VNU Journal of Science:
Computer Science and Communication Engineering
Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

Early CTU Termination and Three-steps Mode Decision Method for Fast Versatile Video Coding

Sang Nguyen Quang[1], Tien Vu Huu[2], Duong Dinh Trieu[1], Minh Dinh Bao[1], Minh Do Ngoc[1], Xiem Hoang Van[1,*]

[1]*VNU University of Engineering and Technology, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*

[2]*Posts and Telecommunications Institute of Technology, 122 Hoang Quoc Viet, Cau Giay, Hanoi, Vietnam*

Received 11 May 2022
Revised 22 October 2022; Accepted 27 November 2022

Abstract: Versatile Video Coding (VVC) has been recently becoming popular in coding video data due to its compression efficiency. To reach this performance, Joint Video Experts Team (JVET) has introduced a number of coding improvement techniques to VVC model. Among them, VVC Intra coding proposed a new concept of quad-tree nested multi-type tree (QTMT) and extended the predicted modes with up to 67 options. As a result, the complexity of the VVC Intra encoding also greatly increases. To make VVC Intra coding more feasible in real¬time applications, we propose in this paper a novel fast mode decision method together with a deep learning based fast QTMT. At the first stage, we use a learned convolutional neural network (CNN) to predict the coding unit map and then fed into the VVC encoder to early terminate the block partitioning process. After that, we design a statistical model to predict a list of most probable modes (MPM) for each selected Coding Unit (CU) size. Finally, we introduce a novel three-steps mode decision algorithm to estimate the optimal directional mode without sacrificing the compression performance. The proposed early CU splitting and fast intra prediction are integrated into the latest VTM reference software. Experimental results show that the proposed method can save 50.2% encoding time with a negligible BD-Rate increase.

Keywords: VVC Intra coding, Early-Terminate Hierarchical, CNN, Most probable mode (MPM).

* Corresponding author.
E-mail address: xiemhoang@vnu.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.375

1. Introduction

Nowadays, digital videos have been playing an important role in storing and transmitting information. To meet the demand for more realism and sharpness, a number of video formats and resolutions have been produced such as high dynamic range (HDR), ultra-high definition (UHD), 360-degree, and high frame rate (HFR). However, the enhancement of video resolutions and the development of video formats also led to a great increase in data volume. For this reason, the popular H.265/HEVC (High Efficiency Video Coding) standard [1] may not be efficient. To overcome this problem, Joint Video Exploration Team (JVET) has recently launched a new video coding standard named Versatile Video Coding (VVC) [2]. As reported in [3], VVC standard not only supports various video formats, ranging from HD to UHD and HDR but also achieves around 50% bitrate saving while providing a similar perceptual quality when compared with the prior HEVC standard.

To achieve such compression efficiency, VVC exploits a number of coding tools. Among them, QTMT (quadtree and multi-type tree) based CTU (coding tree unit) partitional structure, which supports more flexible CU partition shapes compared to the QTBT (quadtree and binary tree) partition structure in HEVC has been introduced. With QTMT, the encoder will evaluate every possible partition structure, and choose the best ones having minimum RDCost, computed in equation (1).

$$RDCost = D + \lambda \times R \quad (1)$$

Where D is the difference between the original and reconstructed information. R refers to the bitrate which is needed to encode the CU and λ is a Lagrange multiplier. With flexible CU partition shapes as demonstrated in Figure 1, VVC achieved great coding performance but also asked for much higher computational complexity.

To achieve high video compression efficiency, a number of methods have been proposed. In [4], targeting on scalability of VVC, authors proposed a novel coding method for improving VVC compression efficiency by creating a new reference frame using frame rate up conversion (FRUC) [5] and put it on the reference list of enhancement layer. In [6], a novel post-processing method for improving VVC reconstructed videos using deep learning method has been introduced.

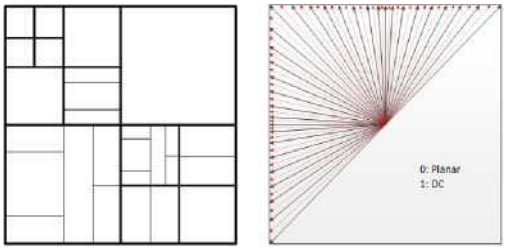

Figure 1. An example of QTMT (a) and Intra directional mode (b) in VVC.

Another important update to make VVC more efficient is the increase of intra-mode directions, which also led to the increment of computation time. If the depth of coding unit (CU) and the direction of intra-mode can be predicted, the computational complexity can be reduced with negligible quality degradation.

Targeting the VVC complexity reduction, Zhang et al. proposed in [7] a texture based fast CU partitioning method. This research includes two algorithms: i) Early terminate the splitting process based on texture energy; and ii) Using texture direction to decide the splitting mode of CUs. In [8], Jin et al., introduced a novel fast QTBT partition decision method based on Convolutional Neural Network (CNN). First, the current frame is divided into a set of 32x32 blocks and CNN model will predict the minimum and maximum partition depth for QTBT partition. In the depth 0, if the classification result is ‘‘0’’ for any patch within CTU, it means that the CTU is smooth and no longer split. Otherwise, CTU is divided into smaller blocks. In step 2, the current depth is 1, if all patches of the current CU are smooth and predicted with label ‘‘0’’ or ‘‘1’’, the processor

calculates RD-Cost of the 64x64 and then calculates RD-Cost of each sub-CUs in the next step. If the classification results of CNN are bigger than "1", it means that the current 64x64 CU has some detailed textures, therefore processor directly divides it into four sub-CUs using quad-tree partition without calculating RD cost. In step 3, the current size of block is 32x32, the encoder will calculate RD cost for each partition depth within the candidate depth range, and finally determine the optimal QTBT partition structure through RDO process. In [9], Yang et al. proposed a statistical learning based low complexity CTU structure decision method. In this method, a decision tree is used to early terminate the block partitioning process without using RDO. There are three features that are calculated to make the decision: Global texture information, Local texture information, and Context information.

To reduce the complexity of intra mode prediction decision process, several fast mode decision algorithms have been proposed. In [9], besides proposing an early block partitioning method, the authors also provided a fast intra mode decision algorithm based on One-Dimensional Gradient Descent Search. This algorithm contains three steps: Initial search mode determination, Bidirectional search patterns and Search step size decision. In [10], Chen et al. analyzed the relationship between RMD-Cost and RD-Cost. In addition, the authors presented a statistic on the ratio of optimal mode belongs to the most probable mode (MPM) list. Following these observations, they proposed a two-strategies algorithm, including: i) Generating a candidate list based on RMD-Cost; and ii) Sorting the candidate list in ascending order of RMD-Cost and the RDO process can be early terminated by using a threshold.

Artificial intelligence, notably the deep learning technique has recently been used in a wide range of vision and image processing applications [11]. Following this direction, Mai Xu et al., [12] introduced a deep learning-based method to reduce the complexity of HEVC encoder. Afterwards, the authors in [13] have customized and integrated this model into the VVC and achieved important encoding time saving. Recently, Abdallah et al., in [14] adopted this neural network in VVC by using multiple thresholds to decide how the square CUs will be partitioned.

Motivated from early works in [13], we introduce in this paper a fast VVC compression mechanism including a CNN based CTU early termination and a novel fast intra mode decision method in which the most selected mode is statistically analyzed and considered to propose a three-step mode selection method [15].

To demonstrate the efficiency of the proposed fast intra coding solution, the rest of this paper is organized as follows. Section 2 briefly introduces block partitioning and the optimal intra mode decision process in VVC. Section 3 presents the proposed low complexity algorithms for intra coding in VVC while Section 4 discusses the experimental results. Finally, Section 5 gives the conclusion of this paper.

2. Background Works

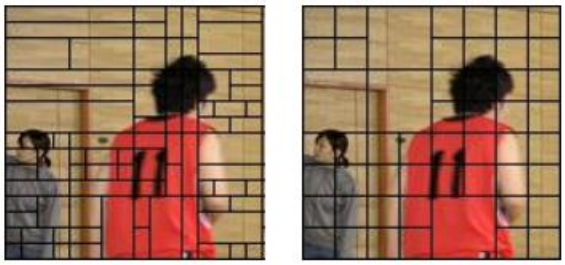

Figure 2. Block partitioning in H.266/VVC (left) and H.265/HEVC (right).

Figure 2 shows an example of block partitioning with VVC (left) and HEVC (right). It can be seen that VVC supports both square and rectangle partition shapes while blocks in HEVC are only square. In HEVC [1], input frames are divided into the basic coding structure called coding tree unit - CTU, with the maximum, allowed Luma block size being 64x64. After that, CTU with size of 2Nx2N is recursively

partitioned into smaller coding units - CU via the quad-tree. By using quad-tree structure, CTU and CU are split into four sub-CU with size of NxN [16].

In VVC, the raw image is also divided into a sequence of CTUs with the same concept in HEVC [17, 18]. However, VVC allows the maximum size of the Luma block in a CTU to be 128x128. Firstly, in depth = 0, if CTU size is larger than the maximum allowed binary and ternary tree size, only quad-tree structure is applied to partition CTU into four smaller CU with size of 64x64. After that, while depth is greater or equal to 1, Quad-tree with nested multi-type tree (QTMT) containing binary and ternary splits is used to partition the current block into sub blocks. Figure 3 shows four splitting types of multi-type tree structure. A CU can be split into sub-CUs by vertical binary splitting (SPLIT_BT_VER), horizontal binary splitting (SPLIT_BT_HOR), vertical ternary splitting (SPLIT_TT_VER), and horizontal ternary splitting (SPLIT_TT_HOR) mode. In most cases, if the width or height of the color component of the CU is smaller than the maximum supported transform length, CU, PU and TU have the same block shape and size in the QTMT coding block structure. In case the width or height of the coding block is larger than the maximum transform width or height, the coding block is automatically split in vertical and/or horizontal direction to meet the supported transform size.

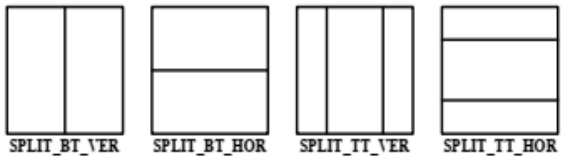

Figure 3. Binary and Ternary Split in H.266/VVC.

If a part of the current CU exceeds the bottom or right picture boundary, it is forced to be split until all samples of every coded CU are not located outside the picture boundaries. For example, when a portion of the current CU exceeds both the bottom and the right picture boundaries, it is forced to be split with QT split mode if the sub-CU size is larger than the minimum QT size, otherwise, the block is forced to be split with SPLIT_BT_HOR mode.

In VVC, with the increase in block size and shapes, the number of intra prediction modes in VVC has been extended to 67 (containing DC, Planar and 65 directional modes), compared with 35 modes in the previous video coding standard. Figure 1b shows these 67 intra prediction modes with the black arrows representing the modes which have been introduced in HEVC and the red arrows representing the new directional modes in VVC.

To reduce the complexity of the encoder, the optimal intra prediction mode has been selected by performing a three-stage process. According to [19], in the first stage, called Rough Mode Decision - RMD, the encoder uses the Hadamard transform and calculates the RMD- Cost based on the Sum of Absolute Transform Difference (SATD) of 35 original prediction modes (indicated by black solid arrows in Figure 1b) by using equation (2).

$$SATD = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|TH * \left(y_{ORG}(i,j) - y_{PRED}(i,j)\right) * TH^{T}\right| \quad (2)$$

where m, n represents the width and height of current CU, y_{ORG} is the original pixel value, y_{PRED} is the predicted pixel value, TH and TH^T are the Hadamard transform matrix and its transpose.

After calculating SATD, a list of N modes having the lowest cost is generated. In the next step of stage 1, the same Hadamard Transform and SATD measure are adopted to evaluate the two neighbors of the N candidate modes (the red arrows in Figure 1b). In the final step of this stage, the list of N candidate modes is updated.

In the second stage, a 6-elements list, called Most Probable Mode (MPM), is added to the candidate list. The MPM list is generated based on the optimal mode of the left and above blocks. In some special cases, the encoder puts the default modes into MPM list.

Finally, the RDO process is applied for all modes in the candidate list. The optimal mode of the current block is selected in this stage.

To reduce the complexity of VVC encoding time, in this paper, an early CU termination method based on deep learning and a fast intra mode selection methods are proposed. Details of both methods are presented in section 3.

3. Proposed Methods

3.1. Statistical Analysis and Observation

Typically, video contents can be divided into three main categories including Screen Content, Natural and Conference. Therefore, to reveal how video content affects to the CTU selection, three video sequences, *BasketballDrillText*, *BQTerrace* and *vidyo1*, respectively, representing three categories above are used.

Figure 4 illustrates the first frame of each video sequence. These videos are encoded with four common quantization parameter (QP) values (22, 27, 32, 37), then the statistics of CU size and intra mode decision are analyzed. Table 1 describes in detail experimental conditions for statistical work and observation.

In VVC, CUs have both square and rectangle shapes when using QT structure and MTT structure, respectively. Therefore, there are 16 types of block sizes are used in VVC. Table 2 shows the distribution of different block sizes in VVC. From the results obtained in Table 2, it can be concluded that:

- If the QP value increases, the percentage of large CUs (N×M with N and M > 16) also increases. Meanwhile, the percentage of small CUs (the remaining size) decreases when the QP value increases.

- In videos with flat textures, such as vidyo1, the large CUs are more used than in videos with complex textures, such as *BQTerrace*.

From the above observations, it is reasonable to design a deep learning method based on QP values to predict the CU size of VVC to reduce the computational complexity.

Figure 4. The first frame of videos used to perform analysis:
(a) BasketballDrillText, (b) BQTerrace, (c) vidyol.

Table 1. Experimental conditions

Parameters	Description
Platform	VTM-12.1
Configuration	All-Intra
QP	22, 27, 32, 37

To examine the VVC Intra mode selection, we perform a statistical analysis for encoding complexity of each stage in the intra mode prediction process. The results of this analysis are shown in Figure 5.

Figure 5 shows that in the intra mode prediction, RMD and MPM stages take about

20% of the total time while the checking RD-Cost stage takes up to 80%. Thus, if this stage is terminated early, the time consumption of the intra mode prediction can be reduced dramatically.

In addition, the optimal mode distribution is also considered in this paper. Table 3 shows the distribution of optimal modes with four QPs. The results show that on average, 29.43% CU selected Planar and DC mode as the optimal intra mode and higher in the video sequence with flat textures, such as vidyo1. Therefore, Planar and DC mode should be added to the final candidate list, which is then used in RDO process.

As informed above, the modes in the MPM list are derived from the neighbor blocks. Thus, the modes in MPM list have a high probability of being selected for the current block. Table 4 shows the distribution of four sequences in case the optimal mode of the RDO process belongs to the MPM list at different QP (22, 27, 32, 37). The results show that up to 75.87% of total blocks encoded in modes belong to MPM list. It can be concluded that there is a strong correlation between the Mopt and MPM list. Thus, in this work, a fast intra mode decision process is proposed to find an optimal mode among the modes in the MPM list and set a condition to early terminate this process.

3.2. Proposed Fast VVC Intra Structure

Figure 6 depicts the proposed fast intra mode decision process with two main stages: i) CU early termination stage decides CU size by using a CNN model and ii) Fast mode decision method (FMD) stage decides the intra prediction mode by using proposed Three-Steps Mode Decision algorithm.

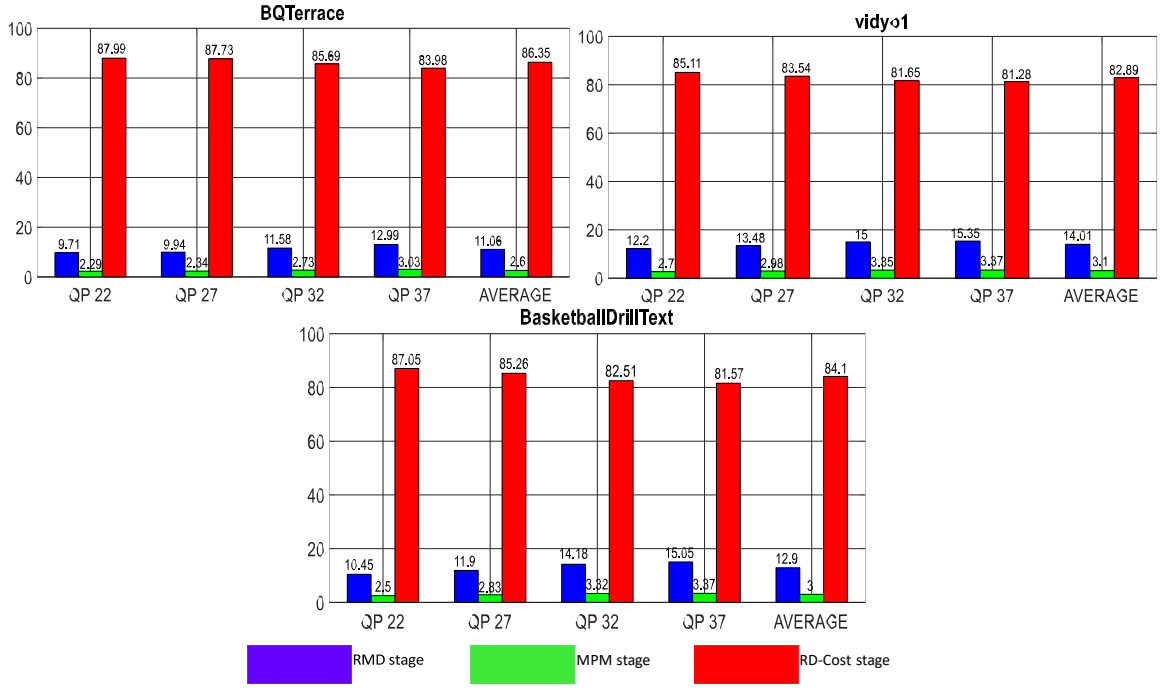

Figure 5. Time consumption of each stage in the intra mode prediction process.

Table 2. Percentage of CU in different sizes

Size	BasketballDrillText				BQTerrace				vidyol				Average			
	QP 22	QP 27	QP 32	QP 37	QP 22	QP 27	QP 32	QP 37	QP 22	QP 27	QP 32	QP 37	QP 22	QP 27	QP 32	QP 37
64x64	0	0	0	0.02	0.06	0.21	0.30	0.56	0.04	0.27	0.87	2.02	0.03	0.16	0.39	0.87
32x32	0	0.02	0.56	3.26	0.40	0.76	1.31	2.14	0.72	2.21	4.66	7.14	0.37	1.00	2.18	4.18
32x16	0.12	0.61	2.83	6.33	0.29	1.12	2.10	3.13	1.85	3.15	4.19	6.61	0.75	1.63	3.04	5.36
32x8	0.17	0.84	2.72	4.17	0.24	1.19	1.95	2.33	1.48	2.14	2.52	3.66	0.63	1.39	2.40	3.39
32x4	0.27	1.27	2.82	3.42	0.20	0.83	1.02	1.40	0.72	0.86	1.31	1.53	0.40	0.99	1.72	2.12
16x32	0.03	0.21	1.74	4.36	0.66	0.86	1.05	1.51	2.81	5.10	6.24	6.74	1.17	2.06	3.01	4.20
8x32	0.03	0.38	1.52	1.66	0.58	0.75	1.04	1.73	5.65	6.83	5.37	4.33	2.09	2.65	2.64	2.57
4x32	0.09	0.34	0.59	0.72	0.4	0.75	1.28	2.26	6.83	5.01	2.92	2.35	2.44	2.03	1.60	1.78
16x16	1.86	4.04	8.81	9.33	0.92	2.49	3.59	4.76	6.66	8.46	9.43	10.43	3.15	5.00	7.28	8.17
16x8	2.95	8.58	11.65	10.29	2.04	6.23	7.70	9.48	8.61	9.62	11.09	12.04	4.53	8.14	10.15	10.60
16x4	5.04	7.67	6.39	6.16	2.85	7.42	8.00	8.18	4.77	5.39	5.33	5.56	4.22	6.83	6.57	6.63
4x16	3.95	5.29	4.84	4.46	1.82	5.08	6.90	7.89	9.12	7.05	5.97	4.92	4.96	5.81	5.90	5.76
8x16	2.73	7.36	10.76	8.65	2.41	5.04	7.22	9.36	12.23	10.82	11.32	10.41	5.79	7.74	9.77	9.47
8x8	17.79	21.92	15.17	11.62	54.16	18.25	15.2	15.22	13.81	13.12	12.01	10.68	28.59	17.76	14.13	12.51
8x4	32.43	21.38	14.36	11.97	20.14	28.82	23.58	16.5	11.42	9.68	8.15	5.88	21.33	19.96	15.36	11.45
4x8	32.54	20.09	15.24	13.59	12.81	20.21	17.76	13.53	13.27	10.30	8.6	5.69	19.54	16.87	13.87	10.94

Table 3. Distribution of optimal intra mode

Sequence	QP	Planar Mode	DC Mode	Directional Mode
BasketballDrillText	22	11.66	3.37	84.97
	27	13.80	3.96	82.24
	32	16.99	5.17	77.85
	37	21.18	6.08	72.74
BQTerrace	22	24.42	7.92	67.66
	27	24.86	6.59	68.54
	32	25.19	6.36	68.45
	37	26.03	6.67	67.30
vidyo1	22	30.76	6.30	62.94
	27	28.17	5.90	65.93
	32	28.87	6.09	65.03
	37	30.35	6.52	63.13
Average		23.52	5.91	70.57

3.2.1. CU Early Termination

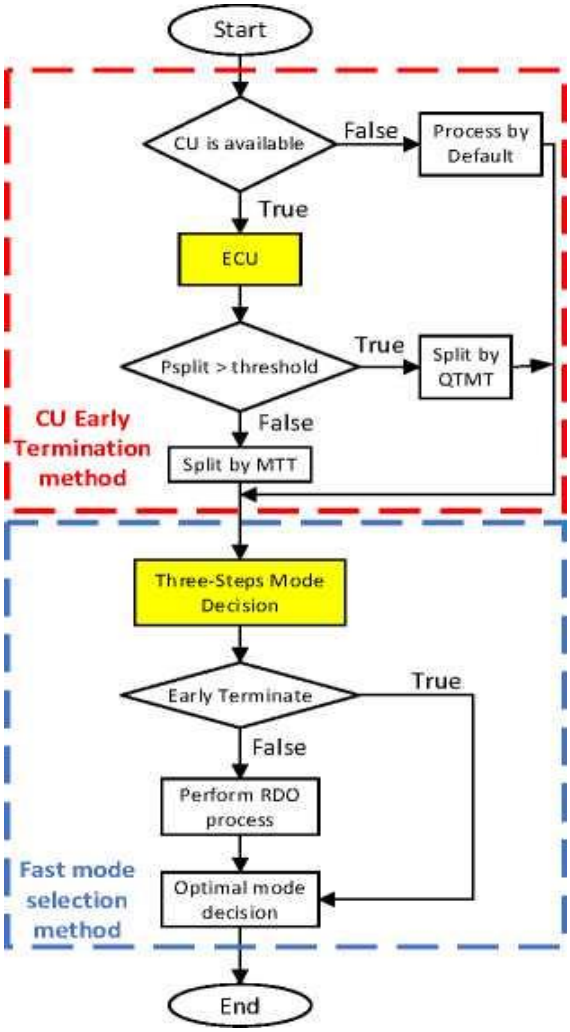

Figure 6. The flowchart of overall proposed method.

From the above CU size analysis, we introduce an early CU termination method following the machine learning fashion. In [12], content features are firstly extracted from video frame and then fed into a hierarchical CNN model (shown in Figure 6). After that, the CU size of HEVC intra-mode is predicted corresponding to the hierarchical CU partition map (HCPM).

Difference from the work in [13], to achieve high prediction accuracy, our method has adapted the loss function of hierarchical CNN model to fit the VVC architecture. Then the trained model is applied to predict the CU size of VVC intra-mode. In addition, we also discuss the optimization achievement associated to the selected loss function (see the appendix).

Figure 7 shows the architecture of the proposed ETH-CNN model. Initially, Luma channel of original CU size 64x64 is used as the input of CNN model. Then, inspired by the hierarchy of quadtree partition, input data is processed at three branches in parallel to predict the HCPM as an output. At each branch, data is firstly preprocessed by mean removal and downed sampling. Three level downed sampling (64x64, 32x32, 16x16) is applied with three parallel processing branches, respectively.

Preprocessed data keep going through three convolution layers at the respective branch to extract the video frame feature. The number of filters applied at each layer on each branch is 16 filters size 4x4 for the first layer, 24 filters size 2x2 for the second layer and 32 filters size 2x2 for the last one.

Table 4. Percentage of CUs have optimal intra mode belong to MPM list

Sequence	QP 22	QP 27	QP 32	QP 37	Average
BasketballDrillText	81.20	79.84	77.71	74.66	78.35
BQTerrace	80.55	77.14	75.87	75.18	77.18
vidyol	73.51	72.77	72.12	69.96	72.09
Average	78.42	76.58	75.23	73.27	75.87

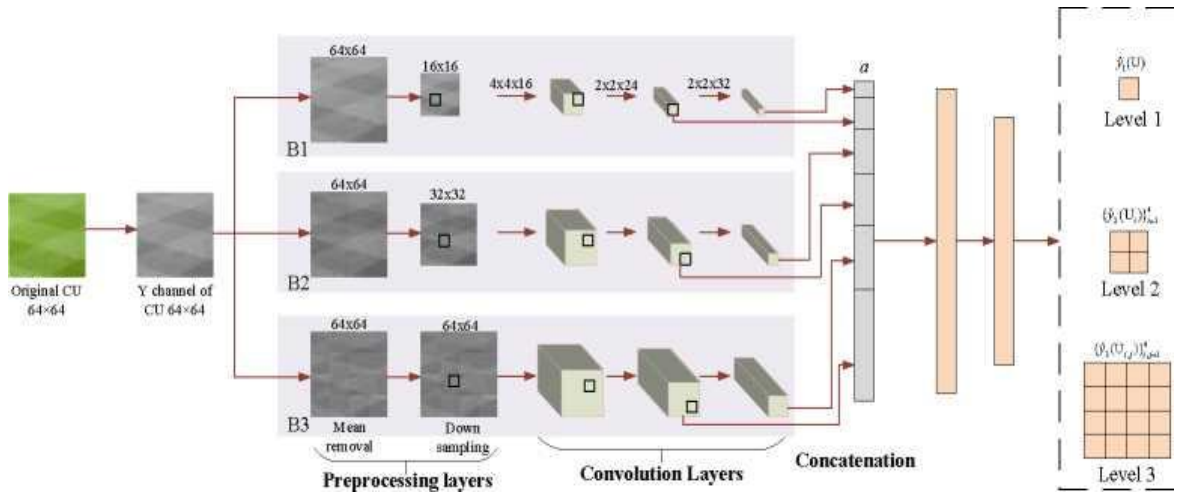

Figure 7. ETH-CNN architecture.

The extracted video frame features are concatenated as a vector a then processed through three fully connected layers to predict the output, HCPM level, based on QP values. In case the higher-level branch predicts that current CU is not divided, the calculation output of HCPM at lower-level branch is terminated.

Finally, the probabilities of labels $\hat{y}_1(\boldsymbol{U})$, $\hat{y}_2(\mathbf{U}_i)$ and $\hat{y}_3(\mathbf{U}_{i,j})$ corresponding to three levels of HCPM are calculated. These probabilities are then compared with a threshold (α_l) to decide whether current CU is split or not. Similarly, to the method in [12], in this work, the bi-threshold α_l is set equal to 0.5.

In this model, the Cross-Entropy loss function is used to calculate the accuracy. The accuracy is defined as the difference between the estimated probability and expected outcome. However, in this case, the labels are binary ("1" for splitting and "0" for non-splitting). Hence, a special version of Cross-entropy Loss called Binary Cross-Entropy (BCE) is used. It is the combination of a Sigmoid activation function and a Cross-Entropy loss function as shown in Equation (3).

$$\begin{aligned} BCE\ Loss &= H(y,\hat{y}) \\ &= -\frac{1}{output\ size}\sum_{k=1}^{output\ size} y_k \log(\hat{y}_k) \\ &+ (1-y_k)\log(1-\hat{y}_k) \end{aligned} \quad (3)$$

where y_k is the ground truth and $\hat{y}_k$ is theprediction probability of the k^{th} training sample. Because the ETH-CNN structure is divided into three branches, representing three levels of HCPM. Therefore, the Loss function L_r of eachsample in a dataset containing *R* training samples can be calculated as the sum of BCE Loss in branches as shown in Equation (4).

$$\begin{aligned} L_r &= H\left(y_1^r(U_i), \hat{y}_1^r(U_i)\right) \\ &+ \sum_{\substack{i\in\{1,2,3,4\} \\ y_2^r(U_i)\neq null}} H\left(y_2^r(U_i), \hat{y}_2^r(U_i)\right) \\ &+ \sum_{\substack{i,j\in\{1,2,3,4\} \\ y_3^r(U_{i,j})\neq null}} H\left(y_3^r(U_{i,j}), \hat{y}_3^r(U_{i,j})\right) \end{aligned} \quad (4)$$

where the ground truth dataset are $\{y_{1r}(U)$, $\{y_2^r(U_i)\}_{i=1}^4$ and $y_3^r(U_{i,j})\}_{i,j=1}^4$ and predicted dataset are $\{\hat{y}_1^r(U)$, $\{\hat{y}_2^r(U_i)\}_{i=1}^4$ and $\{\hat{y}_3^r(U_{i,j})\}_{i,j=1}^4\}_{r=1}^R$.

Then, the ETH-CNN model computes the loss function of the whole training dataset by Equation (5):

$$\boldsymbol{L = \frac{1}{R}\sum_{r=1}^{R} L_r} \quad (5)$$

Finally, to train the ETH-CNN model, a large database is used. For fast convergence, the stochastic gradient descent algorithm with momentum is applied to the optimization process.

In summary, the early CU termination method can be described as the following pseudo code:

Algorithm: Early CU Termination

Input: CU size & video content

Output: CU partition

1: **if** (CU size = 128×128) **then** split into 4 square sub-CUs

2: **if** (CU size = 64×64) or (CU size = 32×32) or (CU size = 16×16) **then**

3: Predicts the probability ($P_{splitQT}$) by ECU model

4: **if** ($P_{splitQT}$ > 0.5) **then**

Return CU is partitioned by QTMT

5: **else**

Return CU is partitioned by MTT

3.2.2. Fast Mode Selection

Since the modes in MPM list have a high probability to become the optimal mode of the RDO process. Thus, in the proposed method, the mode in MPM list which has the lowest RMD-cost is selected as the initial searching mode M_{init1}. Figure 8a is an example of the initial searching mode. There are six arrows representing six modes in MPM list. The red arrow is the mode with the lowest RMD-Cost and it is selected as the initial searching mode M_{init1}. After selecting the initial searching mode M_{init1}, the encoder performs "Three-Steps.

Search" process to estimate the best mode for current CU as the follows:

In the first step, the encoder calculates the RMD-Cost for the directional mode in the left and right of M_{init1}, with stride = 4. On the left side, the modes which are used to calculate RMD-Cost are in form of $M_{init1-4}$. On the right side, the modes which are used to calculate RMD-Cost are in form of $M_{init1+4}$. After calculating all of the satisfying modes, the mode having the lowest RMD-Cost is selected and set as the initial searching mode of the second step (M_{init2}). For example, in Figure 8b, the blue arrow is marked as the best mode and it is set to be M_{init2}.

In the second step, the RMD-Cost of the three modes containing M_{init2}, $M_{init2-2}$ and $M_{init2+2}$ are calculated. In the end of this step, the mode having the lowest RMD-Cost is selected and set as the initial searching mode of the final step (M_{init3}). In Figure 8c, the green arrow is marked as the best mode and it is set to be M_{init3}.

In the final step, two adjacent modes of M_{init3} are used to calculate RMD-Cost. Finally, the mode with the lowest RMD-Cost is collected to perform RDO process. It is noted that the directional mode is marked as M_d as in Figure 8d. If RMD-Cost of the directional mode M_d is equal to RMD-Cost of the initial mode M_{init3}, the process will be early terminated and the optimal intra mode of current CU is set to be M_d. On the other hand, if RMD-Cost of the directional mode M_d is different to RMD-Cost of the M_{init3}, the encoder will perform RDO process with the final candidate list containing Planar, DC and M_d.

After collecting the final candidate list, the encoder performs RDO process and selects the mode having the lowest RD-Cost as the optimal mode for CU.

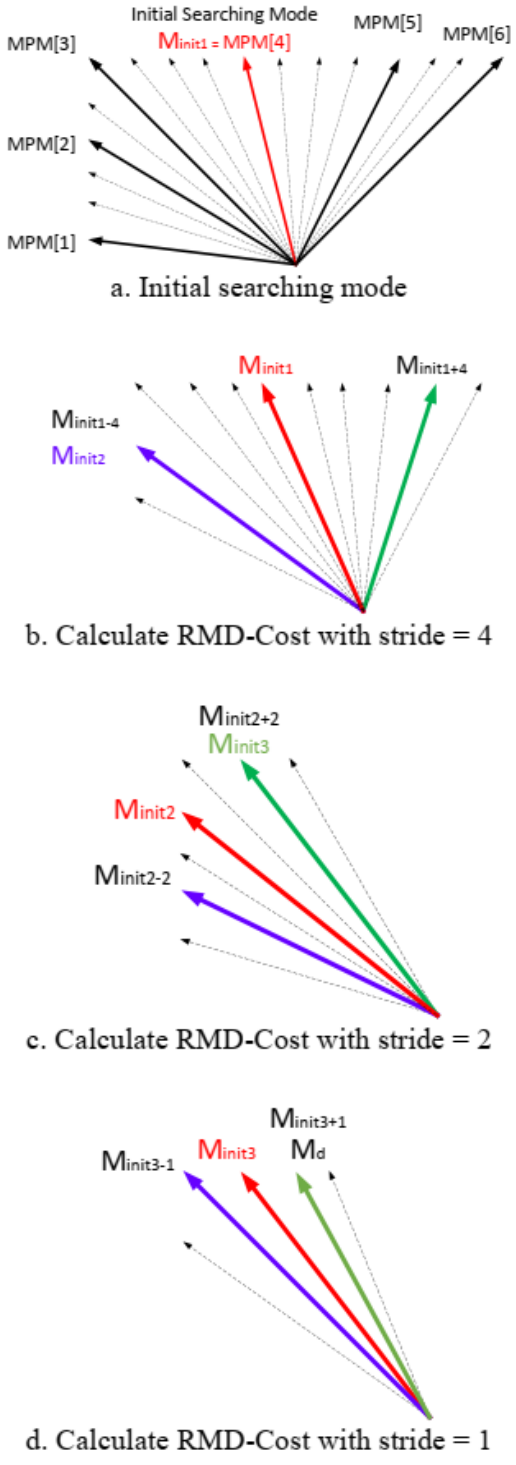

Figure 8. Illustration of the three-Steps Mode Decision process.

4. Experiment and Performance Evaluation

4.1. Dataset and Test Condition

In this experiment, 9 standard video sequences are encoded in All-Intra main configuration with different Quantization Parameters (QPs) 22, 27, 32 and 37. The characteristics of the video sequences are shown in Table 5 while the first frames of these sequences are illustrated in Figure 9.

For training the ETH-CNN model, the dataset in [12] with 2000 images compressed at four QPs is adopted.

4.2. Overall Results and Discussion

To evaluate performance of the proposed method, the encoding time of the proposed method is compared to the encoding time of VVC standard [20]. The time saving (*TS*) of the proposed method is defined by Equation (6):

$$TS = \frac{T_{Proposed} - T_{VVC}}{T_{VVC}} \times 100\% \quad (6)$$

where T_{VVC} represents the total encoding time of the VTM-12.1 [20], $T_{Proposed}$ represents the total encoding time of the proposed method. In addition, BDBR and BDPSNR [21] are computed to evaluate the performance of the proposed method compared to VVC. BDBR shows the difference in bitrate at the equivalent quality, while BDPSNR shows the difference in quality at the equivalent bitrate. The experiment results of time saving, BDBR, BDPSNR of all test sequences are shown in Table 6. Some conclusions can be derived as:

- The encoding time of VVC can be reduced by about 30.04% by using the proposed ECU method for All-Intra configuration. Meanwhile, the increase in BDBR is only 1.39%.

- The proposed FMD can achieve about 10.31% encoding time saving with the payment of 1.82% BDBR.

- In overall, the FMD and ECU bring about 50.17% of the encoding time saving while

asking for 3.74% of BDBR. This is acceptable for many real-time video applications.

Video sequence ParkScene achieves the highest encoding time saving (57.48% in overall). It can be explained that the content of this test sequence contains many complex areas so the proposed method can be used for a large number of CU.

- The reason why the proposed method can help save encoding time complexity is that ECU method reduces the number of recursions to calculate RDO and FMD method eliminates several modes in the candidate list with low probability of being selected as optimal mode.

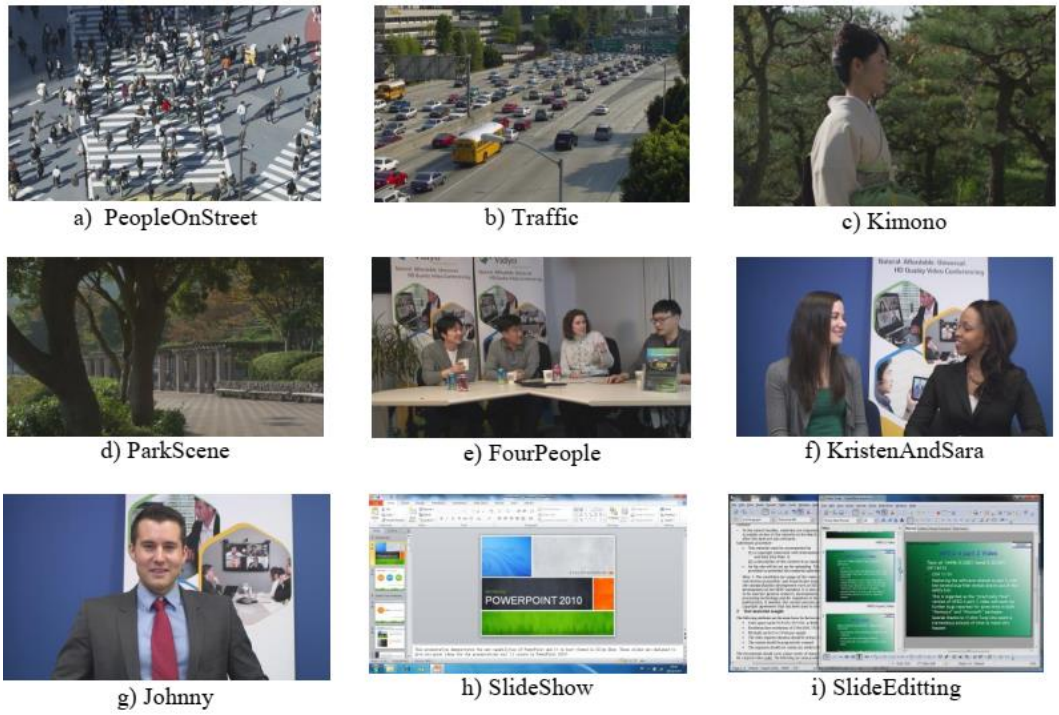

Figure 9. First frame of testing video sequences.

Table 5. The Details of test sequences

Sequence	Spatial Resolution	No. Frame	Frame Rate	Content type
PeopleOnStreet	2560x1600	150	30 Hz	Surveillance
Traffic	2560x1600	150	30 Hz	Surveillance
Kimono	1920x1080	240	24 Hz	Natural
ParkScene	1920x1080	240	24 Hz	Natural
FourPeople	1280x720	600	60 Hz	Conference
KristenAndSara	1280x720	600	60 Hz	Conference
Johnny	1280x720	600	60 Hz	Conference
SlideShow	1280x720	500	20 Hz	Screen content
SlideEditting	1280x720	300	30 Hz	Screen content

Table 6. Encoding Time Saving and BDBR Loss Comparison

Sequence	ECU			FMD			ECU+FMD		
	TS	BDBR	BDPSNR	TS	BDBR	BDPSNR	TS	BDBR	BDPSNR
PeopleOnStreet	-30.04	1.20	-0.06	-11.26	2.14	-0.11	-56.22	3.78	-0.20
Traffic	-28.39	0.88	-0.04	-11.63	1.65	-0.08	-53.09	3.71	-0.19
Kimono	-23.52	0.21	-0.01	-9.77	1.80	-0.06	-43.14	2.49	-0.08
ParkScene	-37.67	0.71	-0.03	-12.24	1.46	-0.06	-57.48	2.66	-0.12
FourPeople	-33.90	1.17	-0.06	-9.72	1.83	-0.10	-53.72	3.49	-0.18
KristenAndSara	-20.78	1.92	-0.09	-10.24	1.78	-0.08	-49.21	3.88	-0.18
Johnny	-33.66	1.15	-0.04	-10.28	2.30	-0.08	-47.41	3.61	-0.12
SlideShow	-33.84	3.00	-0.26	-8.88	2.00	-0.17	-46.43	5.18	-0.46
SlideEditting	-30.78	2.28	-0.31	-8.74	1.43	-0.21	-44.81	4.84	-0.67
Average	-30.04	1.39	-0.10	-10.31	1.82	-0.11	-50.17	3.74	-0.24

Table 7. Evaluation of the proposed algorithm compared to previous work

Sequence	Ref [14]		Proposed	
	TS	BDBR	TS	BDBR
Kimono	-15.16	5.73	-43.14	2.49
ParkScene	-21.32	5.85	-57.48	2.66
FourPeople	-21.82	6.73	-53.72	3.49
KristenAndSara	-23.68	8.82	-49.21	3.88
Johnny	-25.45	8.23	-47.41	3.61
Average	-21.49	7.07	-50.19	3.23

In addition, Table 7 shows the comparison between the proposed method and the method in [14] in terms of time saving and BDBR. The results show that the proposed method achieves a smaller time consumption as well as the bitrate than the previous method. It means that the proposed method provides a lower complexity solution as well as can reduce encoding complexity compared to the state-of-the-art video encoder H.266/VVC.

5. Conclusion

In this paper, to achieve low complexity VVC intra coding, we propose a CNN based early CU splitting algorithm in which a novel loss function is deployed together with a hierarchical network. The employed CNN model is fed into the latest VVC test model and shows that 30% saving time can be achieved. In addition, considering the high correlation between neighboring directional modes, we introduce a novel three-steps mode decision method. Independently, the fast mode decision (FMD) method achieves an additional 10% time saving. In general, both ECU and FMD algorithms help to save around 50% time saving with the pay of 3.7% BDBR loss. For future work, a wiser FMD method can be investigated by selecting better set of initial coding mode.

Acknowledgement

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2020.15.

References

[1] G. J. Sullivan, J. Ohm, W. Han, T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) Standard, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, 2012, pp. 1649-1668.

[2] B. Bross, J. Chen, J. R. Ohm, G. J. Sullivan, Y. K. Wang, Developments in International Video Coding Standardization After AVC, with an Overview of Versatile Video Coding (VVC), Proceedings of the IEEE, Vol. 109, No. 9, 2020, pp. 1463-1493.

[3] F. Bossen, X. Li, K. Suhring, K. Sharman, V. Seregin, AHG Report: Test Model Software Development (AHG3), Document JVET-V0003-V1, 22nd JVET Meeting, by Teleconference, 2021.

[4] X. H. Van, S. N. Quang, F. Pereira, Versatile Video Coding Based Quality Scalability with Joint Layer Reference, IEEE Signal Processing Letters, Vol. 27, 2020, pp. 2079-2083.

[5] X. H. Van, Statistical Search Range Adaptation Solution for Effective Frame Rate Up-conversion, IET Image Processing, Vol. 12, No. 1, 2018, pp. 113-120.

[6] X. H. Van, H. H. Nguyen, Enhancing Quality for VVC Compressed Videos with Multi-Frame Quality Enhancement Model, 2020 International Conference on Advanced Technologies for Communications (ATC), Nha Trang, 2020, pp. 172-176.

[7] Q. Zhang, Y. Zhao, B. Jiang, L. Huang, T. Wei, Fast CU Partition Decision Method Based on Texture Characteristics for H.266/VVC, IEEE Access, Vol. 8, 2020, pp. 203516-203524.

[8] Z. Jin, P. An, C. Yang, L. Shen, Fast QTBT Partition Algorithm for Intra Frame Coding through Convolutional Neural Network, IEEE Access, Vol. 6, 2018, pp. 54660-54673.

[9] H. Yang, L. Shen, X. Dong, Q. Ding, P. An, G. Jiang, Low-Complexity CTU Partition Structure Decision and Fast Intra Mode Decision for Versatile Video Coding, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 30, No. 6, 2020, pp. 1668-1682.

[10] Y. Chen, L. Yu, H. Wang, T. Li, S. Wang, A Novel Fast Intra Mode Decision for Versatile Video Coding, Journal of Visual Communication and Image Representation, Vol. 71, 2020, pp. 102849.

[11] S. Dargan, M. Kumar, M. R. Ayyagari, G. Kumar, A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning, Arch Computational Methods in Engineering, Vol. 27, 2020, pp. 1071-1092.

[12] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, Z. Guan, Reducing Complexity of HEVC: A Deep Learning Approach, IEEE Transactions on Image Processing, Vol. 27, No. 10, 2018, pp. 5044-5059.

[13] X. H. Van, S. N. Quang, M. D. Bao, M. D. Ngoc, D. T. Duong, Fast QTMT for H.266/VVC Intra Prediction using Early- Terminated Hierarchical CNN model, 2021 International Conference on Advanced Technologies for Communications (ATC), Nha Trang, 2021, pp. 195-200.

[14] B. Abdallah, F. Belghith, B. Ayed, N. Masmoudi, Low-complexity QTMT Partition Based on Deep Neural Network for Versatile Video Coding, Signal, Image and Video Processing, Vol. 15, 2021, pp. 1153-1160.

[15] R. Bhandari, A. Vyas, Analysis the Performance of Three Step Search Algorithm for Motion Estimation, International Journal of Engineering Research & Technology, Vol. 2, No. 3, 2014, pp. 146-150.

[16] I. Kim, J. Min, T. Lee, W. Han, J. Park, Block Partitioning Structure in the HEVC Standard, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, 2012, pp. 1697-1706.

[17] J. Chen, Y. Ye, S. H. Kim, Algorithm description for Versatile Video Coding and Test Model 12 (VTM 12), Document JVET-U2002, 21st JVET Meeting, by teleconference, 2021.

[18] High Efficiency Video Coding (HEVC), Rec. ITU-T H.265 and ISO/IEC 23008-2, 2013 (and Later Editions).

[19] N. Zouidi, F. Belghith, A. Kessentini, N. Masmoudi, Fast Intra Prediction Decision Algorithm for the QTBT Structure, 2019 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), Gammarth, 2019, pp. 1-6.

[20] VVCSoftware_VTM-12.1, [Online], Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware VTM/-/tree/VTM-12.1 (accessed on: April 5th, 2021).

[21] G. Bjontegaard, Calculation of Average PSNR Differences Between RD Curves, Document VCEG-M33, 13th ITU-T VCEG Meeting, VCEG, Austin, 2001.

[22] P. M. Gruber, Convex and Discrete Geometry, 1st ed., Springer Berlin, Heidelberg, 2007.

Appendix: BCE Loss Convexity

For the most efficient model optimization, it is necessary to examine the convexity of the selected loss function.

The BCE loss function on the single sample is as:

$$BCE\ Loss = H(y,\hat{y}) = -(y\log(\hat{y})) + (1-y)log\ (1-\hat{y})) \quad (7)$$

Where y is the ground truth label while $\hat{y}$ is the predict label. Because the factor y is binary, we have Equation (7).

$$BCE\ Loss = \begin{cases} -\log(\hat{y}) & if\ y = 1 \\ -\log(1-\hat{y}) & if\ y = 0 \end{cases} = -\log(x) = f(x) \quad \forall x\epsilon(0,1] \quad (8)$$

Thus, $f''(x)$ can be computed as in Equation (8)

$$f''(x) = \frac{1}{x^2 \ln(10)} > 0 \quad \forall x\epsilon(0,1] \quad (9)$$

Because xϵ(0,1], x belongs to the convex set (*C*) Thus, Follow by Corollary 1.1 in [22], *f(x)* will be convex. Thus, BCE loss is convex.

For the case of multiple samples, follow the definition of convex function in [22]. Let *f*: $C \to R$ be a real function on *C*. The function f is convex if *C* is convex and

$$f\big((1-\lambda)x + \lambda y\big) \le (1-\lambda)f(x) + \lambda f(y) \quad \forall x, y \in C, 0 \le \lambda \le 1 \quad (10)$$

Now, let's assume *h(x) = f(x) + g(x)* where *f* and g are convex. *h(x)* is convex can be proved as:

$$f\big((1-\lambda)x + \lambda y\big) \le (1-\lambda)f(x) + \lambda f(y) \quad (11)$$

$$g\big((1-\lambda)x + \lambda y\big) \le (1-\lambda)g(x) + \lambda g(y) \quad (12)$$

$$\Rightarrow f\big((1-\lambda)x + \lambda y\big) + g\big((1-\lambda)x + \lambda y\big) \le (1-\lambda)f(x) + \lambda f(y) + (1-\lambda)g(x) + \lambda g(y) \quad (13)$$

$$\Rightarrow f\big((1-\lambda)x + \lambda y\big) + g\big((1-\lambda)x + \lambda y\big) \le (1-\lambda)\big(f(x) + g(x)\big) + \lambda(f(y) + g(x)) \quad (14)$$

$$\Rightarrow h\big((1-\lambda)x + \lambda y\big) \le (1-\lambda)h(x) + \lambda h(y) \quad (15)$$

Thus, *h(x)* is the convex function. Here, the loss function on the multiple samples is convex function.