This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3164421, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.DOI

VVC/H.266 Intra mode QTMT based CU partition using CNN

SAMEENA JAVAID[1], SAFDAR RIZVI[1], MUHAMMAD TALHA UBAID [2],AND ABDULLAH TARIQ[2]

[1]Department of Computer Sciences,School of Engineering and Applied Sciences, Bahria University, Karachi Campus.
[2]National Center of Artificial Intelligence, KICS, University of Engineering and Technology, Lahore

Corresponding author: Sameena Javaid (e-mail: sameenajaved.bukc@bahria.edu.pk).

ABSTRACT The latest standard for video coding is versatile video coding (VVC) / H.266 which is developed by the joint video exploration team (JVET). Its coding structure is a multi-type tree (MTT) structure. There are two types of trees under the umbrella of the MTT structure. The first one is called a ternary tree (TT) and the second one is a binary tree (BT). Due to the use of brute force quest for residual rate distortion the quad tree and multi-type tree (QTMT) structure of coding unit (CU) split and contributes over 98% of the encoding time. This structure is efficient in coding, however increases computational complexity. Current paper proposes a deep learning technique to predict the QTMT based CU split rather than just the brute-force QTMT method to substantially speed up the time of encoding process for VVC/H.266 intra mode. In the first phase we developed an extensive database containing the ample CU splitting patterns with various streaming videos that is able to encourage the significant decrease of VVC/H.266 complexity by using data driven methods. in the Second phase, in accordance with the dynamic QT-MT structure at numerous locations, we suggest a multi-level exit CNN (MLE-CNN) model with a redundancy removal mechanism at different levels to determine the CU partition. In the third phase, for the training of MLE-CNN model we have established the adaptive loss function and analyzing the both unknown number of partition modes and the focus on RD cost minimization. Finally, a variable threshold decision system is established to achieve the targeted low complexity and RD performance. Ultimately experimental findings show that VVC/H.266 encoding time has reduced to 69.11% from 47.91% with insignificant bjontegaard delta bit rate (BDBR) to 2.919% from 1.023% which performs better than the existing futuristic and modern approaches.

INDEX TERMS Intra mode decision, VVC/H.266, fast coding unit partition, complexity reduction, convolutional neural network

I. INTRODUCTION

Many high-resolution video content and its applications are available nowadays in ultra-high definition (UHD), 4K, and 8K. The continuously increasing high-resolution video stuff needs advanced encoding techniques [1]. The joint video exploration team (JVET) has established the moving picture experts group (MPEG) and video coding experts group (VCEG) to work on the most advanced and high-end video coding standards like VVC/H.266 [2]. The latest version, known as (VTM 8.0) was released at the start of 2020 by JVET. It was a video test model [3]. The coding efficiency of VTM has increased by 40% then the test model (HM) of HEVC/H.265. Earlier, a quad-tree and multi-type Tree (QTMT) was introduced, a multi-type nested tree. The coding unit concept is introduced in the VVC/H.266 instead of prediction unit (PU), transform unit (TU), and coding unit (CU). In QTMT, the shape of CU is rectangular. Leaf nodes of QT are obtained from QT as it is a subpart of the coding tree unit (CTU). The leaf node of QT is minimum as 16 x 16 and maximum as 128 x 128 in size, whereas 128 x 128 in size for CTU. BT and TT cannot divide QT size if 128 x 128 because TT and BT have a maximum root node size of 64 x 64. MTT can divide further leaf nodes of QT. The root node of MTT at this time is the QT leaf node. The division is not allowed anymore after the MTT allowable depth or after the depth limit. Similar case with the width of the MTT node. If the size of the leaf node of BT is minimum or equal and less than or equal to the width of the MTT node, then division will not be considered horizontally. It is also true for the equal or less than the double BT minimum leaf node size. In the vertical case, if the size of the BT leaf node is minimum or equal to the height of the MTT node, then the division is not possible vertically. IT is also true for the equal or less than the double BT minimum leaf node size. Traverse at all the paths of TT, BT, and QT divisions are required by the CU division finally. We calculate the rate-distortion (RD) cost for each depth, and then partition mode selection is based on the lowest RD. Hence computation complexity increased a lot for QTMT structure for the partition of the coding unit compared to HEVC/H.265. In Figure 1 QTMT architecture separates the CTU.

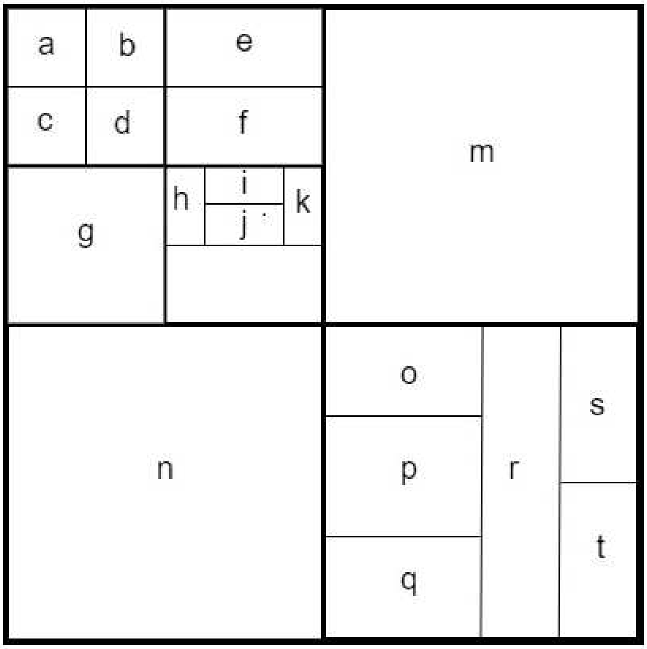


FIGURE 1. MTT Structure of Split Results

Moreover, a significant change in the intra prediction like DC mode is provided, planner 65 modes of prediction instead of 35 intra modes of prediction provided in HEVC/H.265. Using 65 modes of prediction provides the best prediction results. In we have got a better quality of encoding and accuracy, but coding time has increased and computation complexity as well. There are some modern coding tools in which are very useful for the latest video coding standard and provide better coding efficiency [4] [5]. Nevertheless, new advanced coding tools are introduced; as a result, they produce high coding complexity and low speed of video encoding. HM had 19 times less coding complexity than VTM in the test condition of configuration under the (All intra) configuration [6]. Therefore, it has to search for a point where coding performs better with the low coding complexity.

Several researchers applied deep learning techniques to address the H.265/HEVC in terms of coding complexity. Accordingly, constitutional neural network (CNN) based fast algorithm for CU decision mode is proposed in [7] to the complexity of coding. Similarly, texture based encoding techniques in the HEVC/H.265 are introduced, and a high-speed Partition method for the intra CU using the CNN and texture classification is shown in [8] where intra coding complexity is reduced by using the CNN feature of heterogeneous texture feature (HTF). In the HEVC/H.265, fast intra coding methods and fast CU partition are enlightening techniques for the VVC/H.266. These techniques are not so useful for the VVC/H.266 because VVC/H.266 uses the structure that contains the QT structure.

The early termination procedure by using the confidence interval in high-speed VVC/H.266 is presented in [9] for QTBT to detect the RDO redundant partition modes.In [10] paper, a fast decision algorithm is used for the partition of CU using the characteristics of the spatial domain. Their purpose is to reduce the computational complexity of the BT and QT tree structures found in VVC/H.266. The decisions based on Bayesian for the rapid partition are disclosed in the [11]. The main benefit of this work is a correlation sub-coding unit and the main coding unit or parent CU. The output comes in terms of fast VVC/H.266 intra coding. In [12] a fast algorithm is introduced for intra coding early skip way using redundancy removal in the pruning of MTT. Another fast approach for intra coding is presented in [13] to improve VVC/H.266. The fast intra coding speed is achieved by gradient descent and a fast decision system for intra modes. Statistical learning is used to integrate CTU with less complexity-based derivation architecture.

In summary, the dominant research directions for video coding in H.266/VVC are reducing coding time, increasing coding speed, and minimizing the complexity of coding at the cost of coding efficiency decreases a little. Some unique technologies and tools are developed for the H.266/VVC, and some tools and technologies are being improved that were used in the H.265/HEVC are leading to the development of H.266/VVC and increasing its coding efficiency as well. Computational complexity is also improved by improving the tools used for the H.265/HEVC and developing advanced tools for the H.266/VVC. The rest of the paper sections are arranged as follows: Section II contains the related work about video coding standards and VVC standards. Section III contains a detailed discussion on the proposed methodology. Section IV describes the working of CU partitioning and the database description of the CU partition, which is QTMT based. Section V contains the related work about minimizing complexity for the VVC/H.266, and we have proposed the MLE-CNN model for VVC/H.266 CU partition. Section VI displays our results after experiments, verifying our approach accuracy and novelty. In the final section, VII, the papers' conclusion is written.

II. RELATED WORK

Many techniques have been used to improve the partition of the coding block for HEVC/H.265 and the previous one.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3164421, IEEE Access

A. VIDEO CODING STANDARDS

In [14]- [17] the VP9, AV1, AVS2, and HEVC were the main video coding standards before . The joint collaborative team on video coding (JCT-VC) has developed HEVC international video coding standard which becomes a research focus. HEVC uses the simplified partition of the coding tree unit (CTU) and can be divided into two groups. The first one is data-driven and the second is a heuristic approach. To build a statistical model it gets features while encoding in the Heuristic technique. The partition of CTU can be simplified by using the RDO Brute force search with this model. Moreover, in the CTU partition, we can skip the reputation. The encoding time is increased due to the CU partition in HEVC. Few researchers propose the partition of CU at the early stage [18]- [23] . While in the second technique, computational complexity reduction of HEVC is achieved by the data-driven method significantly with the automatically handmade feature extraction and learning. Computational complexity of CTU partition is significantly reduced by the CNN [24], [25]. Using the CTU structure CNN based intelligent decision simplified the process of encoding. Prediction for the intra mode has been designed utilizing multi-classification. CNN uses its layers to predict the best suitable model for the prediction. For the partition of the CU output, an early terminated hierarchical CNN is used [8]. Hence, they minimize its complexity for HEVC.

Comparatively data-driven technique, the heuristic method is less accurate as far as CTU partition accuracy is concerned. High accuracy for the prediction is helpful for RD's complexity and performance. Both of these methods are used to replace the complexity, with learning the different tree types like a quadtree, ternary tree, and binary tree dependent structure of partition block.

B. VERSATILE VIDEO CODING-BASED STANDARDS

The VVC/H.266 CU partition is more flexible due to QTMT and QTBT structures. The complexity of VVC/H.266 is high like HEVC but can be reduced with data-driven and heuristic methods. In [18]- [20] shows the QTBT structure-based work. A fast method for QTBT encoding is proposed in [18], where a temporal frame index uses the full binary tree path. A joint classifier is used to propose a QTBT fast decision method [21]. For the early stop of the partitions of QTBT, a random forest is used in [22] which stops redundant iterations. The CU partition of reduced complexity eliminates the unnecessary intra prediction and partition modes. Similarly, deep-learning-based algorithms for improved coding are described in [18]- [20] for VVC/H.266. The decision for bottom length methods is proposed in [18] and [19] to improve the intra coding method in VVC/H.266 by using a multi-class classification depth range model.

According to data-driven method [23] proposed to predict the length of CU path in every CU 32 x 32 using a CNN to eliminate the search of RDO at intra mode free CU. A different way to apply CNN to predict the CU path length is used at inter mode. CNN input is residual CU because the partitions are correlation-dependent across various frames. The ResNet with 4x4 blocks is used to predict all CU borders directly and accurately predict the CU partition. Hence, they reduce the complexity of VCC [24]. However, the bottom-up decision produces the extra computation when a large CTU produces some significant CU splits and no splits.

III. PROPOSED METHODOLOGY

The QTMT structure is used to split the coding unit, and 67 intra prediction modes are used for the intra prediction. For the fast and accurate prediction, our approach is CNN to predict intra mode CU partition for the VVC/H.266. The novelty of the proposed work is different from existing research in three aspects.

Firstly, we have proposed the latest and novel N-QTMT structure, which is fast and accurate among the existing approaches. In the present work data-driven, QTBT was designed only for the partition of CU in [23], [24], and [25].

Secondly, we use a deep learning approach to extract the features from video frames automatically instead of the handmade features extraction method that is used in [18], [19], and [20].

Lastly, we have designed multi-level CNN. It predicts the big CU partition from previous layers and, using the latter layers' model, predicts the small CU. It is comparatively better than the bottom-up decision approach where CU boundary-based decision is used. So, by using CNN, we can avoid the redundancies and early exit approach.

IV. WORKING OF CU PARTITION WITH DATASET

A. CU PARTITION DESCRIPTION

This section briefly defines the CU partition process in VVC/H.266, which is quite different and flexible from HEVC. A CTU has one CU or repeatedly divides in small squares CU using a quadtree in HEVC standard. The 64 x 64-pixel size is found by default in CTU. In HEVC, an 8x8 minimum size of CU is possible. Whereas in VCC standard, there are many flexible CU partitions. N-QTMT is obtained from the QTBT as far as CU partition is concerned. Moreover, the N-QTMT structure can also divide the CU into square and rectangle forms. Hence these CUs can adopt more complex texture features on the frame of a video. By using the quadtree, the working of the N-QTMT structure can divide the CTU into one CU or smaller CU to get the tiny details from frames. Small CU can further divide themselves using the multi-type tree or the quadtree. Concerning the vertical and horizontal modes, the multi-type tree contains the ternary and binary trees. Figure 2 is the example shown. The range of CU in CTU has the 4x4 as a minimum to 128 x 128 as the maximum in range. Furthermore, the intra mode partition of CU is used for the chrominance and luminance channel individually. A multi-level hierarchical partition method is present to obtain the split CU with the earlier features. The process of splitting 128 x 128 CU into 64 x 64 CU's is shown in Figure 2 as level 1 process. After the level 1 process, 64 X 64 CU's are further split up into 32 x 32 CU's at level

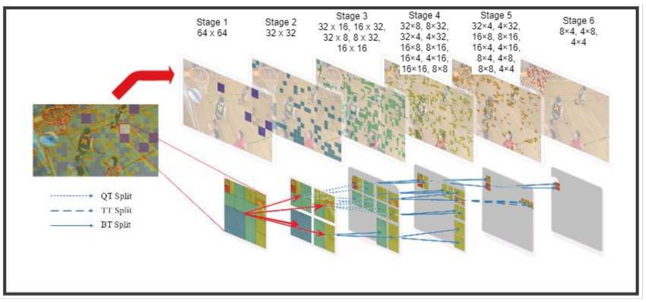


FIGURE 2. The Level of CU Split Separated By Color (Luminance Channel) as It Takes More Encoding Time in VTM Encoder

2 and further on. Stage 1 and 2 support quadtree and non-splitting modes at all six levels. A maximum of six modes (i.e., non-splitting, quadtree, horizontal binary-tree, vertical binary-tree, horizontal ternary-tree, and vertical ternary-tree) are possible for the successive phases minimum width or height for CUs is 4. The possible CU size and split modes concerning levels are illustrated in Figure 2.

The description summarizes that QTMT used in the VVC/H.266 is more advanced and flexible concerning the sizes and types of CU compared to HEVC/H.265.

B. PREPARATION OF DATASET

We have created the database to train and evaluate our model and partition of CU for intra mode. The preparation of data is based on approximate 9000 images [18] and 300 total video sequences [26]- [31]. This collected data contains different resolutions and a variety of contents and is openly available for research purposes. The data is split into three distinct sets for training testing and validation. Total 7200 images and 240 video sequences are used for the training, and 900 images and 30 video sequences are used for the testing. At the same time, 900 images and 30 video sequences are used for validation purposes. A short description of the data is listed in Table 1. VTM 7.0, a reference software, is used to encode images and videos by VVC/H.266. The images were prepared 8x8 in size and multiple in resolution due to VTM support. And video sequences are not more than 10 seconds in length. Four quantization parameters, QP22, QP27, QP32, and QP37, encode the images and video sequences.

After encoding, labels of CU partition are obtained for the Dataset. Labels represent the ground truth information for CU mode division. Each mode can have the six achievable CU modes of the division. The 0 modes can be non-split mode, one mode can be quadtree mode, two modes can be a binary tree for horizontal, three modes can be the binary tree for vertical, and four modes can be a ternary tree for horizontal, and five modes can be a ternary modeled for vertical. For the model training, we saved all CU mode's RD and used it for RD optimization in VVC/H.266.

The corresponding label of the partition of each CTU and RD cost for every CU makes a specimen in our database. All the details about total samples and the total number of CU's are presented in the database are shown in Table 1. The dataset includes 6,809,233 specimens with over 1 billion CUs in total, as shown in the table, providing adequate data for the training of our MLE-CNN model. It describes the database concerning the number of images according to their resolution, the total number of CTUs, and the number of CUs.

TABLE 1. Database Description

Source	Resolution	Num. of Images/Sequences\	Total num. of CTUs	Total num. of CUs
Raw Image Dataset (RAISE) [26]	2880×1920	2,000	2,640,000	372,692,745
	2304×1536	2,000	1,728,000	242,719,640
	1536×1024	2,000	768,000	173,216,005
	768×512	2,000	192,000	58,271,751
	512x256	1,000	110,000	1,281,657.00
Facial Video [28]	1920×1080 (1080p)	6	72,960	9,660,712
Consumer Digital Video Library [29]	1920×1080 (1080p)	30	622,080	139,216,238
	640×360 (360p)	59	40,520	20,699,422
Xiph.org [30]	2048×1080 (2K)	18	95,232	21,108,370
	1920×1080 (1080p)	24	471,840	125,995,868
	1280×720 (720p)	4	30,600	15,913,824
	704×576 (4CIF)	5	12,400	5,411,228
	720×486 (NTSC)	7	10,545	4,765,478
	352×288 (CIF)	25	14,368	8,603,450
	352×240 (SIF)	4	688	753,882
Aggregated		9,182	6,809,233	1,200,310,270

The various split modes that provide the detailed CU proportion are presented in Figure 2. It illustrates the dependency of splits of modes on specific CU. Its range varies from 2 to 6, and rules are already discussed for partition in the previous section, "working of CU partition." Various split modes are unstable, as the ternary tree CU division. Especially mode number four and mode number five are dominant for this. Secondly, in CU, non-split case 0 mode is a primary element for the many CU sizes. So, simple image classification is easy compared to the multi-level Partition of CU because it has not balanced classes with a single output. To deal with this problem, we focus on the CNN model explanation.

V. COMPLEXITY MINIMIZATION FOR INTRA MODE OF VVC/H.266

A. CU PARTITION LEARNING USING MLE-CNN

We will describe the MLE-CNN learning mechanism of CU partition, which is QTMT based in VVC/H.266 The all-viable CU in a certain CTU must be marked down to upward sequence using the RDO brute force search in a standard encoder used for the VVC/H.266. But we propose to predict the CU partition using MLE-CNN. It is a top to bottom and stepwise approach. Hence, we have a fast-encoding process. The overall structure of MLE-CNN is illustrated in next section.

MLE-CNN takes the 128 x 128 CTU luminance channel as an input and follows the conventional layer to get the features map of 128 x 128. We apply a maximum of 6 decision units of divided modes for 6 level CU partitions using these extracted features maps. For the extraction of texture features using

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3164421, IEEE Access

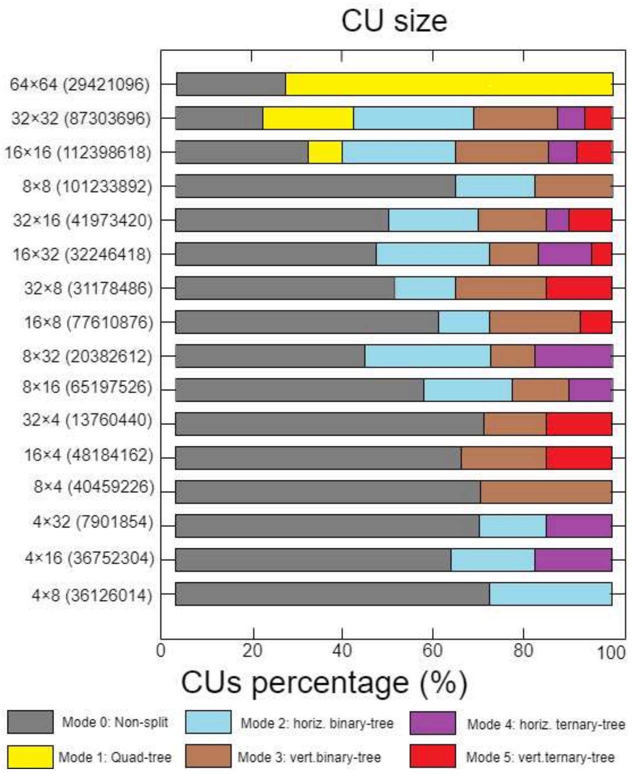


FIGURE 3. Luminance Channel Have Percentage of CU in Various Split Modes

the MLE-CNN, several consecutive convolution layers are required through features map flow. So, these convolution layers are called conditional convolution as well. A small network inputs the features map into it and then predicts the first CU partition. The early termination of the CU partition at the present state means the prediction results are non-split. If not, then the next stage would have the location of every divided CU with the feature maps. The conditional convolution and small network details are described here.

1) Conditional Convolution

The neural network works well if it is trained on several features and deep. That is why we use deep network MLE-CNN and extract more texture features to use. The structure of our network is quite flexible as it depends on the size of CU. The size of CU can be different on some level. We have deep extracted features that are useful. Figure 4(b) shows the Efficient ResNet model. If $\omega \times h$ is the CU size then min(ω, h) is the minimum length of axis for CU. The granularity of the CU partition is computed by it. Assume of CU of the current axis length is smallest are a_p and a_c similarly the processed input feature maps are shown with residual units $\eta_r \varepsilon$ (0, 1, 2) then the formula would be

$$\eta_r = \begin{cases} log_2(\frac{a_p}{a_c}) & 4 \leq a_c \leq 64 \\ 1 & a_c = 128 \end{cases} \qquad (1)$$

So, the residual units contain the convolution operations. These are one stride overlapping and zero-padding operation. In this operation, features map size would not be altered after this sub-network would take the input of residual units with processed feature maps. This flexible design provides a unique property to the MLE-CNN. The residual unit has total 6 indexes which are $\kappa\varepsilon\{1, 2, 3, 4, 5, 6\}$ is define at the time when known CU is satisfy the $\kappa = log_2\{\frac{256}{min(\omega,h)}\}$ We need some parameters for training to satisfy the condition of input in small network which is the same CU must have the same features for all residual units with a value of k index is same.

2) Small-Networks

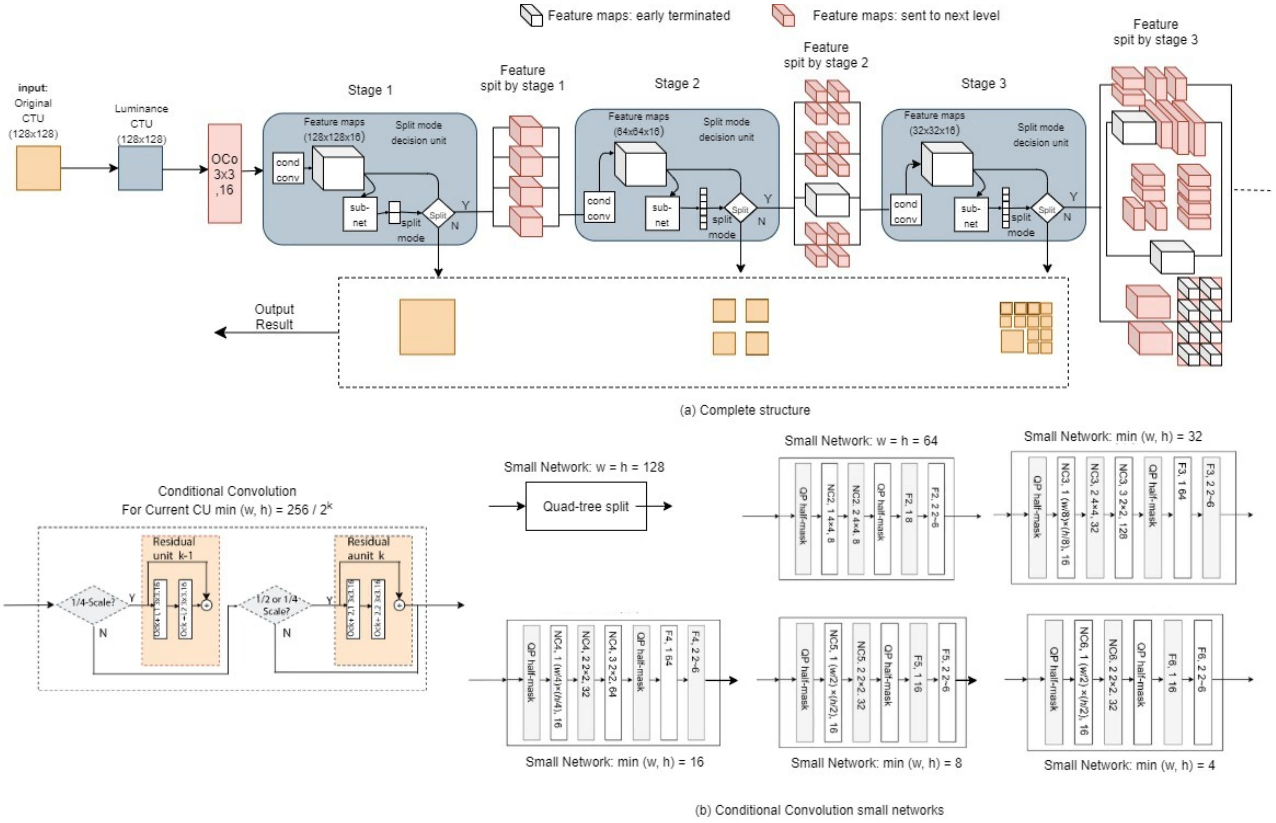
The partition of the CU may be 64 x 64 or smaller in every sub-network; all the connected layers and convolutions process the feature maps to predict where to split. The CU size and sub-network configurations are closely related to each other, as shown clearly in Figure 4(b). For detailed features of the CU partition, two or three convolution layers are used to process the input of feature maps in each sub-network. The Height and width of the Kernel of each layer of convolution are defined as the power of 2. Moreover, the height and width of kernels are equal to the two dimensions of kernel stride. Hence all the kernels are not overlapping. The convolution, which is non-overlapping, is adaptive to no overlapping CU concerning size and location in the final partition. The convolution layers produce the output of feature maps used to get the split mode. The output of the accurate prediction of the vector and range of its length depends on the size of CU. Its range is from 2 to 6. QP also plays an essential role in CU partition. If it is decreased, then split tendency would be increased and vice versa. The external feature is supplemented as QP after the fully connected first convolution layer. The QP-related features are considered in the MLE-CNN, so that we have used the operation of half mask. The half-extracted feature maps are multiplied by the normalized QP value. At the different QP values, the MLE-CNN can learn the partition of CU. In a nutshell, we can say the control of the CU partition procedure depends on the output of the small network. If a non-split CU is predicted, then at the current stage exit procedure is executed. If not, the next stage is processed, and exit is not executed.

Our approach combines the ResNet model with small network operations to design a multi-level design. It is a significant model for the CU partition and is a QTMT base for the VVC/H.266 structure. This design significantly reduces the computational complexity of MLE-CNN because it exits after it finds non-split. It simply skips the redundancies. The experimental results are shown in section 6 for verification and explanation purposes.

B. MLE-CNN TRAINING AND LOSS FUNCTION VALUES

This proposed Model is typically better than other classification designs because of three reasons as follows:

1) The size-dependent split mode, as its range differs from 2 to 6. Details are in section IV
2) Flexible exit strategy for the different exit modes is not fixed. Details are given in figure 3.



(a) Complete structure

(b) Conditional Convolution small networks

FIGURE 4. Structure of MLE-CNN.

3) There is not only a cross-entropy function that exists. In VVC/H.266, there are various RD costs for every split mode.

Therefore, the adaptive loss function must for the MLE-CNN to the properties mentioned above. For the wide and h high CU, all expected split modes are mentioned with M(ω,h). Each element M is indexed of current possible split mode in the M(ω,h), where M will be 0 to 5. We take small batches for training which size-wise is the same. Consider as the index of CU and N as the batch size. The apply the loss function of cross-entropy as follows.

$$L_{CE,B} = -\frac{1}{N}\Sigma_{n=0}^{N}\Sigma_{m=M} * y_{m,n}log(y_{m,n}) \tag{2}$$

The above equation shows binary label ground truth $y_{m,n}$ and $y_{m,n}$ the probability for prediction for the nth CU at m split mode. As far as proportional unbalanced for split mode is concerned, various penalty weights are referred to according to the unbalance. Hence cross-entropy changed to the following.

$$L_{CE} = -\frac{\Sigma_{n=1}^{N}\left(\frac{1}{P_m}\right)^{n}\Sigma_{m=M} * y_{m,n}log(y_{m,n})}{\Sigma_{n=1}^{N}\left(\frac{1}{P_m}\right)^{n}} \tag{3}$$

Where CU proportion of quantitative is shown as P_m and split mode is m $\Sigma_{m=M}P$=1 is satisfactory. Moreover, $\alpha\varepsilon$ [0,1] is not a fixed vector to show the weights of penalty. If α = 0 in this situation there is no penalty is applied $p_m m \varepsilon M$ shows. The other case where α is one means that the inverse of p_m is proportional to penalty weights. It works if MLE-CNN is not trained properly. The split modes of prior distribution are hard to learn the setting. Hence bad prediction accuracy would be observed. The practical use of α is between (0, 1) then we can get the balance between prediction reliability and accuracy. In our case, the value of α is 0.4 tuned by our validation data set. The experiments section is explained in detail parameters tuning.

In equation (3) two addressed properties are explained but the third property which is describing the RD loss function is followed here.

$$L_{RD} = \frac{1}{N}\Sigma_{n=1}^{N}\Sigma_{m=M} * y_{m,n} * \left(\frac{\Upsilon_{m,n}}{\Upsilon_{n,min}} - 1\right) \tag{4}$$

Where $r_{m,n}$ nth CU RD cost at 'm' split mode. $r_{n,min}$ is the current CU's least cost from all the possible split modes. This equation $(r_{m,n}/r_{n,min} - 1)$ shows the cost of RD is normalized. Now if the predicted probability is not accurate or wrong then the penalty would increase. Add the equation (3) and (4) the total loss for MLE-CNN is

$$L = L_{CE} + \beta.L_{RD} \tag{5}$$

Where beta shows the RD cost importance which is a positive scalar value. The MLE-CNN can be trained for the reduced L.

C. MLE-CNN DECISION APPROACH

In an ideal case, the working of the MLE-CNN is to check out the redundancy of CU in the process of original RDO hence reduce the complexity of encoding. Secondly it predicts completely CU partition. But the proposed model can predict the wrong predictions as well. The bad RD performance shows if the partition of a CU is wrongly predicted. We propose the variable-threshold decision method to get a best-balanced point between RD performance and the complexity of encoding. In our proposed method we apply combinations of $\tau_s$with $\tau_{s}{}_{s=2}^{6}$ have 0, 1 value, to all levels of MLE-CNN. The s in the formula shows the level index. Look back on the predicted probability $y_{m,n}$ formula, which shows any number of CU with them split mode in the mini-batch. From set m or candidate mode, we have picked the M at level number 1; the MLE-CNN need not predict VTM-encoder because it is already deterministic. The values of this variable threshold begin from the second level. This $y_{n,max} = max_{m\varepsilon M} y_{n,m}$ formula shows the predicted probability maximum. The CU expected modes are represented m belongs M and $y_{m,n} \geq \tau_s{\cdot}y_{n,max}$ shows the marked encoder RDO probability mode with rest of the modes are ignored. The confidence of the MLE-CNN prediction is controlled by the τ_s threshold. The MLE-CNN has split all the CU modes at the threshold τ_s value = 1. For RDO process 1 mode with this formula $\tau_s$with $\tau_{s}{}_{s=2}^{6}$ is selected for the process. Due to this described setting, we have found the minimum complexity for the encoding. If RD is bad, then there is high encoding complexity observed. On the other hand, if τ_s=0, it means the RDO process has marked all the CU. It means the RD value is not bad with less computational complexity. In an experiment, we observed that the threshold value is set from 0 to 1. Next, we propose the level selection $\tau_s$with $\tau_{s}{}_{s=2}^{6}$ in MLE-CNN. Each stage can have different accuracy values for prediction. Figure 5 illustrates the different threshold values τ_s with respect to the prediction accuracies in MLE-CNN. Section V shows the details. The split size is variable corresponding to the various level and sizes. MLE-CNN can handle classification problems with multiple classes, where top-half accuracy is presented, shows the level 2 accuracy as the best accuracy. At level 6 the accuracy is better after the level 1 accuracy at $\tau_s$with $\tau_{s}{}_{s=2}^{6}$ threshold value. Its' performance is outclassed with large values. Comparatively $\tau_s$with $\tau_{s}{}_{s=2}^{6}$ this is near to zero. For all other stages, unequal accuracy is inadequate.

There are some strategies to pick the variable threshold to ensure prediction accuracy MLE-CNN.

1) Time-Saving Possibility

If $\frac{1}{5}\Sigma_{s=2}^{6}\tau_s \geq 0.4$ then $\tau_2 \geq \tau_6 \geq \tau_3 \approx \tau_4 \approx \tau_5$ average threshold

2) RD Working Good Possibility

If $\tau_2 \geq \tau_4 \approx \tau_3 \approx \tau_5 \geq \tau_6$ is average threshold then $\frac{1}{5}\Sigma_{s=2}^{6}\tau_s \geq 0.4$

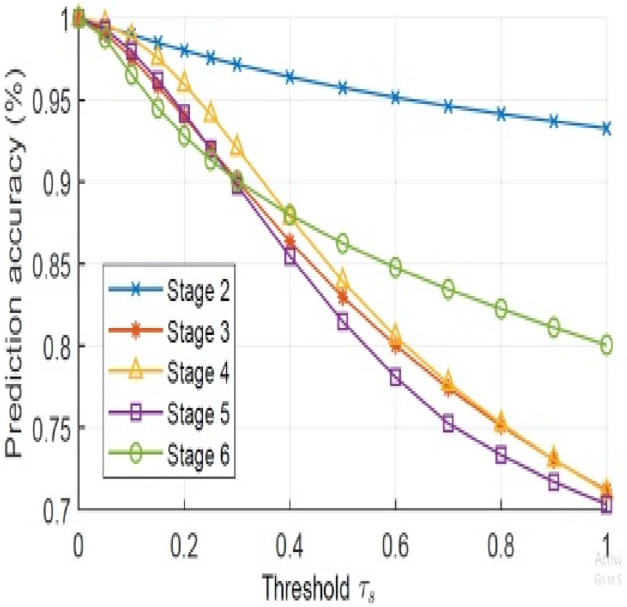


FIGURE 5. MLE-CNN Prediction Accuracy on Validation Data

VI. RESULTS OF EXPERIMENTS

It is the most critical section of this paper. The complexity of intra mode of VVC/H.266 is reduced with the help of experiments of our proposed approach and evaluating our designed model and its performance. The first part of this Section V(A) shows the different settings for the experiment according to our proposed approach. And the second part of this section presents the evaluation of RD performance and complexity concerning the modern and futuristic research works illustrated in [23], [24], and [25]. Finally, the third section of this paper is about the time consumption of the model. And then decisive remarks of this conducted research work are described in section V(D).

A. SETTINGS FOR EXPERIMENTS

In experiments, we implemented all the perspectives to reduce complexity in VVC/H.266 using VTM 7.0 reference software. The evaluation of our experimental techniques is based on the 900 images (test images) and 60 videos (test-images) sequences in an image database. The four QP 22, 27, 32, and 37 values are used on which AI configuration of video clips and images are encoded. We compare the time reduction rate of the encoding with the actual VTM software. We saved to calculate the complexity decrease. In [32] measure the RD performance with Bjontegaard-delta-PSNR (BDPSNR) and Bjontegaard-Delta-Bit-rate (BDBR). To perform all experiments, we use intel® CPU – e5-2680 V4 with 2.40GHz processing speed, 256GB ram, and Linux-Ubuntu- 16.04, 64bit-OS. For training purposes, we use

TABLE 2. MLE-CNN Variable Threshold Values

Mode	Threshold Values					
	2	3	4	5	6	Average
Fast	0.65	0.44	0.46	0.45	0.5	0.5
Medium	0.4	0.3	0.3	0.25	0.24	0.3

NVidia GeForce-RTX-3060-Ti GPU. But the testing of the performance of the encoding task was performed without GPU. Hence, we could compare the performance of encoding fairly.

1) MLE-CNN Setting

The CU partition of chrominance and luminance is calculated independently in the VVC/H.266 standard. So, model training for the various color channels individually for MLE-CNN. The overall 19 models of MLE-CNN we used for training for channels of the color size of CU. Figure 6 shows the order of various MLE-CNN and each model for its trainable elements. The total CU's with rectangular shapes put them to model with a height less than the width.

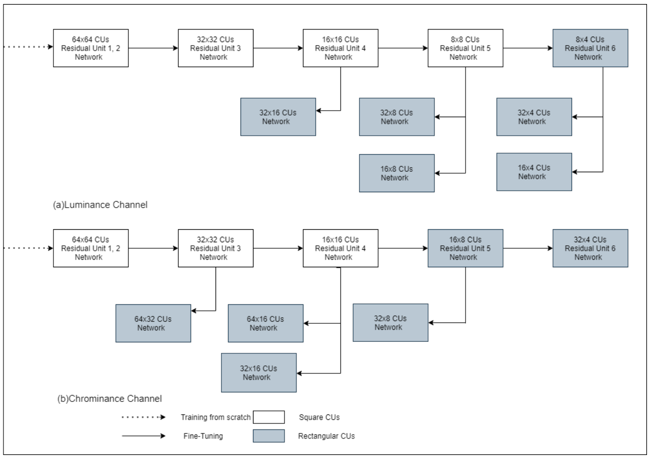

64x64 CUs Residual Unit 1, 2 Network
32x32 CUs Residual Unit 3 Network
16x16 CUs Residual Unit 4 Network
8x8 CUs Residual Unit 5 Network
8x4 CUs Residual Unit 6 Network
32x16 CUs Network
32x8 CUs Network
32x4 CUs Network
16x8 CUs Network
16x4 CUs Network
(a)Luminance Channel
64x64 CUs Residual Unit 1, 2 Network
32x32 CUs Residual Unit 3 Network
16x16 CUs Residual Unit 4 Network
16x8 CUs Residual Unit 5 Network
32x4 CUs Residual Unit 6 Network
64x32 CUs Network
64x16 CUs Network
32x8 CUs Network
32x16 CUs Network
(b)Chrominance Channel
Training from scratch
Square CUs
Fine-Tuning
Rectangular CUs

FIGURE 6. Training Process of Each Model in MLE-CNN

If we find a CU with a width less than the height, transpose that partition pattern, and content is needed. All the hyperparameters that are used to train the MLE-CNN were calibrated on the validation data set of our database [33]. In MLE-CNN precisely the loss function, we set 0.3 for and 1.0 for the . In starting, all the biases and weights are initialized randomly. For every model trained from zero, the batch size was 36 and iterated 500,000 times. The 10-3 was the learning rate and then reduced exponentially by 1% after 2200 iterations. The parameters were not changed but with Adam's algorithm [34] the trainable components parameters were fine-tuned. When MLE-CNN enters the phase of inference, the selection of variable threshold values as described in section 4.2 and table 2. The table shows the threshold values among the fast mode and medium mode. The average value during the fast mode was 0.5 and 0.3 during the medium mode.

B. EVALUATION OF PERFORMANCE

According to coding efficiency and complexity decrease we have the state-of-the-art models [23], [24] and [25] so we compare our MLE-CNN model performance with them. These models [23] and [24] use the QTMT based partition of CU for VVC/H.266 as we did. Moreover, we test the complexity minimization approach for HEVC/H.265 using deep learning, later used for the VVC/H.266 standard. The comparative results are shown in Tables 3 and Table 4. Our results are based on the 60 video sequences and 900 images respectively. The Table 3 demonstrates our less time-consuming method which decreases the 59.57%≈ 69.11% time of encoding for a video sequence, this time reduction is quite considerable. It was 55.65% ≈ 59% [23], [24] 52.48% ≈ 64.44% and [25] 38.19% ≈ 41.79%.

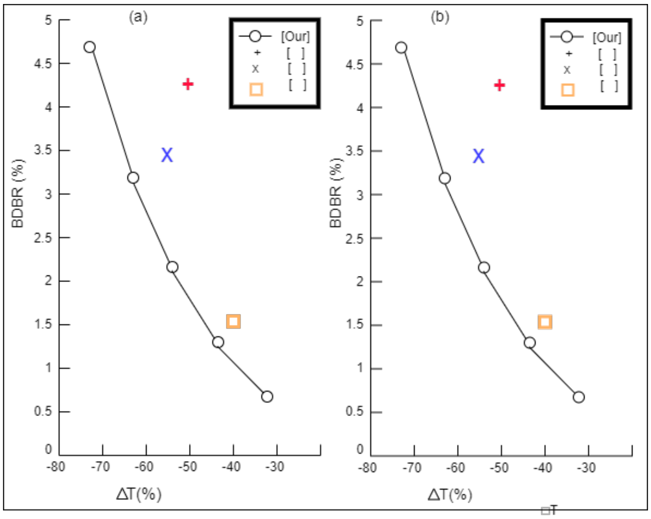


FIGURE 7. (a) Video Complexity RD Performance (b) Images Complexity RD Performance

In addition to the RD performance, our proposed method has the minimum BDBR and 1.023% of redundancy. The average loss in the BD-PSNR is 0.055dB. These above mentioned our achieved values are better than the [23], [24], and [25] advanced models. In terms of this applied matrix BD-PSNR, BDBR and our model performs fast comparatively [23]- [25]. Our MLE-CNN performs best in terms of RD performance and complexity on the video clips. It is possible due to the data-driven approach that MLE-CNN uses. So, it shows high accuracy, and secondly, it uses the direct prediction method. In addition, RDO search ignores the redundancies. Images have the same results that are illustrated in table 4.

In Figure 7 shows the most comprehensive analysis about the Complexity of RD performance of multiple methods using the four QP points. Section V(A) described the variable threshold points by using our method. Our proposed approach shows in the form of a curve at the left bottom side comparatively all other lines for images and video. It simply means our model consumes less encoding time than others at the equal BDBR value. In the same encoding time, our model performs better RD performance value. Consequently, our proposed approach is verified.

C. MLE-CNN TIME CONSUMPTION ANALYSIS

To increase the efficiency of encoding of VVC/H.266, our model must use less time for computation. Otherwise, it will create overhead. So, we deeply analyze the time consumption of MLE-CNN running time and compare it with VTM 7.0 encoder. The Figure 8 described the ratio of total time consumed in encoding and time used for the MLE-CNN. The average outcome for overall test videos and images with the equivalent 4 QP points. MLE-CNN produces 5% or even less overhead for most of the resolutions compared to real VTM. It is explained in the Figure 8—the similar explanation described for the videos and images overheads. For images, the average overhead time is 3.02% and 3.67% for the videos, respectively. This overhead is a small part of the overall encoding time. This less time achievement is because of redundancies removal from the CU partition. Note that this partition is QTMT based. Finally, the results for all encoding time are minimized to 64.53% average and 45.96% at Fast mode and medium mode. The verification of the performance is described in Section V(B).

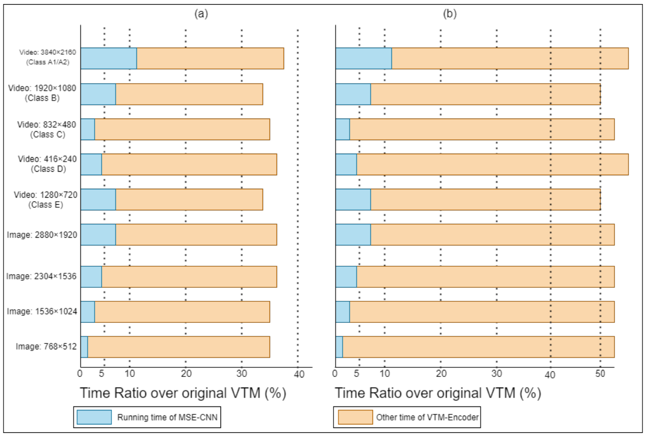


FIGURE 8. Run Time Performance of MLE-CNN and VTM Encoder

D. INVESTIGATION OF THE STUDY

This section investigates our proposed MLE-CNN to judge the working of its parts and their effectiveness. Table 5 shows the results in this regard. First of all, we tested single-level exit CNN comparatively multi-level CNN. In this case, we are not considering the RD cost in the loss function. Toward fast mode, In Section V(B), the value of $\beta = 0$ and the variable threshold value s are constant for the level. After that multi-level structure, RD's variable threshold and cost are included consecutively. In the table 5 categories 1, 2, 3, and 4 are given as multi-level, RD cost, variable threshold, and Fast mode, respectively.

TABLE 3. Performance of Complexity Rd on Video Sequence

Class	Sequence	Approach	BD-BR (%)	BD-PSNR (dB)	T(%)			
					QP=22	QP=27	QP=32	QP=37
A1	Campfire	[23]	4.328	-0.12	-62.4	-61.36	-62.07	-60.85
		[24]	2.876	-0.072	-51.8	-57.84	-35.33	-46.4
		[25]	1.655	-0.046	-35.7	-28.64	-40.36	-35.59
		MLE-CNN: “fast”	4.165	-0.116	-65.7	-68.21	-68.02	-64.12
		MLE-CNN: “medium”	2.015	-0.056	-43.7	-47.2	-52.1	-51.74
	FoodMarket4	[23]	2.349	-0.077	-61.7	-55.05	-52.29	-44.26
		[24]	3.454	-0.124	-60.4	-67.47	-41.65	-51.72
		[25]	2.254	-0.073	-36	-27.32	-33.5	-34.31
		MLE-CNN: “fast”	2.784	-0.091	-68.8	-61.97	-56.56	-44.31
		MLE-CNN: “medium”	1.256	-0.042	-49.2	-44.09	-42.08	-33.58
	Tango2	[23]	3.367	-0.047	-65.3	-59.38	-49.9	-35.29
		[24]	6.852	-0.136	-51.4	-72.07	-22.7	-29.24
		[25]	2.604	-0.046	-34.1	-34.69	-38.63	-36.28
		MLE-CNN: “fast”	3.485	-0.051	-70.8	-66.65	-53.02	-26.91
		MLE-CNN: “medium”	1.521	-0.024	-52.6	-50.06	-39.91	-20.53
A2	CatRobot1	[23]	6.748	-0.152	-61.8	-61.95	-59.75	-55.49
		[24]	5.266	-0.146	-53	-62.05	-40.3	-40.55
		[25]	1.484	-0.039	-32.3	-20.74	-36.13	-33.63
		MLE-CNN: “fast”	4.875	-0.112	-69.1	-64.9	-61.4	-56.11
		MLE-CNN: “medium”	2.163	-0.053	-49.4	-45.8	-43.39	-43.03
	DaylightRoad2	[23]	2.796	-0.064	-63	-61.56	-61.17	-57.58
		[24]	7.968	-0.149	-59.8	-71.35	-56.03	-64.67
		[25]	1.212	-0.041	-40.9	-30.84	-23.12	-27.92
		MLE-CNN: “fast”	2.781	-0.067	-72.5	-68.06	-63.65	-58.25
		MLE-CNN: “medium”	1.163	-0.03	-56.5	-49.62	-45.93	-45.35
	ParkRunning3	[23]	2.687	-0.133	-64.9	-62.54	-62.4	-61.8
		[24]	3.167	-0.14	-50.2	-57.2	-36.12	-52.14
		[25]	0.918	-0.045	-23.2	-24.65	-33.86	-34.51
		MLE-CNN: “fast”	2.675	-0.132	-60.5	-60.47	-62.09	-68.97
		MLE-CNN: “medium”	1.146	-0.056	-35.6	-36.27	-41.64	-53.2
B	MarketPlace	[23]	2.004	-0.076	-61.6	-60.73	-59.32	-58.32
		[24]	3.286	-0.122	-51.7	-68.29	-43.56	-63.12
		[25]	1.166	-0.045	-23.1	-35.44	-34.33	-40.12
		MLE-CNN: “fast”	1.891	-0.072	-64.8	-64.42	-65.38	-65.97
		MLE-CNN: “medium”	0.803	-0.031	-41.6	-43.42	-47.78	-53.72
	RitualDance	[23]	3.859	-0.183	-58.7	-58.32	-57.22	-54.4
		[24]	4.053	-0.191	-60.4	-70.14	-49.14	-58.13
		[25]	1.55	-0.075	-16.5	-24.19	-41.36	-38.64
		MLE-CNN: “fast”	2.693	-0.129	-67.2	-64.29	-61.78	-58.64
		MLE-CNN: “medium”	1.071	-0.052	-46.4	-44.97	-43.61	-44.51
	BasketballDrive	[23]	3.553	-0.091	-59.7	-61.18	-60.12	-54.79
		[24]	5.923	-0.148	-56.3	-70.28	-55.24	-59.31
		[25]	1.595	-0.042	-37.8	-39.95	-41.65	-36.07
		MLE-CNN: “fast”	3.873	-0.101	-71.4	-72.16	-67.86	-62.6
		MLE-CNN: “medium”	1.642	-0.044	-53	-55.62	-51.52	-48.03
	BQTerrace	[23]	1.75	-0.074	-47.6	-58.27	-57.78	-58.01
		[24]	4.378	-0.177	-48.9	-70.15	-67.68	-63.55
		[25]	1.459	-0.066	-22.3	-48.56	-49.64	-50.2
		MLE-CNN: “fast”	2.574	-0.123	-54.2	-69.53	-67.74	-66.09
		MLE-CNN: “medium”	1.111	-0.054	-29.4	-51.23	-50.41	-51.45
	Cactus	[23]	3.541	-0.112	-60.1	-60.11	-58.78	-59.22
		[24]	4.555	-0.143	-57.8	-67.39	-61.31	-62.65
		[25]	1.304	-0.042	-41.4	-41.55	-43.73	-40.13
		MLE-CNN: “fast”	2.846	-0.091	-69.9	-68.32	-66.23	-66.4
		MLE-CNN: “medium”	1.124	-0.036	-49.4	-49.07	-47.6	-51.14
C	BasketballDrill	[23]	4.285	-0.194	-56.7	-60.37	-59.61	-56.83
		[24]	4.596	-0.208	-58.2	-62.68	-63.21	-53.45
		[25]	1.744	-0.08	-41.9	-45.66	-44.42	-41.05
		MLE-CNN: “fast”	4.722	-0.212	-63.2	-61.55	-60.85	-58.35
		MLE-CNN: “medium”	1.625	-0.074	-40.1	-37.83	-39.26	-39.93
	BQMall	[23]	4.193	-0.213	-59.3	-58.68	-59.18	-58.51
		[24]	4.975	-0.251	-62	-67.25	-64.64	-59.23
		[25]	1.143	-0.058	-44	-43.91	-43.26	-40.1
		MLE-CNN: “fast”	3.102	-0.158	-70.7	-68.22	-65.89	-65.01
		MLE-CNN: “medium”	1.17	-0.06	-52.9	-50.51	-46.87	-48.78
	PartyScene	[23]	1.939	-0.13	-56.1	-57.26	-58.02	-59.28
		[24]	2.238	-0.152	63.6	-66.9	-67.57	-52.39
		[25]	0.779	-0.052	-44.9	-45.95	-47.78	-48
		MLE-CNN: “fast”	1.857	-0.124	-65.7	-66.15	-64.2	-62.5
		MLE-CNN: “medium”	0.612	-0.041	-47.3	-47.52	-43.7	-42.27
	RaceHorses	[23]	3.181	-0.171	-60.7	-58.76	-58.39	-57.71
		[24]	2.746	-0.146	-57.2	-60.9	-54.99	-51.4
		[25]	0.997	-0.054	-44.8	-41.9	-45.77	-44.66
		MLE-CNN: “fast”	2.503	-0.135	-66.9	-65.12	-63.4	-67.31
		MLE-CNN: “medium”	0.963	-0.052	-47.4	-44.07	-42.93	-51.42
D	BasketballPass	[23]	3.38	-0.198	-57.8	-58.51	-58.02	-56.73
		[24]	2.914	-0.169	-51.8	-55.8	-50.54	-44.78
		[25]	0.898	-0.053	-41.3	-41.03	-41.44	-39.32
		MLE-CNN: “fast”	3.664	-0.214	-66.2	-64.96	-62.34	-56.97
		MLE-CNN: “medium”	1.405	-0.082	-47	-46.8	-44.78	-39.21
	BlowingBubbles	[23]	2.284	-0.146	-54.6	-54.71	-54.47	-57.34
		[24]	1.804	-0.119	-59.7	-59.55	-60.04	-44.02
		[25]	0.611	-0.039	-42.2	-42.64	-43.62	-47.47
		MLE-CNN: “fast”	2.383	-0.151	-62.5	-64.1	-61	-59.8
		MLE-CNN: “medium”	0.922	-0.06	-42	-43.64	-39.63	-40.93
	BQSquare	[23]	1.134	-0.083	-54.6	-54.41	-54.1	-53.54
		[24]	1.988	-0.147	-56	-60.21	-62.96	-42.92
		[25]	1.399	-0.103	-53.5	-56.68	-59.81	-63.04
		MLE-CNN: “fast”	2.035	-0.149	-62.3	-61.45	-64	-62.36
		MLE-CNN: “medium”	0.743	-0.054	-42.7	-42.91	-47.02	-45.35
	RaceHorses	[23]	3.416	-0.202	-56.6	-56.23	-55.8	-57.83
		[24]	2.19	-0.13	-53.5	-54.17	-50.9	-43.34
		[25]	0.861	-0.051	-41.6	-42.19	-43.71	-43.11
		MLE-CNN: “fast”	2.917	-0.171	-63.1	-60.96	-60.62	-60.51
		MLE-CNN: “medium”	1.2	-0.071	-41.6	-40.72	-40.76	-43.49
E	FourPeople	[23]	3.765	-0.197	-58.6	-56.86	-57.52	-58.52
		[24]	5.937	-0.307	-66.7	-71.65	-68.51	-62.48
		[25]	1.336	-0.07	-44.9	-44.41	-40.76	-36.78
		MLE-CNN: “fast”	3.295	-0.173	-71.6	-68.1	-64.01	-63.88
		MLE-CNN: “medium”	1.334	-0.07	-55.3	-50.24	-45.8	-48.12
	Johnny	[23]	6.479	-0.24	-60.8	-58.49	-57.56	-54.35
		[24]	6.603	-0.246	-61.7	-62.15	-62.48	-57.19
		[25]	2.643	-0.098	-50	-43.49	-46.12	-43.77
		MLE-CNN: “fast”	5.084	-0.188	-71.1	-67.32	-62.69	-56.29
		MLE-CNN: “medium”	2.327	-0.087	-54.5	-50.2	-47.19	-40.72
	KristenAndSara	[23]	4.707	-0.215	-58.5	-56.6	-55.88	-53.71
		[24]	5.413	-0.247	-62.5	-62.12	-61.74	-51.83
		[25]	2.233	-0.102	-47.9	-43.79	-46.29	-46.29
		MLE-CNN: “fast”	3.925	-0.181	-73.1	-67.78	-64.36	-59.17
		MLE-CNN: “medium”	1.761	-0.082	-56.5	-51.74	-47.9	-45.73
Average		[23]	3.443	-0.142	-59.1	-58.7	-57.7	-55.65
		[24]	4.236	-0.167	-57	-64.44	-53.48	-52.48
		[25]	1.448	-0.06	-38.2	-38.56	-41.79	-40.95
		MLE-CNN: “fast”	3.188	-0.134	-66.9	-65.67	-63.05	-59.57
		MLE-CNN: “medium”	1.322	-0.055	-47	-46.52	-45.08	-44.65

TABLE 4. Performance of Complexity Rd on Images

Source	Resolution	Approach	BD-BR (%)	T (%)			
				QP=22	QP=27	QP=32	QP=37
DataBase	2880×1920	[23]	2.866	-64.16	-63.17	-61.11	-58.62
		[24]	3.406	-54.96	-66.01	-48.91	-60.52
		[25]	1.27	-37.34	-35.02	-39.58	-41.47
		MLE-CNN: “fast”	2.359	-65.42	-64.35	-64.97	-64.08
		MLE-CNN: “medium”	1.036	-43.83	-42.39	-47.41	-50.06
	2304×1536	[23]	2.787	-61.45	-60.71	-58.72	-59.05
		[24]	3.675	-59.26	-68.5	-54.91	-63.5
		[25]	1.161	-38.34	-37.41	-39.76	-40.55
		MLE-CNN: “fast”	2.369	-64.03	-63.67	-64.49	-66.02
		MLE-CNN: “medium”	1.022	-42.69	-42.15	-46.38	-51.62
	1536×1024	[23]	2.35	-61.62	-60.67	-60.34	-59.4
		[24]	2.902	-57.85	-64.71	-59.44	-59.55
		[25]	0.848	-37.28	-39.56	-41.41	-39.62
		MLE-CNN: “fast”	2.179	-65.39	-63.54	-65.65	-67.85
		MLE-CNN: “medium”	0.864	-42.93	-42.02	-45.66	-52.77
	768×512	[23]	2.115	-60.89	-60.45	-60.16	-60.33
		[24]	2.325	-56.64	-63.67	-61.58	-54.24
		[25]	0.857	-38.93	-42.36	-45.24	-44.94
		MLE-CNN: “fast”	2.234	-66.17	-64.62	-65.3	-67.46
		MLE-CNN: “medium”	0.838	-45.02	-43.17	-44.73	-51.33
Average		[23]	2.529	-62.03	-61.25	-60.08	-59.35
		[24]	3.077	-57.18	-65.72	-56.21	-59.45
		[25]	1.034	-37.97	-38.59	-41.5	-41.65
		MLE-CNN: “fast”	2.285	-65.25	-64.04	-65.1	-66.35
		MLE-CNN: “medium”	0.94	-43.62	-42.43	-46.04	-51.45

1) Single Level/Multilevel CNN Architecture

In the single-level CNN, all the networks that get the input of level 2 feature maps (conditional convolution) are different from our proposed MLE-CNN, where we use feature maps from multi-levels. Specifically, the first layer of every residual unit of single-level exit CNN is extended 48 from 16. The MLE-CNN and the SLE-CNN both have equal numbers of trainable parameters. Therefore, we have compared the SLE-CNN with the MLE-CNN as categories 1 and 2. The table 5 shows clearly the good coding efficiency of MLE-CNN as compared to the SLE-CNN. So, The MLE-CNN is significantly better with the 0.150 dB of BD-PSNR escalation, and BDBR is 2.968% saved.

2) RD Cost With and Without LOSS Function

In the loss function, we find the RD cost, which shows the split modes and their good organization during the training time of MLE-CNN. We compare the loss functions one time with and the second time without the RD cost. Table 5 shows categories 2 and 3 for this. The 0.243% BDBR minimized redundancy in RD cost, and BD PSNR is 0.008dB improved. Moreover, at four points of QP, the encoding time is saved to 2.82% from 0.85%.

3) Variable Threshold Even/Uneven to Level

As far as MLE-CNN implementation is concerned, the variable threshold values are different at the multiple levels of partition of CU. The accuracy of the prediction at different levels is adjustable. For the comparison of the MLE-CNN variable, threshold values with uniform and no uniform values are used in the table 5 as 3 and 4. In fair comparison for both settings (4, 5), 0.5 is the average value for the threshold for all levels. In table 5, category four is exceeded by category three. The BDBR value is 0.140% time saving, and PSNR increases to 0.007dB with encoding time. In summary, the 1 to 4 category, minimization of complexity, and RD performance complexity are continuously upgraded. It means that flexible variable threshold and Loss function's RD cost are better for our proposed model MLE-CNN.

VII. CONCLUSION

We have suggested a deep learning focusing approach in this paper to predict the partition of CU, which is QTMT dependent and used to make fast and accurate intra mode of VVC/H.266 encoding. The VVC/H.266 partition of CU is quite flexible as compared to the HEVC/H.265. We developed an extensive size database for the various CU partition patterns and then examined the present split mode of CU at multiple CU's levels. Next, we proposed a deep learning model, MLE-CNN, which uses the small networks and conditional convolution with the power of the used network. We, then, the MLE-CNN model has established that contains the early exit mechanism that can ignore the marking process of non-functioned CU. In addition, the variable threshold decision system was established; it has got the best value in the middle of the complexity of encoding and output of RD.

After the experiments were conducted, we found the results of our proposed approach from 47.91% to 69.11% on average save the encoding time with the insignificant BDBR boost on video from 1.023% to 2.919%, surpassing the existing modern research works. Using the deep learning approach, the encoding time of the inter mode for VVC/H.266 can also save more time than now in the near future. There is a lot of potential for the deep neural networks and convolution neural networks to optimize the other parts of VVC/H.266, for example, the inter mode and intra angular selection to optimize the CU partitions. Furthermore, different advanced and futuristic techniques and their implementation in the field of network or field-programmable gate array can accelerate the techniques of CU partition specifically. In the coming years, this can be seen as a promising future job for facilitating quick VVC/H.266 encoders.

TABLE 5. Important Results

2*Study	2*Multi-Level	2*RD Cost	2*Variable Threshold	2*BD-BR(%)	2*BD-PSNR(dB)	T (%)			
						QP=22	QP=27	QP=32	QP=37
1				6.539	-0.299	-59.11	-61.56	-62.50	-59.34
2	X			3.571	-0.149	-65.48	-63.01	-60.66	-55.47
3	X	X		3.328	-0.141	-66.33	-64.56	-62.48	-58.29
4 (fast mode)	X	X	X	2.919	-0.134	-69.11	-65.67	-63.05	-59.57

REFERENCES

[1] Bross B, Andersson K, Bläser M, Drugeon V, Kim SH, Lainema J, Li J, Liu S, Ohm JR, Sullivan GJ, Yu R. General video coding technology in responses to the joint call for proposals on video compression with capability beyond HEVC. IEEE Transactions on Circuits and Systems for Video Technology. 2019 Oct 25;30(5):1226-40.

[2] Chen J, Karczewicz M, Huang YW, Choi K, Ohm JR, Sullivan GJ. The joint exploration model (JEM) for video compression with capability beyond HEVC. IEEE Transactions on Circuits and Systems for Video Technology. 2019 Oct 7;30(5):1208-25.

[3] Bossen F, Li X, Sühring K. Test model software development. InDocument JVET-Q0003 2020 Jan.

[4] Wieckowski A, Hinz T, George V, Brandenburg J, Ma J, Bross B, Schwarz H, Marpe D, Wiegand T. NextSoftware: an alternative implementation of the joint exploration model (JEM). InJVET-H0084, Joint Video Exploration Team (JVET) 2017 Oct.

[5] Chen J, Karczewicz M, Huang YW, Choi K, Ohm JR, Sullivan GJ. The joint exploration model (JEM) for video compression with capability beyond HEVC. IEEE Transactions on Circuits and Systems for Video Technology. 2019 Oct 7;30(5):1208-25.

[6] Ye Y, Boyce JM, Hanhart P. Omnidirectional 360° video coding technology in responses to the joint call for proposals on video compression with capability beyond HEVC. IEEE Transactions on Circuits and Systems for Video Technology. 2019 Nov 15;30(5):1241-52.

[7] Liu Z, Yu X, Gao Y, Chen S, Ji X, Wang D. CU partition mode decision for HEVC hardwired intra encoder using convolution neural network. IEEE Transactions on Image Processing. 2016 Aug 18;25(11):5088-103.

[8] Zhang Y, Wang G, Tian R, Xu M, Kuo CJ. Texture-classification accelerated CNN scheme for fast intra CU partition in HEVC. In2019 Data Compression Conference (DCC) 2019 Mar 26 (pp. 241-249). IEEE.

[9] Shi J, Gao C, Chen Z. Asymmetric-kernel CNN based fast CTU partition for HEVC intra coding. In2019 IEEE International Symposium on Circuits and Systems (ISCAS) 2019 May 26 (pp. 1-5). IEEE.

[10] Lin TL, Jiang HY, Huang JY, Chang PC. Fast binary tree partition decision in H. 266/FVC intra Coding. In2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW) 2018 May 19 (pp. 1-2). IEEE.

[11] Fu T, Zhang H, Mu F, Chen H. Fast CU partitioning algorithm for H. 266/VVC intra-frame coding. In2019 IEEE International Conference on Multimedia and Expo (ICME) 2019 Jul 8 (pp. 55-60). IEEE.

[12] Park SH, Kang JW. Context-based ternary tree decision method in versatile video coding for fast intra coding. IEEE Access. 2019 Nov 28;7:172597-605.

[13] Yang H, Shen L, Dong X, Ding Q, An P, Jiang G. Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding. IEEE Transactions on Circuits and Systems for Video Technology. 2019 Mar 11;30(6):1668-82.

[14] Mukherjee D, Bankoski J, Grange A, Han J, Koleszar J, Wilkins P, Xu Y, Bultje R. The latest open-source video codec VP9-an overview and preliminary results. In2013 Picture Coding Symposium (PCS) 2013 Dec 8 (pp. 390-393). IEEE.

[15] Chen Y, Murherjee D, Han J, Grange A, Xu Y, Liu Z, Parker S, Chen C, Su H, Joshi U, Chiang CH. An overview of core coding tools in the AV1 video codec. In2018 Picture Coding Symposium (PCS) 2018 Jun 24 (pp. 41-45). IEEE.

[16] He Z, Yu L, Zheng X, Ma S, He Y. Framework of AVS2-video coding. In2013 IEEE International Conference on Image Processing 2013 Sep 15 (pp. 1515-1519). IEEE.

[17] Shen L, Zhang Z, Liu Z. Effective CU size decision for HEVC intracoding. IEEE Transactions on Image Processing. 2014 Jul 23;23(10):4232-41.

[18] Jia M, Gao Y, Li S, Yue J, Ye M. An explicit self-attention-based multimodality CNN in-loop filter for versatile video coding. Multimedia Tools and Applications. 2021 Jul 24:1-5.

[19] Jin Z, An P, Yang C, Shen L. Fast QTBT partition algorithm for intra frame coding through convolutional neural network. IEEE Access. 2018 Sep 28;6:54660-73.

[20] Zhao J, Wang Y, Zhang Q. Adaptive CU Split Decision Based on Deep Learning and Multifeature Fusion for H. 266/VVC. Scientific Programming. 2020 Aug 1;2020.

[21] Zouidi N, Belghith F, Kessentini A, Masmoudi N. Complexity reduction of versatile video coding standard: a deep learning approach. Journal of Electronic Imaging. 2021 Mar;30(2):023002.

[22] Zhang Q, Wang Y, Huang L, Jiang B. Fast CU partition and intra mode decision method for H. 266/VVC. IEEE Access. 2020 Jun 24;8:117539-50.

[23] Fu T, Zhang H, Mu F, Chen H. Fast CU partitioning algorithm for H. 266/VVC intra-frame coding. In2019 IEEE International Conference on Multimedia and Expo (ICME) 2019 Jul 8 (pp. 55-60). IEEE.

[24] Yang H, Shen L, Dong X, Ding Q, An P, Jiang G. Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding. IEEE Transactions on Circuits and Systems for Video Technology. 2019 Mar 11;30(6):1668-82.

[25] Abdallah B, Belghith F, Ayed MA, Masmoudi N. Low-complexity QTMT partition based on deep neural network for Versatile Video Coding. Signal, Image and Video Processing. 2021 Jan 19:1-8

[26] Xu M, Deng X, Li S, Wang Z. Region-of-interest based conversational HEVC coding with hierarchical perception model of face. IEEE Journal of Selected Topics in Signal Processing. 2014 Apr 2;8(3):475-89.

[27] Bouaafia S, Khemiri R, Messaoud S, Ben Ahmed O, Sayadi FE. Deep learning-based video quality enhancement for the new versatile video coding. Neural Computing and Applications. 2021 Sep 8:1-5.

[28] Domański M, Stankiewicz O, Wegner K, Grajek T. Immersive visual media—MPEG-I: 360 video, virtual navigation and beyond. In2017 International Conference on Systems, Signals and Image Processing (IWSSIP) 2017 May 22 (pp. 1-9). IEEE.

[29] Boyce J, Alshina E, Abbas A, Ye Y. JVET-H1030: JVET common test conditions and evaluation procedures for 360○ video. Joint Video Explor. Team, Macau, China, Rep. JVET-H1030. 2017 Oct.

[30] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. InProceedings of the thirteenth international conference on artificial intelligence and statistics 2010 Mar 31 (pp. 249-256). JMLR Workshop and Conference Proceedings.

[31] Pettee M. Interdisciplinary Machine Learning Methods for Particle Physics (Doctoral dissertation, Yale University).

[32] Wang Z, Wang S, Zhang X, Wang S, Ma S. Fast QTBT partitioning decision for interframe coding with convolution neural network. In2018 25th IEEE International Conference on Image Processing (ICIP) 2018 Oct 7 (pp. 2550-2554). IEEE.

[33] Fu T, Zhang H, Mu F, Chen H. Fast CU partitioning algorithm for H. 266/VVC intra-frame coding. In2019 IEEE International Conference on Multimedia and Expo (ICME) 2019 Jul 8 (pp. 55-60). IEEE.

[34] Yang H, Shen L, Dong X, Ding Q, An P, Jiang G. Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding. IEEE Transactions on Circuits and Systems for Video Technology. 2019 Mar 11;30(6):1668-82.

SAMEENA JAVAID has received her M.S. degree in Computer Science from Bahria University Karachi Campus (BUKC), Pakistan, in 2016. Currently she is working as Assistant Professor at Department of Computer Sciences, School of Engineering and Applied Sciences, BUKC. She is also pursuing her Ph.D. degree in Computer Science from BUKC, Karachi, Pakistan. Her current research interests include computer vision, machine learning, deep learning, and ubiquitous computing.

SYED SAFDAR ALI RIZVI has received his M.S. degree in Computer Science from Muhammad Ali Jinnah University (MAJU), Karachi, Pakistan, in 2008. He has earned Ph.D. degree from Electrical and Electronic Engineering Department, Universiti Teknologi PETRONAS, Malaysia, in 2014. Currently, he is working as Associate Professor and Head of the Department at Department of Computer Sciences, School of Engineering and Applied Sciences, Bahria University Karachi Campus (BUKC), Karachi, Pakistan. His research interest includes Wireless communication, 5G technologies, Wireless Heterogeneous Networks along with Artificial Intelligence and Machine Learning interventions in Communication technologies.

MUHAMMAD TALHA UBAID has recently completed his M.S. degree in Computer Science with specialization in Computer Vision domain from the University of Central Punjab (UCP), Lahore, Pakistan. Currently, he is working as a Research Officer in the Intelligent Criminology Research Laboratory, National Center of Artificial Intelligence. His areas of interest are Computer Vision, Machine Learning and Deep Learning.

ABDULLAH TARIQ has recently completed his B.S. degree in Computer Science with specialization in the Computer Vision domain from the University of Engineering and Technology (UET), Lahore, Pakistan. Currently, he is working as a Research Officer in the Intelligent Criminology Research Laboratory, National Center of Artificial Intelligence. His areas of interest are Computer Vision, Machine Learning and Deep Learning.