessed: Jun. 29, 2024. [Online]. Available: https://hevc.hhi.aunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.18
[45] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1122–1131, doi: 10.1109/CVPRW.2017.150.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[47] I. Good, "Rational decisions," *J. Roy. Stat. Soc. B, Methodol.*, vol. 14, pp. 107–114, Jan. 1952. [Online]. Available: www.jstor.org/stable/2984087
[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, Sardinia, Italy, vol. 9, 2010, pp. 249–256. [Online]. Available: https://proceedings.mlr.press/v9/orot10a/glorot10a.pdf
[49] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement.*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/di16-abadi.pdf
[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.* Vancouver, BC, Canada: Curran Associates, 2019, pp. 8026–8037, Art. no. 721. [Online]. Available: https://dl.acm.org/doi/10.5555/3454287.3455008
[51] N. Najafabadi and M. Ramezanpour, "Mass center direction-based decision method for intraprediction in HEVC standard," *J. Real-Time Image Process.*, vol. 17, no. 5, pp. 1153–1168, Oct. 2020, doi: 10.1007/s11554-019-00864-z.
[52] B. Heidari and M. Ramezanpour, "Reduction of intra-coding time for HEVC based on temporary direction map," *J. Real-Time Image Process.*, vol. 17, no. 3, pp. 567–579, Jun. 2020, doi: 10.1007/s11554-018-0815-7.
[53] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Shring, *VTM Common Test Conditions and Software Reference Configurations for SDR Video*, document Joint Video Exploration Team (JVET), JVET-T2010, Oct. 2020.
[54] M. Coban, F. Le Léannec, K. Naser, J. Ström, and L. Zhang, *Algorithm Description of Enhanced Compression Model 10 (ECM 10)*, document JVET-AE2025, Oct. 2023.
[55] Y.-W. Huang, J. An, H. Huang, X. Li, S.-T. Hsiang, K. Zhang, H. Gao, J. Ma, and O. Chubach, "Block partitioning structure in the VVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3818–3833, Oct. 2021, doi: 10.1109/TCSVT.2021.3088134.

MATEUSZ LORKIEWICZ received the M.S. degree from the Faculty of Electronics and Telecommunications, Poznań University of Technology (PUT), in 2018, where he is currently pursuing the Ph.D. degree with the Institute of Multimedia Telecommunications, Faculty of Computing and Telecommunications. He is also an Assistant Professor with the Institute of Multimedia Telecommunications, Faculty of Computing and Telecommunications, PUT. He has published several papers (proceedings of international conferences) on audio supervision and video encoding control. He is the co-author of ISO standardization articles in the field of depth estimation and free-viewpoint television. His professional and research interests include audio and video signal processing, video compression, video compression control algorithms, and artificial neural networks.

OLGIERD STANKIEWICZ (Member, IEEE) received the Master of Engineering degree, in 2006, the Ph.D. degree from the Faculty of Electronics and Telecommunications, Poznań University of Technology (PUT), in 2014, and the Habilitation degree from the Faculty of Computing and Telecommunications, PUT, in 2020. Currently, he is an Assistant Professor and the Chief of the Laboratory of Multimedia, PUT. In 2005, he won Second Place in the IEEE Computer Society International Design Competition (CSIDC) held in Washington D.C. He is actively involved in ISO standardization activities, where he contributes to the development of the video coding standards in JCT-3V, MPEG-I, FTV, and VCM. In 2011 and 2014, he was a Coordinator of the development of MPEG reference software for 3D-video coding standards based on AVC. Since 2023, he has been a Coordinator of the development of MPEG reference software for VCM video coding technology. He has published over 100 papers (journals, proceedings of international conferences, and also MPEG/JPEG databases) on free view television, depth estimation, view synthesis, and hardware implementation in FPGA. His professional and research interests include signal processing, video compression algorithms, computer graphics, and hardware solutions. He is the co-inventor in several patents and pending patent applications in European and U.S. patent offices.

MAREK DOMAŃSKI (Life Senior Member, IEEE) received the M.S., Ph.D., and Habilitation degrees from Poznań University of Technology, Poland, in 1978, 1983, and 1990, respectively. Since 1993, he has been a Professor with Poznań University of Technology, where he leads the Chair (Department) of Multimedia Telecommunications and Microelectronics. Since 2005, he has been the Head of Polish Delegation to MPEG. He has co-authored highly ranked technology proposals submitted in response to MPEG calls for scalable video compression, in 2004, and 3D video coding, in 2011. He also led the team that developed one of the very first AVC decoders for TV set-top boxes, in 2004, and various AVC, HEVC, and AAC-HE codec implementations and improvements. He is the author of three books and over 300 papers in journals and proceedings of international conferences. The contributions were mostly on image, video, audio compression, virtual navigation, free-viewpoint television, image processing, multimedia systems, 3D video and color image technology, digital filters, and multidimensional signal processing. He is the co-inventor in several patents and pending patent applications in European and U.S. patent offices. He was the General Chairman/Co-Chairman and host of several international conferences: Picture Coding Symposium, PCS 2012; IEEE International Conference on Advanced and Signal-Based Surveillance, AVSS 2013, European Signal Processing Conference, EUSIPCO 2007; 73rd and 112nd Meetings of MPEG; International Workshop on Signals, Systems, and Image Processing, IWSSIP 1997 and 2004; and International Conference Signals and Electronic Systems, ICSES 2004. He served as a member for various steering, program, and editorial committees of international journals and international conferences.

HSUEH-MING HANG (Life Fellow, IEEE) received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1978 and 1980, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1984. He is currently an Emeritus Professor with National Yang Ming Chia Tung University (NYCU), Hsinchu. From 1984 to 1991, he was with the AT&T Bell Laboratories, Holmdel, NJ, USA, and then he was with NYCU, from 1991 to 2021. From 2006 to 2009, he was appointed as the Dean of the EECS College, National Taipei University of Technology (NTUT). From 2014 to 2017, he was the Dean of the ECE College, NYTU. He has been actively involved in the international MPEG standards, since 1984. He holds 16 patents (Taiwan, U.S., and Japan) and has published over 200 technical articles related to image/video compression, signal processing, and video codec architecture. His current research interests include deep-learning-based image/video processing and compression. He was an IEEE Circuits and Systems Society Distinguished Lecturer (2014–2015). He was a Board Member of the Asia–Pacific Signal and Information Processing Association (APSIPA) (2013–2018). He was a General Co-Chair of the IEEE International Conference on Image Processing (ICIP), in 2019. He was a recipient of the IEEE Third Millennium Medal. He was an Associate Editor (AE) of IEEE TRANSACTIONS ON IMAGE PROCESSING (1992–1994) and (2008–2012) and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (1997–1999).

WEN-HSIAO PENG (Senior Member, IEEE) received the Ph.D. degree from National Chiao Tung University (NCTU), Taiwan, in 2005. He was with the Intel Microprocessor Research Laboratory, USA, from 2000 to 2001, where he was involved in the development of ISO/IEC MPEG-4 fine granularity scalability. Since 2003, he has been actively participated in the ISO/IEC and ITU-T video coding standardization process and contributed to the development of SVC, HEVC, and SCC standards. He was a Visiting Scholar with the IBM Thomas J. Watson Research Center, USA, from 2015 to 2016. He is currently a Professor with the Computer Science Department, National Yang Ming Chiao Tung University, Taiwan. He has authored 90 journals/conference papers, 60 ISO/IEC and ITU-T standards contributions, and holds ten patents. His research interests include learning-based video/image compression, deep/machine learning, multimedia analytics, and computer vision. He was the Chair of the IEEE Circuits and Systems Society (CASS) Visual Signal Processing (VSPC) Technical Committee, from 2021 to 2022. He was a Distinguished Lecturer of the IEEE CASS, from 2022 to 2023; and APSIPA, from 2017 to 2018. He was the Technical Program Co-Chair for 2021 IEEE VCIP, 2011 IEEE VCIP, 2017 IEEE ISPACS, and 2018 APSIPA ASC; the Publication Chair for 2019 IEEE ICIP; the Area Chair/Session Chair/Tutorial Speaker/Special Session Organizer for PCS, IEEE ICME, IEEE VCIP, and APSIPA ASC; and the Track/Session Chair and a Review Committee Member for IEEE ISCAS. He served as the Associate Editor-in-Chief for Digital Communications for the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS and an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He was a Lead Guest Editor, a Guest Editor, a SEB Member for IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS, and a Guest Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS.