

Article

Faster Intra-Prediction of Versatile Video Coding Using a Concatenate-Designed CNN via DCT Coefficients

Sio-Kei Im ^{1,2}  and Ka-Hou Chan ^{1,2,*} ¹ Faculty of Applied Sciences, Macao Polytechnic University, Macau 999078, China; marcusim@mpu.edu.mo² Engineering Research Centre of Applied Technology on Machine Translation and Artificial Intelligence, Macao Polytechnic University, Macau 999078, China

* Correspondence: chankahou@mpu.edu.mo

Abstract: As the next generation video coding standard, Versatile Video Coding (VVC) significantly improves coding efficiency over the current High-Efficiency Video Coding (HEVC) standard. In practice, this improvement comes at the cost of increased pre-processing complexity. This increased complexity faces the challenge of implementing VVC for time-consuming encoding. This work presents a technique to simplify VVC intra-prediction using Discrete Cosine Transform (DCT) feature analysis and a concatenate-designed CNN. The coefficients of the (DTC-)transformed CUs reflect the complexity of the original texture, and the proposed CNN employs multiple classifiers to predict whether they should be split. This approach can determine whether to split Coding Units (CUs) of different sizes according to the Versatile Video Coding (VVC) standard. This helps to simplify the intra-prediction process. The experimental results indicate that our approach can reduce the encoding time by 52.77% with a minimal increase of 1.48%. We use the Bjøntegaard Delta Bit Rate (BDBR) compared to the original algorithm, demonstrating a competitive result with other state-of-the-art methods in terms of coding efficiency with video quality.

Keywords: versatile video coding (VVC); high-efficiency video coding (HEVC); discrete cosine transform (DCT) concatenate-designed CNN



Citation: Im, S.-K.; Chan, K.-H. Faster Intra-Prediction of Versatile Video Coding Using a Concatenate-Designed CNN via DCT Coefficients. *Electronics* **2024**, *13*, 2214. <https://doi.org/10.3390/electronics13112214>

Academic Editor: Stefanos Kollias

Received: 8 April 2024

Revised: 27 May 2024

Accepted: 28 May 2024

Published: 6 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, there has been an increased demand for immersive video experiences, particularly in the form of multimedia. This is because live videos provide viewers with a real-time visual experience and feeling that recorded videos cannot match. For instance, people watch live football matches and use video chat to communicate with distant family and friends directly, and corporations use video conferencing to hold meetings with staff in different places. Also, intra-frame techniques decode the highest-quality frames, while their coding time is controlled and predictable, and intra-frame data do not depend on data references from other frames (in inter-frames), allowing them to maintain a high level of data integrity. In contrast, inter-frames involve identifying the region in the reference frame that is most similar to the current CU and then calculating their differences or residuals. It is clear that intra- and inter-prediction are two very different techniques. Therefore, intra-frame frames may be more suitable for important scenarios such as real-time communications or low-latency broadcasting, where instant recovery of frames is critical. To enhance the viewing experience, live videos should have high resolutions, typically 1080p or more. The Cisco Visual Networking Index (VNI) predicted that by 2023, more than 66.0% of connected TVs would be 4K, and 1080p and 4K video would account for 95.0% of global video traffic. In practice, live videos with high resolution and extensive data can pose certain challenges. Broadcasting numerous live videos daily on both television and the internet encounters two main challenges. Firstly, the available bandwidth is often insufficient to handle the high bitrate required for transmitting such videos. Secondly, live videos

require low-delay encoding, meaning that each frame must be encoded quickly. Fortunately, the next-generation Versatile Video Coding (VVC) standard [1] effectively addresses the first challenge. Research has demonstrated that VVC has achieved a significant reduction in bitrate, saving between 30.0% and 50.0% compared to the previous High-Efficiency Video Coding (HEVC) standard [2], while maintaining similar video quality. However, the efficiency of VVC depends greatly on the use of coding techniques, which significantly increases the complexity of coding as more advanced techniques are applied [3]. The more advanced techniques used, the greater the coding complexity, which can make encoding each frame time-consuming for higher resolutions [4]. This complicates the resolution of the second challenge.

Several studies have been conducted to address the challenge of managing the encoding complexity of VVC. One proposed approach to mitigate this complexity is to identify the specific number of frames with restricted complexity according to the desired level. However, the enhancements achieved are modest, ranging from 60.0% to 100.0%. Also, a method for analysing complexity in VVC was introduced in [5], contrasting and overcoming the constraints of the approach presented in [6] while enabling a broader spectrum of complexity control [7]. However, this study has two shortcomings. Its complexity control only occurs at the first level, where all control parameters are fixed throughout the coding process, resulting in insufficient control accuracy [8–10]. Secondly, when using the Quantisation Parameter (QP), the Coding Unit (CU) is usually split into smaller units, leading to more messages being qualitatively represented during the entropy process of CABAC [11,12]. Using multiple estimators in coding results in more coding time and bitstreams being generated, which causes an unnecessary increase in the bitrate and objective delay [13,14]. To address these issues, a weighted combination of CABAC estimators is introduced to provide a faster method for entropy coding of VVC. Ref. [15] introduced a fast intra method to reduce the coding complexity by removing non-promising modes. Next, an entropy-based method was proposed to speed up CU partitioning in [16], and a fast block partitioning method was proposed by [17], which aimed to skip the CU partitioning and rate distortion consideration process by using an optimal coding tool selection. In order to extract and exploit features more efficiently, some methods have been proposed to accelerate CU partitioning based on intra-frame selection. For JVET intra coding, ref. [18] proposed a CNN-based fast partitioning method. Additionally, ref. [19] proposed a fast inter-coding method to terminate the CU partitioning process early by jointly using multi-domain information. This approach allows the estimator to be based on the current coding conditions, such as QP and CU size, thus enabling adaptive prediction accuracy.

Figure 1 displays samples encoded by VVC Test Mode (VTM-16.0) using different QP values in intra mode. The encoded frames with $QP \in \{22, 27, 32, 37\}$ are split into a total of 52,841, 42,516, 37,964, and 34,082 CUs, respectively. The complexity allocation can be adjusted based on the importance of different regions, which reduces the target encoding complexity while maintaining video quality. Motivated by this, this work presents a DCT-based complexity analysis approach for faster video encoding and a pre-processing algorithm based on the relationship between QP and DCT coefficients, which explores the DCT feature by using a concatenate-designed CNN for CU split prediction.

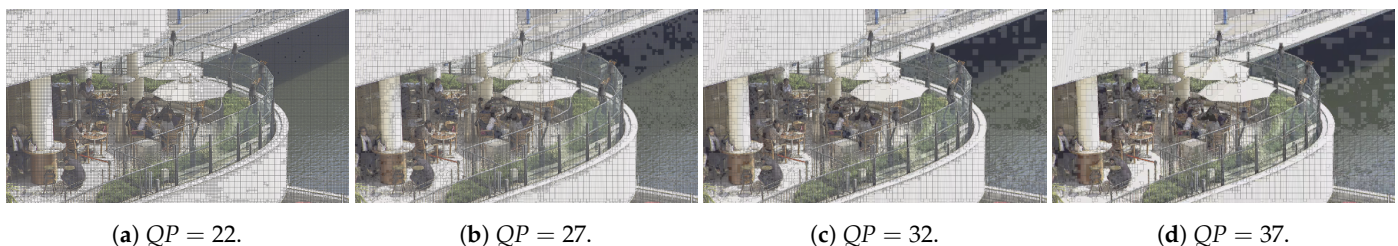


Figure 1. The complexity of the CU structure is highly dependent on the QP values that control the video quality, so more complex parts are split into smaller CUs.

The organisation of this article is briefly described below: Section 2 provides an overview of related work on faster intra-prediction approaches and their integration with CNN technology. Section 3 describes the architecture of the proposed concatenate-designed CNN for the CU Partition model, including its core technical work, in detail. Section 4 presents an introduction to the test sequences used, the implementation and configuration of the experiment, and comparisons with other methods. Finally, the article is concluded in Section 5.

2. Related Works

Extensive research has been conducted on the encoding complexity of VVC since its release in July 2020. The VVC Testing Model (VTM) has also been officially released. The VTM includes both encoder and decoder functionality, which outperforms HEVC in compression performance due to the introduction of 67 intra-frame prediction modes and the quad-tree division structure [20,21].

2.1. Heuristic Approach

The complexity of the division mode decision also increases to accommodate various texture structures [22]. The CU division involves calculations in both the top-down and bottom-up directions to achieve a logical structure. The partitioning process takes place during the top-down step by traversing the parent CU and applying the partition mode list. The rate-distortion cost is recorded under each partition mode until the leaf node is reached. In turn, the bottom-up stage occurs between the rate-distortion costs of the child CU and the parent CU. The division mode with the smallest rate-distortion cost is then selected as the optimal division structure for the current division structure [23]. Therefore, the two-way process has improved compression efficiency and also increased computational complexity in intra-prediction.

2.2. Statistical Method

To ease the complexity of VVC, a number of studies have aimed to simplify VVC by exploring the Coding Tree Unit (CU) partitioning schemes, which have a significant impact on coding complexity. References [24–26] investigated methods for optimising the selection of CU sizes during the encoding block partitioning process to reduce complexity. Based on this work, ref. [27] introduced an early CU depth prediction approach at the frame level. The aim of this approach is to obtain the Rate-Distortion Optimisation (RDO) search process by skipping less-frequently-used CU depths from previous frames, skipping the middle frames from inter-mode prediction [28]. Also, many studies primarily focus on two major research directions: quality enhancement and complexity control [9,29]. Similar strategies have been proposed in the literature for the CU plane, skipping the prediction of non-square units when the CU size is small [30]. These strategies focus on limiting the search area for CU depths based on the depth information of neighbouring CUs, resulting in reduced coding complexity but lower quality. In contrast to the CU skipping approach, ref. [31] introduced a fast CU size selection method for complexity reduction in VVC based on a Bayesian decision rule [32]. This technique makes use of computationally efficient features to quickly and accurately determine CU sizes by minimising the Bayesian risk of the rate distortion cost, by considering the current CU and comparing it with the splitting threshold to decide whether the current CU should be further split.

2.3. Deep Learning Techniques

In addition, ref. [33] proposed a k-nearest neighbour approach, which adapts the pyramid motion divergence-based technique according to the texture features contained in the coding block; the texture distribution position and texture complexity in the current CU can be determined, so that some bad candidate splitting modes can be discarded early. To improve the CU selection process, ref. [34] presented methods for determining the size of the prediction unit (PU) and transform units (TUs) early in the inter-prediction process.

Moreover, ref. [35,36] examined the Coding Block Flag (CBF) and RD cost of the current PU to stop the prediction process of the subsequent PU to reduce complexity. In addition to CU, PU, and TU size decisions, there are other components of VVC that increase coding complexity, such as multidirectional intra-predictions and multiple estimators in entropy coding. In order to reduce the complexity of VVC, ref. [37] introduced a coarse mode search scheme to selectively check for potential modes and provide fast intra mode decisions for the VVC encoder. In addition, ref. [38] developed a dynamic estimator selection algorithm for CABAC that simplifies the complexity process for all estimators' calculation. In addition to these methods, machine learning and data mining techniques have been used to reduce the complexity of VVC [39–41].

Inspired by these studies, this article makes use of the neural network approach to determine the CU coding structure for faster intra-prediction. This approach takes into account the stacking of convolutional layers within a CNN extractor and presents a concatenate-designed CNN connection to improve accuracy. Moreover, the predicted coding block size is further passed to the CABAC algorithm, which is used for the weighted combination of multiple estimators during entropy coding. This article demonstrates a traversal analysis of the depth characteristics of 23 video sequences and the distribution features of 67 intra-frame modes under Common Test Conditions (CTCs) (details in Section 5).

3. Complexity Analysis

To achieve an effective trade-off between video quality and bitrate, it is necessary to perform a detailed analysis of the operation and features of the VTM encoder. The computational complexity of the encoder increases with larger CU sizes and prediction modes. The complexity of the texture reflecting the CU is also higher with more coefficients after DCT transformation. The DCT matrix after quantisation can be used to predict the coding complexity of the VVC. In practice, there is an inverse relationship between the selected CU size and the DCT coefficients when there are subtle changes in the local texture. Therefore, an efficient intra-frame fast VVC method can be designed by using a CNN for the prediction of the CU and its quantised DCT coefficient for fast intra-frame prediction.

3.1. CU Size Distribution Analysis in VVC

This work first presents the CU size distribution under various QPs; we also select the 21 test sequences provided by the CTCs, considering their class, resolutions, bit depths, and texture features to ensure the reliability and feasibility of the experiments. Note that their depth information is a key feature when it comes to CU size classification. The recursive partitioning features of CUs mean that CU depth and size are closely related: if a CU adopts a quad-tree partitioning structure, further partitioning will not include QT partitioning. With knowledge of the depth information of a CU, its current size information can be inferred from the depth information of the intra-prediction. In standard VTM encoding, the largest and smallest CU sizes are 64×64 and 4×4 , respectively. The CU size is expressed by 2^{7-d} , where $d = 1, 2, 3, 4, 5$ denotes the depth of the quad-tree. The optimal CU numbers with respect to $QP \in \{22, 27, 32, 37\}$ are shown in Figure 2.

For VVC intra-prediction, larger CU sizes are usually suitable for simple content regions, while complex and variable regions require smaller CU sizes. This optimised CU partitioning leads to a more detailed feature description, but results in a more complex CU structure. To simplify the distribution analysis of the various CU sizes, Figure 2 statistically illustrates all optimal CU sizes with different QP values following CTCs. Table 1 gives the average results for different CU sizes: 27.80%, 24.90%, 25.91%, 16.21%, and 5.18%. This indicates that the larger-sized CUs ($d = 1, 2, 3$) are selected as optimal CUs for most regions. As the CU size decreases, the distribution of the different classes becomes smaller. As QP increases, the distribution of CUs in $d = 1, 2$ and $d = 3, 4$ becomes larger and smaller by about 10%, respectively. Therefore, it is preferable to choose a small transform unit size to preserve all the details of a frame when the QP value is low. A large transform unit size is

recommended to preserve few details when the QP value is high. This case is represented by class C, D, E, and F sequences with high motion and rich textures. The maximum and minimum of CUs under $d = 1, 2$ are 46.26% and 28.57%, respectively, at $QP = 37, 32$. The maximum distribution of CUs in $d = 3, 4$ is only 25.32% and 16.11% in $QP = 27, 22$. This indicates that different QPs have a significant impact on the optimal CU selection for the same coding regions. This may be due to the fact that small transform blocks are often used in this type of video to preserve image detail.

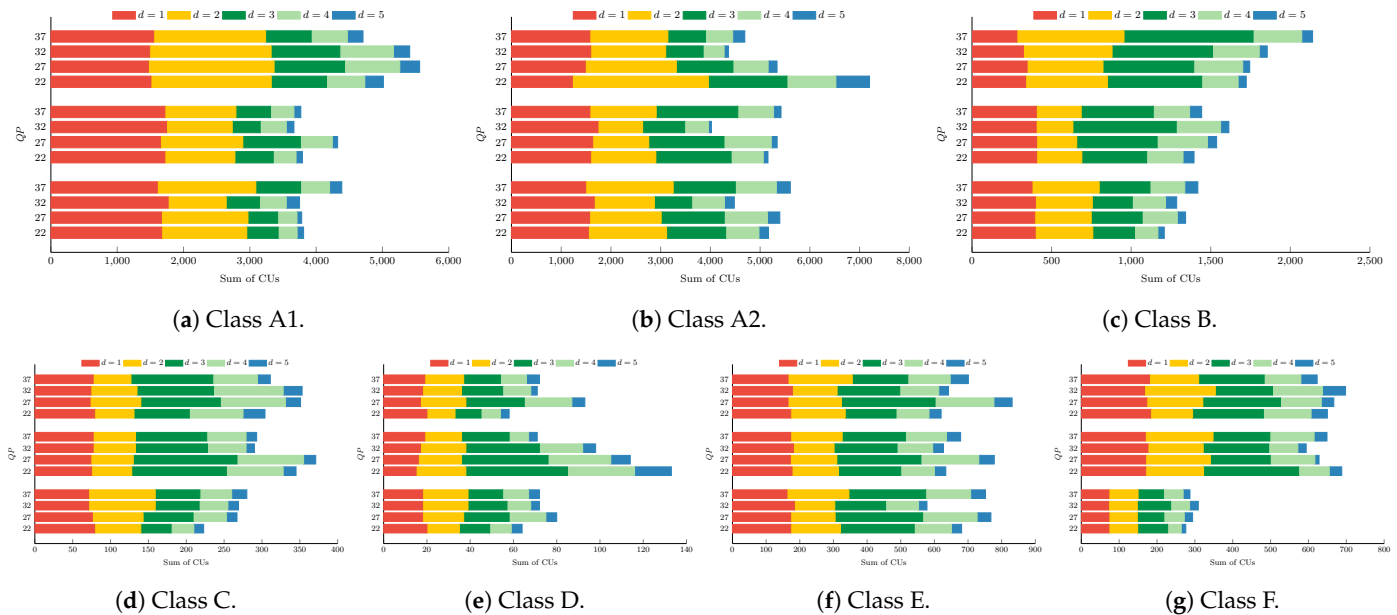


Figure 2. Distribution of various CU sizes with respect to different QPs .

Table 1. Mean number (percentage) of optimal CU structures at various depths (d).

Class	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$	Total CUs
A1	1630 (38.77)	1379 (31.46)	675 (15.26)	480 (10.78)	167 (3.73)	4331
A2	1558 (30.96)	1525 (29.10)	1182 (22.58)	707 (13.54)	210 (3.83)	5182
B	373 (24.84)	397 (25.27)	481 (30.01)	249 (15.95)	59 (3.93)	1559
C	75 (25.11)	63 (20.98)	86 (27.68)	61 (19.52)	21 (6.71)	306
D	18 (23.28)	17 (21.51)	25 (29.21)	16 (19.10)	6 (6.91)	83
E	173 (25.41)	148 (21.54)	204 (29.31)	127 (18.15)	39 (5.59)	691
F	140 (26.24)	127 (24.43)	146 (27.31)	87 (16.44)	30 (5.67)	530
Avg.	(27.80)	(24.90)	(25.91)	(16.21)	(5.18)	

3.2. Impact on QP and DCT Coefficients

The DCT coefficient reflects the complexity of the original content. The objective of DCT transformation is to transfer features from the colour domain to the frequency domain, as presented in Figure 3. The top-left coefficient (the yellow one in the second matrix) is the DC coefficient with zero frequency in both dimensions, which contains the most energy and usually has a large value. The remaining coefficient is the AC, with smaller values and located in the bottom-right, which represents the high-frequency components of the coding block. A higher value at the bottom-right side indicates a more complex original texture requiring a more complex CU structure.

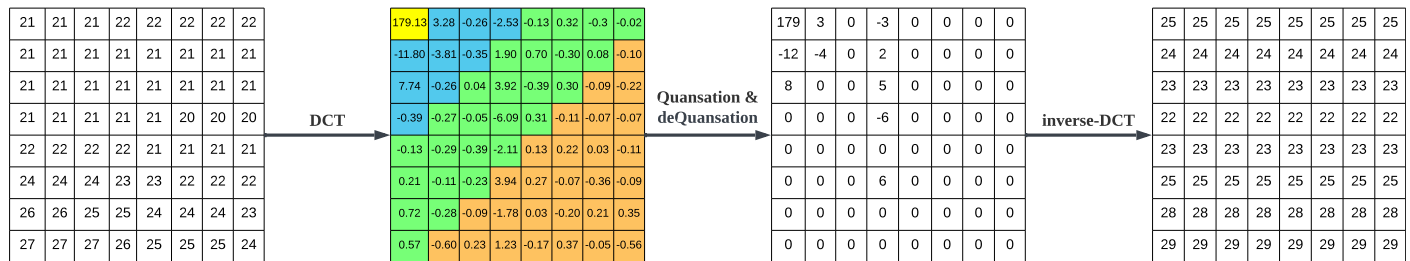


Figure 3. The computation of encoding and decoding features. The original colour is projected into the frequency domain using DCT transformation. The zero, low, medium, and high frequencies are, respectively, filled by yellow, blue, green, and orange colours. The resulting DCT coefficients are then passed to quantisation to reduce the higher-frequency domain, resulting in a simplified feature for optimal CU determination.

In practice, the DCT coefficients have to pass to quantisation (scaled-down) during coding, with an adapted QP resulting in more significant scaling down. Most of the ACs with different frequencies will become zero or near-zero after quantisation, leading to the decoding content after inverse-DCT being less complicated and simplifying the optimal CU structure. In addition, Table 2 provides another perspective on the changes in the number of CUs when QP is increased. As expected, a higher $QP = 32, 37$ value will result in more higher-frequency (the orange colour) elements being discarded after the (de)quantisation process, simplifying the complexity of the original feature. Additionally, Table 2 also provides the derivative of the CU with respect to QP . It is evident that higher-resolution sequences (Classes A1, A2, and B) require more CUs because the DCT coefficients between the quantised DCs and ACs become more complex when a lower QP is used. The CU requirements of higher sequences also decrease significantly with increasing QP , most notably for $d = 4.5$ (see Table 1). The main reason for this is that the complexity of the feature decreases, thus eliminating the need for a smaller CU size to represent its content. This can effectively reduce the prediction time of the optimal CU structure.

Table 2. Mean number of optimal CU structures at various QPs .

Class	$QP = 22$	$QP = 27$	$QP = 32$	$QP = 37$	$\frac{d(CU)}{d(QP)}$
A1	4656	4288	4175	4006	−43.33
A2	5743	5360	5240	4289	−96.93
B	1768	1585	1442	1242	−35.07
C	348	308	290	267	−5.40
D	112	93	61	46	−4.40
E	802	710	635	596	−13.73
F	548	511	500	489	−3.93

4. Proposed Concatenate-Designed CNN for CU Partition

In order to simplify the calculation of VVC intra-frame prediction, this work makes use of a concatenate-designed CNN to extract CU pixel features [42], with the use of multiple classifiers to determine the VVC unit size well. The proposed approach extends the convolutional layer and adaptive kernel size into the CNN module with multiple classifiers to perform more feature extraction, enabling it to efficiently handle CUs of different sizes, not solely 64×64 . The accuracy of CU partition feature extraction can be improved by capturing the information of convolutional channels in the structure of fast CU partition. Therefore, a CU partition optimisation strategy is used to balance the complexity of VVC intra-frame prediction with the performance of rate-distortion. As illustrated in Figure 4, this work presents a dual-stream encoder branch to receive the colour

feature and its DCT coefficients of the input CU. Both input streams are connected to two concatenate-designed VGGreNets [43], with multiple classifiers employed for classification tasks. Also, the proposed model connects the intra-prediction component to the proposed neural model and receives its predicted results to determine the split or not. Since this model is well trained, that can provide an accuracy determination once it receives an input CU. This modification allows users to potentially skip the 67 intra-prediction modes (including Plannar, DC, and 65 angulars), releasing more encoding time for acceleration. Additionally, an attention mechanism is employed to collect the hidden feature within the VGGreNet process, producing the weight feature for the help with the determination of the final CU structure prediction.

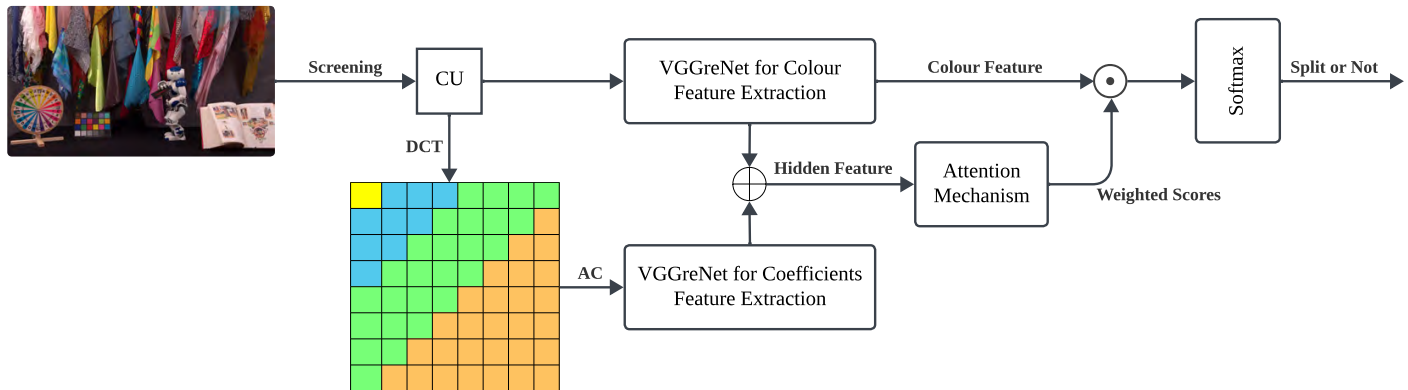


Figure 4. The main structure of the proposed model.

4.1. Feature Extraction by VGGreNet

To improve efficiency and reduce memory and energy consumption, this work presents a dual-stream VGGreNet network for feature extraction. The network performs CNN operations on both the CU colour feature and the DCT coefficients. The VGGreNet has applied some modifications to the connected CNN layer for the practical problem of CU partition. The colour channel is converted to YUV (luma and chroma) and sent to the first VGGreNet for visual extraction. Next, the CU feature is also transformed to the frequency domain by DCT transformation, with only the AC coefficients sent to the second VGGreNet, and the DC element is discarded. In fact, the DC element has a zero frequency that is meaningless for the original content's complexity. It usually has a larger value than the AC coefficients, which dilutes the AC weights after normalisation and causes more neural nodes to become dead neurons. By using VGGreNet, this extractor can adapt to CU inputs of various resolutions by adjusting the structure of the layer(s) and producing the main feature accordingly. These dual-stream networks cover both the colour and frequency of the CU and its split CUs.

As presented in Figure 5, The first three CNN layers contain a kernel range (3×3) stacked by the batch normalisation, representing a concatenate-designed CNN that can apply multiple classifiers to this structure (more discussion in Section 4.2). The final convolutional (CNN_4) presents a simple structure but is connected to a max-pooling layer with a reusable pooling layer (P_r) that performs recurrently. The max-pooling layer captures the most important feature within the context and discards the rest. By repeating this process, the most important feature can be obtained. Assuming that the input CU has a size of S and is halved by a max-pooling layer, the process can be expressed as follows:

$$S_{(t+1)} = \left\lceil \frac{S_{(t)}}{P_r} \right\rceil \quad (1)$$

where $L_{(t+1)}$ is connected to the reused CNN_4 layer if $L > 1$ repeatedly. In practice, VGGreNet has a deeper and lighter network structure that is adaptive. It incorporates a multi-branch structure consisting of multiple convolutional kernels with different sample

sizes. This allows it to capture features at different scales while reducing the space and time complexity of the computation. Note that a Fully Connected (FC) layer is connected to the reusable convolutional layer, ensuring the produced feature sizes are consistent to avoid errors in subsequent processes. This layer projects the features into the colour channel and the hidden feature produced by multiple classifiers. It prepares the colour feature ($\mathbf{c}$) for further production for the final decision.

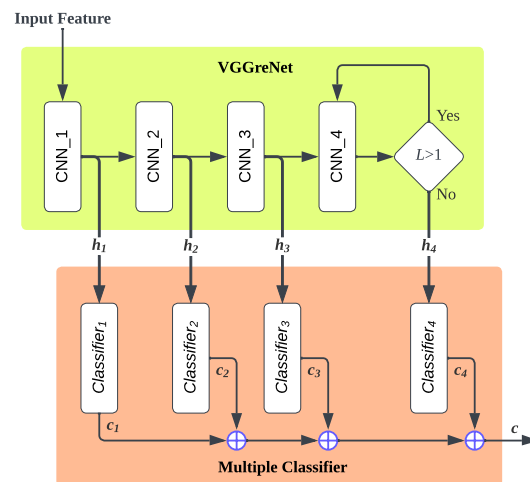


Figure 5. An illustration of feature extraction by VGGreNet, and the process of collecting these features using the concatenate-designed CNN with multiple classifiers.

4.2. Multiple Classifiers for Concatenate-Designed CNN

In traditional CNN-based classifiers, the FC layer only receives the output of the final CNN layer and makes the final prediction by Softmax. This output contains a large number of local features extracted by the CNN, but lacks global content information. This is because the CNN layer focuses on extracting local features from the local texture and is inherently unable to capture/ignore the global context of the entire input. Therefore, when passing hidden features to the final classifier, it is not possible to obtain a complete view of the entire information by relying only on the filtered features. To overcome this issue, this paper proposes the use of multiple classifiers to enhance the performance of the classifier. As presented in Figure 5, there are four classifiers employed to correct the hidden features from the CNN layer within the VGGreNet. The objective is to collect hidden features from each CNN layer and integrate them into the classification process. This will provide the final prediction with additional information to make a better decision. The complete form of the proposed $Classifier_i$ is expressed as

$$\begin{aligned} Classifier_i(h_i) &= FC(AdaptiveMaxPool(h_i)) \\ c_i &= Softplus(Classifier_i(h_i)) \end{aligned} \quad (2)$$

As described in (2), each classifier is composed of FC layers, which are essentially several linear layers stacked on top of each other. The primary objective of these layers is to project the input h_i into a fixed size (128×128) output c_i by applying an AdaptiveMaxPool layer. In practice, this work further adapts the Softplus activation function to generate active neurons from the original information. In contrast with other feature, primarily extraction, approaches, each proposed classifier focuses only on extracting features based on its recently received hidden feature (h_i). This process improves the accuracy of the classifiers and provides an additional reference for the final determination. The extracted features can take into account both the global structure and local patterns, resulting in more detailed and accurate information contained in the colour feature $\mathbf{c} = \{c_1, c_2, \dots, c_i\}$. Here $\mathbf{c}$ contains $i \times 128$ feature sizes.

4.3. Attention Mechanism for CU Splitting

An attention mechanism is an advanced method for extracting features. It allows the model to focus on specific parts of the input while generating the output using a weight scoring algorithm. This produces weighted features that enhance meaningful connections and discard noise between different states. In this study, the proposed attention mechanism is connected to the hidden features from VGGreNet. By using the weight feature of attention, the model can identify the image components that have the most significant impact on a particular prediction or classification. These weighted connections offer valuable insight into the model's focus and decision-making process, improving interpretability and understanding. With respect to Figure 4, the h_i^{YUV} and h_i^{DCT} , respectively, denote the hidden features generated by the two VGGreNets, which are passed to the attention mechanism by the following combination:

$$h_i = h_i^{YUV} \oplus h_i^{DCT} \quad (3)$$

The attention mechanism thus receives a set of hidden states, $\mathbf{h} \in \{h_1, h_2, \dots, h_i\}$, to consider the weight values. The proposed formula integrates the attention mechanism to weigh each h_i and calculate its $score_i$ of the current hidden feature as

$$score_i = L2 \left(\sqrt{\exp \left(\frac{(\mathbf{W}_Q \mathbf{h})(\mathbf{W}_K \mathbf{h})^T}{\sqrt{s}} \right)} \right) (\mathbf{W}_V \mathbf{h}) \quad (4)$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V \in \mathbb{R}^{128 \times 128}$, respectively, represent the query, key, and value matrix introduced by [44], and s denotes the variance in $\mathbf{h}$. This approach uses attention-based computation to model interactions between different CU sizes, based on a combination of DCT and YUV features. This approach allows for the acquisition of global information and contextual information. Additionally, low-frequency DCT coefficients (the blue colour in Figure 3) can be transferred directly to the lower DCT matrix; thus, this top-to-bottom alternating information can be applied to the attention mechanism by $\left\lfloor \frac{M}{2} \times \frac{M}{2} \right\rfloor$ splitting, where M denotes the current CU size for the next depth layer, implemented recursively to ensure that the number of CUs processed remains constant. To enhance convergence in (4), an advanced normalisation function $L2(\sqrt{\exp(\cdot)})$ is utilised [45,46], which provides faster convergence and maintains performance comparable to Softmax normalisation. This work also introduces a trainable procedure for adaptively adapting label information to predict upcoming states h_i , as follows:

$$\mathbf{C} = score_i \times \text{Softplus}(\sigma(\text{Linear}(c_t)) \odot \tanh(\text{Linear}(c_t))) \quad (5)$$

As mentioned in Section 4.2, c_t represents the complexity of the colour feature. However, it may have a negative value during the experiment, which can become an exception that is detrimental to the result determination. To overcome this, we optimise the final output feature by applying the σ and $\tanh$ activation functions with the aim of projecting c_t into the ranges $(\pm 0, +1)$ and $(-1, +1)$, respectively. The first projection scales the range to a normalisation range that is good for parameter training during backpropagation. Next, the second projection can discard the negative value while training the unweighted feature. Therefore, a combination of $\odot$ is presented before the Softplus activation function to prevent the neural nodes from becoming dead neurons before prediction. And the $score_i$ determines how the relation to the previous or next c_i should be considered and weights the current feature against the entire set of $\mathbf{C}$. Finally, with a Feed-Forward Network FFN with Softmax, the probability output can be obtained as follows:

$$\text{Prob}(\text{Split}, \neg \text{Split}) = \text{Softmax}(\text{FFN}(\mathbf{C})) \quad (6)$$

For better convergence and evaluation, each CU determines its complexity based on the texture it has learned. The final predicted CU is the average of the predictions of all individual decision trees; this part makes use of the results of the decision tree in the VVC standard. In the step of measuring the accuracy of the model on the test set, the proposed model uses the Root Mean Square Error (RMSE) as a loss function for the backpropagation of the training parameters, and also uses recall to compare the predicted CU with the attention scores. These metrics assess the consistency between the model's prediction and the perceived CU of the predicted feature. The advantage of this approach is that it not only achieves the highest accuracy, but also reveals which features are influential in predicting CU outcomes. This provides valuable insights into the user experience and potential areas for future improvement.

5. Experimental Results

This section discusses the dataset selected for the study and provides a detailed description of the configurations and environments used in the development process. It reports and analyses the experimental results, including comparisons with existing approaches in such areas. To focus on intra-prediction, the training process used a series of images as datasets instead of video-related datasets: Uncompressed Colour Image Database (UCID) [47] and DIV2K [48]. This selection improved the network's intra-prediction optimisation without additional coding procedures, reducing memory requirements and training time. Once the model was well trained, it could be interpreted in VTM-16.0 to complete the encoding process in accordance with the CTC.

5.1. Dataset Preparation and Training Strategy

To implement this work, we made use of Pytorch [49] to achieve the neural network, training on four NVIDIA Quadro RTX A4000 datasets with 16.0 GB of memory per GPU, for a total of 64.0 GB of device memory. Note that VTM-16.0 is modified to compress 40 DIV2K images and 120 UCID images and decode the encoded data by the official decoder. It contains about 32K 64×64 CUs as training data and about 1.6K CUs as validation data, both randomly selected for different QPs, and no overlap between the training and validation data, demonstrating the generalisability of our method. Moreover, the proposed model employs an advanced training strategy to improve the performance of our proposed model. This strategy involves the incorporation of a learning rate scheduler that dynamically adjusts the learning rate during the training process to achieve adaptive reduction. We trained our model with a batch size of 100. The learning rate was initially set to 1×10^{-3} and the training process continued until the learning rate reached 1×10^{-8} , as determined by the scheduler. To alleviate divergence problems in the early stages of training, a warm-up update function was introduced to ensure smoother convergence during optimisation. For faster training, we used half-precision floating-point and Distributed Data Parallel (DDP) to reduce memory consumption and speed up computation, and we also employed the optimiser of Adam [50] to facilitate efficient parameter updates.

In order to present a competitive performance, we present the test results according to the CTC that each selected test sequence(s) was preformed under in the All-Intra configuration. Table 3 details the properties of the test sequences we selected; note that Class F is excluded because it contains differential resolution and consists of a screen frame that is not captured by the camera. In practice, the experiment kept using $QP \in \{22, 27, 32, 37\}$ to ensure the credibility of these experimental results, reporting the result from the first 30 frames of each sequence.

Table 3. Selected test sequences and their properties for each class.

Class	Sequence Name	Resolution	Frame Count	Frame Rate	Bit Depth
A1	<i>Tango2</i>	3840 × 2160	294	60	10
	<i>FoodMarket4</i>	3840 × 2160	300	60	10
	<i>Campfire</i>	3840 × 2160	300	30	10
A2	<i>CatRobot</i>	3840 × 2160	300	60	10
	<i>DaylightRoad2</i>	3840 × 2160	300	60	10
	<i>ParkRunning3</i>	3840 × 2160	300	50	10
B	<i>Cactus</i>	1920 × 1080	500	50	8
	<i>BasketballDrive</i>	1920 × 1080	500	50	8
	<i>BQTerrace</i>	1920 × 1080	600	60	8
C	<i>RaceHorsesC</i>	832 × 480	300	30	8
	<i>BQMall</i>	832 × 480	600	60	8
	<i>PartyScene</i>	832 × 480	500	50	8
D	<i>RaceHorses</i>	416 × 240	300	30	8
	<i>BQSquare</i>	416 × 240	600	60	8
	<i>BlowingBubbles</i>	416 × 240	500	50	8
E	<i>FourPeople</i>	1280 × 720	600	60	8
	<i>Johnny</i>	1280 × 720	600	60	8
	<i>KristenAndSara</i>	1280 × 720	600	60	8
F	<i>BasketballDrillText</i>	832 × 480	500	50	8
	<i>SlideEditing</i>	1280 × 720	300	30	8
	<i>SlideShow</i>	1280 × 720	500	20	8

5.2. Results Analysis and Discussion

The evaluation of fast algorithms depends on their capability to reduce coding time while preserving frame quality under the All-Intra configuration. Therefore, we focus on three parameters from our experimental results and use Bjøntegaard Delta Bit Rate (BDBR) [51] as the performance metric, indicating time savings with respect to bitrate changes. The time savings achieved by the proposed approach to reduce encoding complexity are demonstrated against the VTM-16.0 reference. Additionally, more advanced approaches are also reported for comparative analysis with CNNAC [52] and Deep-QTMT [53,54], respectively, implemented by HM-12.0, VTM-7.0, and VTM-10.0. Specifically, the encoder is configured for a 10-bit depth, temporal subsampling is set to encode at eight-bit colour, and the Rate Distortion Optimised Quantisation (RDOQ) feature is disabled to highlight the gain from the *QP*. This setting allows colour and depth bits to improve compression efficiency and ensure high video quality even in a full scene under intra-prediction.

Table 4 presents a comparison of the coding performance of the proposed method with other current state-of-the-art methods. To ensure a fair comparison, the method configured in these experiments includes only the fast CU decision method and we disable the RDOQ option and the early merge mode decision method in HM and VTM.

Table 4. Performance comparison of different works with the proposed methods.

Class	Sequence Name	CNNAC [52]		DeepQTMT [53]		[54]		Proposed	
		BDBR (%)	ΔTime (%)	BDBR (%)	ΔTime (%)	BDBR (%)	ΔTime (%)	BDBR (%)	ΔTime (%)
A1	<i>Tango2</i>	-	-	3.10	−51.72	-	-	1.61	−57.79
	<i>FoodMarket4</i>	-	-	3.40	−53.15	-	-	1.72	−55.42
	<i>Campfire</i>	-	-	2.91	−59.87	1.02	−34.80	1.71	−59.32
A2	<i>CatRobot</i>	-	-	3.28	−55.99	1.06	−38.71	1.63	−58.87
	<i>DaylightRoad2</i>	-	-	2.11	−49.81	-	-	1.32	−57.96
	<i>ParkRunning3</i>	-	-	2.25	−61.10	-	-	1.13	−56.52

Table 4. Cont.

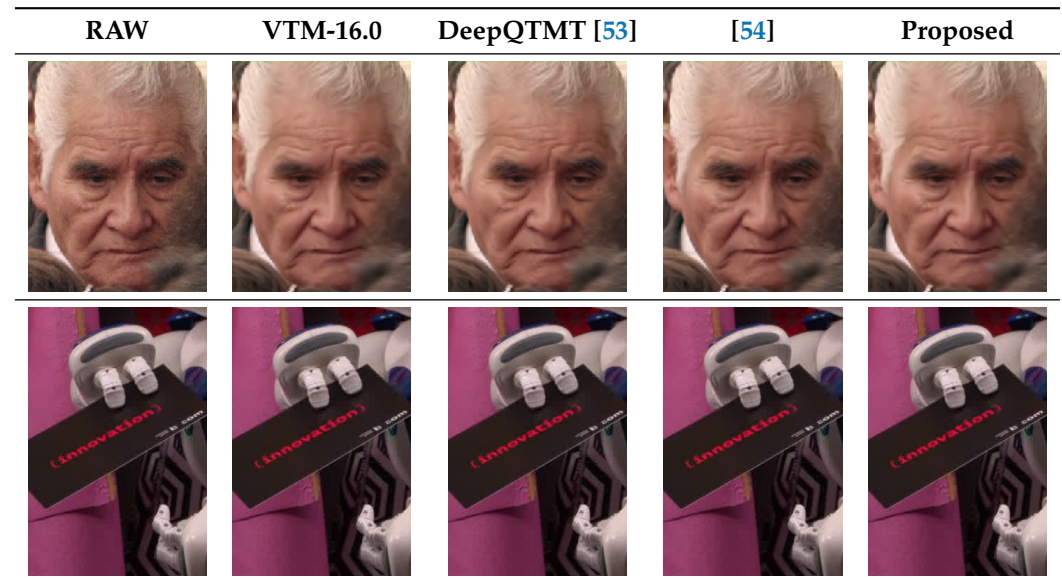
Class	Sequence Name	CNNAC [52]		DeepQTMT [53]		[54]		Proposed	
		BDBR (%)	Δ Time (%)	BDBR (%)	Δ Time (%)	BDBR (%)	Δ Time (%)	BDBR (%)	Δ Time (%)
B	<i>Cactus</i>	1.83	−45.98	1.86	−60.56	0.97	−33.89	1.58	−55.89
	<i>BasketballDrive</i>	1.92	−44.45	-	-	-	-	1.34	−57.20
	<i>BQTerrace</i>	1.71	−45.30	1.79	−56.94	0.97	−33.89	1.35	−57.87
C	<i>RaceHorsesC</i>	1.20	−46.11	1.61	−57.89	0.89	−37.69	1.62	−52.01
	<i>BQMall</i>	1.26	−42.14	-	-	-	-	1.05	−50.37
	<i>PartyScene</i>	1.13	−43.10	1.16	−58.94	1.16	−34.83	1.70	−50.99
D	<i>RaceHorses</i>	1.16	−43.46	1.88	−53.34	1.34	−36.03	1.33	−44.36
	<i>BQSquare</i>	1.07	−44.48	1.33	−55.16	0.94	−38.93	1.76	−48.61
	<i>BlowingBubbles</i>	1.18	−41.02	1.57	−53.40	1.08	−34.75	1.15	−48.07
E	<i>FourPeople</i>	2.22	−45.10	2.20	−59.74	1.05	−37.66	1.72	−46.87
	<i>Johnny</i>	2.10	−42.58	2.18	−57.29	-	-	1.94	−44.46
	<i>KristenAndSara</i>	2.16	−48.75	2.75	−60.01	0.97	−37.62	1.42	−47.30
Average		1.58	−44.37	2.21	−56.56	1.04	−36.25	1.48	−52.77

5.3. Discussion

This demonstration details that the proposed algorithm is both versatile and effective at reducing coding complexity for CU partition. The proposed algorithm can completely replace the complex RDO process and finalise the partitioning decisions of the CUs, providing robust performance in all sequences. The results of the partitioning predictions are very close to those of the VVC reference. The results indicate that the method described in [52] reduces coding time by approximately 44.37% and saves an average of 1.58% of BDBR. Similarly, the method in [53] reduces encoding time by an average of 56.56% but increases BDBR by an average of 2.21%. Moreover, [54] reports a reduction in encoding time of 36.25% and an average saving of 1.04% of BDBR. The results of this work show that encoding time can be reduced by an average of 52.77% and BDBR can be increased by 1.48%. It is worth noting that the higher resolutions (Classes A1, A2, and B) show a more significant improvement, with an average reduction in encoding time of 57.4% and an increase in BDBR of 1.49%. The most significant reduction in encoding time was observed for the sequence *Campfire*, with a reduction of 59.32% and an increase in BDBR of 1.71%. The sequence *ParkRunning3* offers the least time saving in encoding, with a 59.52% reduction in the encoding time and a 1.13% increase in BDBR. Our proposed method is cost-effective compared to other similar work, as it reduces encoding time by 1.0% while using only 0.02% of the BDBR. As a result, the algorithm introduced in this study can significantly reduce the coding time while maintaining coding quality. This highlights the potential of the algorithm to improve computational efficiency in relevant applications.

To evaluate the quality loss of the proposed method objectively and subjectively, this work mainly studies the complexity of prediction in VVC intra-frames. Thus, we compare the first frames coded in the All-Intra condition of the *Tango2* and *CatRobot* sequences with $QP = 32$ and intense motion under complex backgrounds. We have provide some selected regions for subjective quality comparisons, as illustrated in Table 5, to facilitate the observation of image loss details. It should be noted that the resolution of the selected frames varies, resulting in different sizes of the detailed CU, which is marked as a 64×64 size in our target. However, the loss is imperceptible to the human eye and can be disregarded. Similar results were observed in the other test sequences from the CTCs. Therefore, the method proposed in this work performs well in terms of both image quality and coding efficiency.

Table 5. Examples of video quality comparisons between various approaches. They were all captured from the first decoded frame of *Tango2* and *CatRobot* with $QP = 32$.



6. Conclusions

This paper proposes a fast intra-frame VVC algorithm to reduce the coding complexity and decrease the bitrate. We present a decision partitioning algorithm and a machine learning-based CU pattern prediction algorithm to extract frequency features from the DCT matrix by using a neural network and an attention mechanism to discriminate the texture complexity. To improve accuracy, we use multiple classifiers to analyse the layers of the serial network separately in a hierarchical fashion, reducing the pressure of relying on a single classifier. For decision-making, it provides fast convergent pattern selection by weighting the extracted features and using the proposed normalisation function $L2(\sqrt{\exp(\cdot)})$ after combining the weighted features. Compared to VTM-16.0, the proposed algorithm improves the average coding time by 52.77% with only a 1.48% increase in the BD BR. The experimental results confirm the effectiveness of the proposed algorithm and future research should prioritise the reduction of performance loss.

Author Contributions: Conceptualisation, K.-H.C. and S.-K.I.; Methodology, K.-H.C. and S.-K.I.; Software, K.-H.C.; Validation, S.-K.I.; Formal analysis, K.-H.C. and S.-K.I.; Investigation, K.-H.C.; Resources, K.-H.C. and S.-K.I.; Data curation, K.-H.C. and S.-K.I.; Writing—original draft preparation, K.-H.C.; Writing—review and editing, S.-K.I.; Visualisation, K.-H.C.; Supervision, S.-K.I.; Project administration, S.-K.I.; Funding acquisition, S.-K.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Macao Polytechnic University (Research Project RP/FCA-06/2023).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bross, B.; Wang, Y.K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; Ohm, J.R. Overview of the Versatile Video Coding (VVC) Standard and its Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3736–3764. [\[CrossRef\]](#)
2. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [\[CrossRef\]](#)
3. Nguyen, T.; Xu, X.; Henry, F.; Liao, R.L.; Sarwer, M.G.; Karczewicz, M.; Chao, Y.H.; Xu, J.; Liu, S.; Marpe, D.; et al. Overview of the Screen Content Support in VVC: Applications, Coding Tools, and Performance. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3801–3817. [\[CrossRef\]](#)

4. Schwarz, H.; Coban, M.; Karczewicz, M.; Chuang, T.D.; Bossen, F.; Alshin, A.; Lainema, J.; Helmrich, C.R.; Wiegand, T. Quantization and Entropy Coding in the Versatile Video Coding (VVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3891–3906. [\[CrossRef\]](#)
5. Bossen, F.; Suhring, K.; Wiecekowsky, A.; Liu, S. VVC Complexity and Software Implementation Analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3765–3778. [\[CrossRef\]](#)
6. Zouidi, N.; Kessentini, A.; Hamidouche, W.; Masmoudi, N.; Menard, D. Complexity assessment of the intra prediction in Versatile Video Coding. *Multimed. Tools Appl.* **2023**, *82*, 27751–27770. [\[CrossRef\]](#)
7. Shu, Z.; Li, J.; Peng, Z.; Chen, F.; Yu, M. Intra Complexity Control Algorithm for VVC. *Electronics* **2022**, *11*, 2572. [\[CrossRef\]](#)
8. Barannik, V.; Sidchenko, S.; Barannik, N.; Barannik, V. Development of the method for encoding service data in cryptocompression image representation systems. *East. Eur. J. Enterp. Technol.* **2021**, *3*, 103–115. [\[CrossRef\]](#)
9. Bross, B.; Chen, J.; Ohm, J.R.; Sullivan, G.J.; Wang, Y.K. Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC). *Proc. IEEE* **2021**, *109*, 1463–1493. [\[CrossRef\]](#)
10. Hamidouche, W.; Pescador, F.; Biatek, T.; Francois, E. Editorial Real-Time Implementation of VVC Standard for Consumer Electronic Devices. *IEEE Trans. Consum. Electron.* **2022**, *68*, 93–95. [\[CrossRef\]](#)
11. Sun, X.; Qiu, J.; Ma, Y.; Tao, Y.; Zhao, J.; Dong, Z. Encryption-based Coordinated Volt/Var Control for Distribution Networks With Multi-Microgrids. *IEEE Trans. Power Syst.* **2023**, *38*, 5909–5921. [\[CrossRef\]](#)
12. Im, S.; Chan, K. Higher precision range estimation for context-based adaptive binary arithmetic coding. *IET Image Process.* **2020**, *14*, 125–131. [\[CrossRef\]](#)
13. Liu, F.; Chen, Z. Multi-Objective Optimization of Quality in VVC Rate Control for Low-Delay Video Coding. *IEEE Trans. Image Process.* **2021**, *30*, 4706–4718. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Im, S.; Chan, K. A propagation model for package loss refinement in VVC. *Electron. Lett.* **2022**, *58*, 759–761. [\[CrossRef\]](#)
15. Dong, X.; Shen, L.; Yu, M.; Yang, H. Fast Intra Mode Decision Algorithm for Versatile Video Coding. *IEEE Trans. Multimed.* **2022**, *24*, 400–414. [\[CrossRef\]](#)
16. Saldanha, M.; Sanchez, G.; Marcon, C.; Agostini, L. Configurable Fast Block Partitioning for VVC Intra Coding Using Light Gradient Boosting Machine. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 3947–3960. [\[CrossRef\]](#)
17. Feng, A.; Liu, K.; Liu, D.; Li, L.; Wu, F. Partition Map Prediction for Fast Block Partitioning in VVC Intra-Frame Coding. *IEEE Trans. Image Process.* **2023**, *32*, 2237–2251. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Amna, M.; Imen, W.; Ezahra, S.F.; Mohamed, A. Fast intra-coding unit partition decision in H.266/FVC based on deep learning. *J. Real-Time Image Process.* **2020**, *17*, 1971–1981. [\[CrossRef\]](#)
19. Liu, W.; Li, Y.; Nie, M. Multitask Learning-Based Early MTT Partition Decision for Versatile Video Coding. In *Lecture Notes in Computer Science*; Springer Nature: Singapore, 2024; pp. 488–499. [\[CrossRef\]](#)
20. Chen, J.; Sun, H.; Katto, J.; Zeng, X.; Fan, Y. Fast QTMT Partition Decision Algorithm in VVC Intra Coding based on Variance and Gradient. In Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, 1–4 December 2019. [\[CrossRef\]](#)
21. Ding, G.; Lin, X.; Wang, J.; Ding, D. Accelerating QTMT-based CU partition and intra mode decision for versatile video coding. *J. Vis. Commun. Image Represent.* **2023**, *94*, 103832. [\[CrossRef\]](#)
22. Jing, Z.; Li, P.; Zhao, J.; Zhang, Q. A Fast CU Partition Algorithm Based on Gradient Structural Similarity and Texture Features. *Symmetry* **2022**, *14*, 2644. [\[CrossRef\]](#)
23. Huang, Y.; Xu, J.; Zhu, C.; Song, L.; Zhang, W. Precise Encoding Complexity Control for Versatile Video Coding. *IEEE Trans. Broadcast.* **2023**, *69*, 33–48. [\[CrossRef\]](#)
24. Belememis, P.; Panagou, N.; Loukopoulou, T.; Koziri, M. Review and comparative analysis of parallel video encoding techniques for VVC. In *Proceedings of the Applications of Digital Image Processing XLIII*; Tescher, A.G., Ebrahimi, T., Eds.; SPIE: Bellingham, WA, USA, 2020. [\[CrossRef\]](#)
25. Si, L.; Zhu, W.; Zhang, Q. Fast Adaptive CU Partition Decision Algorithm for VVC Intra Coding. *IEEE Access* **2023**, *11*, 119766–119778. [\[CrossRef\]](#)
26. Bouaafia, S.; Khemiri, R.; Sayadi, F.E.; Atri, M. Fast CU partition-based machine learning approach for reducing HEVC complexity. *J. Real-Time Image Process.* **2019**, *17*, 185–196. [\[CrossRef\]](#)
27. Wang, Y.; Liu, Y.; Zhao, J.; Zhang, Q. A Low-Complexity Fast CU Partitioning Decision Method Based on Texture Features and Decision Trees. *Electronics* **2023**, *12*, 3314. [\[CrossRef\]](#)
28. Cui, J.; Zhang, T.; Gu, C.; Zhang, X.; Ma, S. Gradient-Based Early Termination of CU Partition in VVC Intra Coding. In Proceedings of the 2020 Data Compression Conference (DCC), Snowbird, UT, USA, 24–27 March 2020. [\[CrossRef\]](#)
29. Viitanen, M.; Sainio, J.; Mercat, A.; Lemmetti, A.; Vanne, J. From HEVC to VVC: The First Development Steps of a Practical Intra Video Encoder. *IEEE Trans. Consum. Electron.* **2022**, *68*, 139–148. [\[CrossRef\]](#)
30. Kuo, Y.T.; Chen, P.Y.; Lin, H.C. A Spatiotemporal Content-Based CU Size Decision Algorithm for HEVC. *IEEE Trans. Broadcast.* **2020**, *66*, 100–112. [\[CrossRef\]](#)
31. Zhang, Q.; Zhao, Y.; Jiang, B.; Wu, Q. Fast CU Partition Decision Method Based on Bayes and Improved De-Blocking Filter for H.266/VVC. *IEEE Access* **2021**, *9*, 70382–70391. [\[CrossRef\]](#)
32. García-Lucas, D.; Cebrián-Márquez, G.; Díaz-Honrubia, A.; Mallikarachchi, T.; Cuenca, P. A fast full partitioning algorithm for HEVC-to-VVC video transcoding using Bayesian classifiers. *J. Vis. Commun. Image Represent.* **2023**, *94*, 103829. [\[CrossRef\]](#)

33. Park, J.; Lee, J.; Kim, B.; Jeon, B. Learning-Based Early Transform Skip Mode Decision for VVC Screen Content Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 6041–6056. [\[CrossRef\]](#)
34. Xing, H.; Tian, L.; Cao, M.; Li, C. A multi-embedding domain video steganography algorithm based on TU partitioning and intra prediction mode. *Neurocomputing* **2024**, *575*, 127247. [\[CrossRef\]](#)
35. Shang, X.; Li, G.; Zhao, X.; Han, H.; Zuo, Y. Fast CU size decision algorithm for VVC intra coding. *Multimed. Tools Appl.* **2023**, *82*, 28301–28322. [\[CrossRef\]](#)
36. Teng, G.; Xiong, D.; Ma, R.; An, P. Decision tree accelerated CTU partition algorithm for intra prediction in versatile video coding. *PLoS ONE* **2021**, *16*, e0258890. [\[CrossRef\]](#)
37. Wieckowski, A.; Brandenburg, J.; Bross, B.; Marpe, D. VVC Search Space Analysis Including an Open, Optimized Implementation. *IEEE Trans. Consum. Electron.* **2022**, *68*, 127–138. [\[CrossRef\]](#)
38. Im, S.; Chan, K. Dynamic estimator selection for double-bit-range estimation in VVC CABAC entropy coding. *IET Image Process.* **2024**, *18*, 722–730. [\[CrossRef\]](#)
39. Tissier, A.; Hamidouche, W.; Mdalsi, S.B.D.; Vanne, J.; Galpin, F.; Menard, D. Machine Learning Based Efficient QT-MTT Partitioning Scheme for VVC Intra Encoders. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 4279–4293. [\[CrossRef\]](#)
40. Wang, M.; Zhang, J.; Huang, L.; Xiong, J. Machine Learning-Based Rate Distortion Modeling for VVC/H.266 Intra-Frame. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021. [\[CrossRef\]](#)
41. Park, J.; Kim, B.; Lee, J.; Jeon, B. Machine Learning-Based Early Skip Decision for Intra Subpartition Prediction in VVC. *IEEE Access* **2022**, *10*, 111052–111065. [\[CrossRef\]](#)
42. Chan, K.H.; Im, S.K.; Ke, W. Multiple classifier for concatenate-designed neural network. *Neural Comput. Appl.* **2021**, *34*, 1359–1372. [\[CrossRef\]](#)
43. Chan, K.H.; Im, S.K.; Ke, W. VGGreNet: A Light-Weight VGGNet with Reused Convolutional Set. In Proceedings of the 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC), Leicester, UK, 7–10 December 2020. [\[CrossRef\]](#)
44. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
45. Im, S.; Chan, K. Vector quantization using k-means clustering neural network. *Electron. Lett.* **2023**, *59*, e12758. [\[CrossRef\]](#)
46. Im, S.K.; Chan, K.H. Context-Adaptive-Based Image Captioning by Bi-CARU. *IEEE Access* **2023**, *11*, 84934–84943. [\[CrossRef\]](#)
47. Schaefer, G.; Stich, M. UCID: An uncompressed color image database. In *Proceedings of the SPIE Proceedings*; Yeung, M.M., Lienhart, R.W., Li, C.S., Eds.; SPIE: Bellingham, WA, USA, 2003. [\[CrossRef\]](#)
48. Timofte, R.; Gu, S.; Wu, J.; Van Gool, L.; Zhang, L.; Yang, M.H.; Haris, M.; Shakhnarovich, G.; Ukita, N.; Hu, S.; et al. NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018. [\[CrossRef\]](#)
49. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2019; pp. 8024–8035.
50. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980. [\[CrossRef\]](#)
51. Bjøntegaard, G. Calculation of average PSNR differences between RD-curves. *VCEG-M33* **2001**.
52. Ma, C.; Liu, D.; Peng, X.; Wu, F. Convolutional Neural Network-Based Arithmetic Coding of DC Coefficients for HEVC Intra Coding. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018. [\[CrossRef\]](#)
53. Li, T.; Xu, M.; Tang, R.; Chen, Y.; Xing, Q. DeepQTMT: A Deep Learning Approach for Fast QTMT-Based CU Partition of Intra-Mode VVC. *IEEE Trans. Image Process.* **2021**, *30*, 5377–5390. [\[CrossRef\]](#) [\[PubMed\]](#)
54. Jing, Z.; Zhu, W.; Zhang, Q. A Fast VVC Intra Prediction Based on Gradient Analysis and Multi-Feature Fusion CNN. *Electronics* **2023**, *12*, 1963. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.