RANDOM FOREST ORIENTED FAST QTBT FRAME PARTITIONING

Thomas AMESTOY[⋆,†], *Alexandre MERCAT*[‡], *Wassim HAMIDOUCHE*[⋆], *Cyril BERGERON*[†] *and Daniel MENARD*[⋆]

[⋆] Univ Rennes, INSA Rennes, CNRS, IETR - UMR 6164, Rennes, France. Emails: firstname.lastname@insa-rennes.fr
[†] Thales SIX GTS France, HTE/STR/MMP Gennevilliers, France. Emails: firstname.lastname@thalesgroup.com
[‡] Tampere University, Korkeakoulunkatu 10, Tampere, 33720 Finland. Emails: firstname.lastname@tuni.fi

ABSTRACT

Block partition structure is a critical module in video coding scheme to achieve significant gap of compression performance. Under the exploration of future video coding standard by the Joint Video Exploration Team (JVET), named Versatile Video Coding (VVC), a new Quad Tree Binary Tree (QTBT) block partition structure has been introduced. In addition to the QT block partitioning defined by High Efficiency Video Coding (HEVC) standard, new horizontal and vertical BT partitions are enabled, which drastically increases the encoding time compared to HEVC. In this paper, we propose a fast QTBT partitioning scheme based on a Machine Learning approach. Complementary to techniques proposed in literature to reduce the complexity of HEVC Quad Tree (QT) partitioning, the propose solution uses Random Forest classifiers to determine for each block which partition modes between QT and BT is more likely to be selected. Using uncertainty zones of classifier decisions, the proposed complexity reduction technique is able to reduce in average by 30% the encoding time of JEM-v7.0 software in Random Access configuration with only 0.57% Bjøntegaard Delta Rate (BD-BR) increase.

***Index Terms*—** Video Compression, VVC, QTBT, JEM, Complexity Reduction, Machine Learning, Random Forest

1. INTRODUCTION

The Joint Video Exploration Team (JVET) has been recently investigating several new coding solutions under the Joint Exploration Model (JEM) software [1, 2] to show the benefits of developing a new standard called Versatile Video Coding (VVC) with coding capability beyond High Efficiency Video Coding (HEVC) [3]. These new coding tools already increase the coding efficiency by up to 40% compared to HEVC [4]. However, bitrate savings come with a significant complexity increase of 10 times the HEVC encoding time in Inter coding configuration. This complexity increase may interfere with the deployment of VVC standard on embedded platforms and live applications.

At the encoder side, computationally expensive tools have been added especially for the partitioning scheme, i.e. choosing the appropriate encoding block size for each part of the image. In JEM each frame is split into equal size blocks, called Coding Tree Unit (CTU). Each CTU is then split recursively with Quad Tree Binary Tree (QTBT) partitioning [5]. **Fig. 1** illustrates the QTBT partition of a CTU, with Binary Tree (BT) partition modes in green and Quad Tree (QT) partition mode in red. Possible BT partition modes are Binary Tree Horizontal (BTH) and Binary Tree Vertical (BTV) that allows splitting a Coding Unit (CU) in two equal rectangles horizontally and vertically, respectively. When BT partition mode is used on a CU, QT partition mode is no longer allowed. In JVET Common Test Conditions (CTC) [6], only 3 successive BT partitions are allowed. Enabling BT partition modes improves compression efficiency at the cost of considerable increase of partitioning scheme complexity and, consequently, of the encoding time.

In HEVC, the most common way to reduce the computational complexity of the encoding process is to reduce the QT partitioning scheme by testing less partition configurations. These solutions leverage intermediate encoding information (RD cost, co-located partitions, etc.) [7, 8], texture characteristics [9, 10], motion divergence [11, 12], Machine Learning (ML) and Deep Learning methods [13–15]. More recently, some solutions have already investigated the complexity reduction of QTBT partitioning scheme. Authors in [16] and [17], use Convolution Neural Networks (CNNs) to predict a depth description of QTBT partition of the CTUs. Wang et al [18] use Motion Divergence Field and gradient of luminance samples to model the Rate Distorsion (RD) cost and then prune unlikely branches of the QTBT partitioning tree. Complexity reduction techniques in [16], [17] and [18] are able to reduce encoding time by 42%, 32% and 50% for 0.65%, 0.52% and 1.3% of Bjøntegaard Delta Rate (BD-BR) increase in average, respectively.

In this paper, we propose a ML solution based on Random Forest (RF) classifiers to speed up the QTBT partitioning scheme. RF classifiers are trained off-line for every CU size to avoid expensive exploration of the partition mode classified as unlikely. The classifier estimates the partition mode between QT and BT that should be tested. Therefore, the proposed solution is complementary with QT partitioning complexity reduction techniques proposed for HEVC [8] [19]. To limit the RD loss induced by misclassification, uncertainty zones are introduced for the classifier decisions. When classifier decision is included in the uncertainty zone, both QT and BT partition modes are tested. The proposed complexity reduction solution is able to reach on average 30% encoding time saving with only 0.57% BD-BR increase.

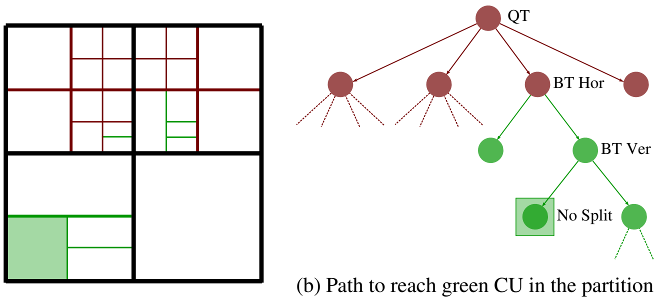
The rest of the paper is organized as follows. Section 2 goes through background of RF classifiers and presents the binary classification problem. Section 3 investigates the training dataset constitution. The training process to build RF classifiers is described in Section 4. Section 5 presents the experimental results. Finally, Section 6 concludes this paper.

2. RANDOM FOREST FOR QT VS BT CLASSIFICATION PROBLEM

2.1. Background on Random Forests

Classification by RF [20] is a classical method in ML. RF classifiers predict the value of a target variable, named class, from values of several input variables, named features. RFs methods bag many

This project has received funding from Bpifrance Financement under grant DOS0061463/00 (EFIGI FUI project).



(a) QTBT partition of a CTU.

(b) Path to reach green CU in the partition tree.

Fig. 1: Quad Tree Binary Tree in JEM. In red QT partition mode and in green BT modes.

single little-correlated decision trees and gather the results from all the trees to make a decision. Decision trees are constructed by a recursive partitioning of the data set into subsets called nodes. At each node, a threshold that achieves optimal separation of the classes is selected among one of the input features. In this work, the output of the RF classifier is the class receiving the most votes from the decision trees.

To de-correlate one tree of the RF to another, the nodes of the decision trees are created using random subsets of features. Moreover, each decision tree is trained on a random subset of the training dataset. By de-correlating trees, RF classifiers achieve a better trade-off between correct classification and training data over-fitting compared to a single decision tree classifier.

2.2. Binary Classification: QT vs BT Partition Mode

To find CTU partition that achieves the best RD performance, the encoder recursively explores all possible partition modes. This process is called full Rate Distorsion Optimization (RDO) search. For each CU, the encoder computes a RD cost of the whole CU. Then RD costs of QT and BT partition modes are computed, BT partition modes being composed of BTH and BTV partition modes. The encoder selects the partition mode that minimizes the RD cost.

To reduce the number of tested partition modes, a binary RF classifier (introduced in Section 2.1) is used to classify each CU in QT or BT partition mode. The proposed binary classifier, called $QTvsBT$, is trained off-line. When classification decision is QT partition mode, the RDO process is applied on QT partition mode and BT partition modes are ignored. When classification decision is BT partition modes, QT partition mode is ignored and RDO processes are applied on both BT partition modes.

3. TRAINING DATASET CREATION

Let a training instance be the entity composed of the chosen set of input features and the associated output class. This section details the steps of the training dataset creation, i.e. the set that contains all the training instances used to build the RF classifier.

3.1. Training Setup

The effectiveness of ML is highly linked to the diversity and relevance of the training dataset. To characterize a video content, Spatial Information (SI) and Temporal Information (TI) metrics are used [21]. The SI estimates the amount of spatial details while the TI measures the quantity of motion in the video sequence. To cover a wide range of content types, the training dataset is extracted from 10 training sequences spanning a large range of SI and TI space and distributed across 6 classes (A1, A2, B, C, D, E). The training sequences are included in JVET CTC [6]: *DaylightRoad* and *Campfire* (class A1), *Traffic* (class A2), *BasketballDrive* and *BQTerrace* (class B), *Flowervase* and *BQMall* (class C), *BQSquare* and *Keiba* (class D), *Johnny* (class E).

The training instances are extracted from training sequences encodings carried-out with the JEM-v7.0 in Random Access (RA) configuration at the 4 Quantization Parameter (QP) values considered in this work: 22, 27, 32 and 37. For the CUs, the corresponding output class is defined as the optimal partition mode selected after full RDO process. A separate training dataset is created for different CU sizes including 128×128, 64×64, 32×32 and 16×16.

Sequences with high resolution and high frame rate provide more CTUs to the training datasets compared to the low resolution and low frame rate sequences. To avoid being biased by these particular training sequences, the datasets used for training are composed of a fixed number of CTUs by training sequence. Furthermore, to reduce the problem of data imbalance, the training dataset for every CU size is composed randomly with the same amount of instances classified into QT output class and BT output class.

3.2. Features Evaluation and Selection

In our application, a feature is a property of the CU used to determine which partition mode between QT and BT should be selected. This subsection presents the evaluation and selection of features.

3.2.1. Evaluated Features

Based on related works, the features used by the RF classifiers are computed relying on texture of pixel luminance samples and motion divergence. In the following, motion divergence is considered through Motion Divergence Field (MDF), MDF being the array of Movement Vectors (MVs) of every 4x4 pixels blocks. The MVs point to the closest reference frame in term of temporal distance.

Evaluated features are divided into 3 categories: features computed on **whole CU**, features based on **sub-quarters of the CU** and features based on **inconsistency among CU sub-quarters**. Features computed on the **whole CU** are the following:

- *QP*: Quantization parameter used to encode the sequence.
- *VarPix*: Variance of luminance samples.
- *Grad*: Gradients in horizontal ($grad_x$) and vertical ($grad_y$) directions of the luminance samples (2 features).
- *RatioGrad*: ratio of gradients $\frac{grad_x}{grad_y}$.
- *VarMv*: $|\sigma^2_{\mathrm{MVx}} + \sigma^2_{\mathrm{MVy}}|$ with σ^2_{MVx} and σ^2_{MVy} respectively variances of horizontal and vertical MVs of MDF.
- *MaxDiffMv*: maximum 1-norm distance between MVs of MDF, noted mv, and their mean, noted $\overline{mv}$, as in Equation (1).

$$\begin{aligned}\text{MaxDiffMv} &= \max_{mv \in \text{MDF}} (||mv - \overline{mv}||_1) \\ &= \max_{mv \in \text{MDF}} (|mv_x - \overline{mv}_x| + |mv_y - \overline{mv}_y|)\end{aligned} \tag{1}$$

Features based on **sub-quarters of the CU** are the following:

- *QuarterVarPix*: VarPix on 4 sub-quarters (4 features).
- *QuarterVarMv*: VarMv on 4 sub-quarters (4 features).

- *QuarterMaxDiffMv*: MaxDiffMv on 4 sub-quarters (4 features).

For any feature f of current CU, f_1 is the feature computed on top-left sub-quarter, f_2 on top-right, f_3 on bottom-left and f_4 on bottom-right. Let $\delta H(f)$ and $\delta V(f)$ be Horizontal Inconsistency (HI) and Vertical Inconsistency (VI) as defined by Equation (2).

$$\begin{aligned} \delta H(f) &= |f_1 - f_2| + |f_3 - f_4| \\ \delta V(f) &= |f_1 - f_3| + |f_2 - f_4| \end{aligned} \quad (2)$$

The aim of HI and VI is to highlight which rectangular parts of the CU have the highest differences. HI is linked to BTH partition mode and VI to BTV partition mode. Features based on **inconsistency among sub-quarters** of the CU are the following:

- *InconsPix*: HI and VI of mean, variance and gradients-ratio of luminance samples (6 features).
- *InconsMv*: HI and VI of mean and variance of MDF (4 features).
- *DiffInconsPix*: difference between HI and VI for luminance based features (3 features).
- *DiffInconsMv*: difference between HI and VI for MDF based features (2 features).

3.2.2. Feature Selection

Mutual Information (MI) measures the decreasing of entropy $\mathbb{H}$ of a class C when the feature F is known. MI is expressed as $MI(F, C) = \mathbb{H}(C) - \mathbb{H}(C|F)$. For the current study, the feature evaluation is conducted with MI as metric. **Fig. 2** gives the MI of all evaluated features according to the CU size.

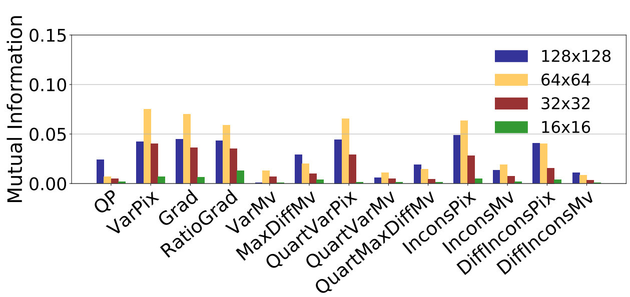


Fig. 2: Mutual Information of Features for different CU sizes.

Results show that features based on texture have higher MI than features based on MDF, independently of CU size. In other words, features based on texture are more relevant than features based on MDF to estimate which partition mode between QT and BT is to be tested, independently of CU size. Therefore, a unique set of features is selected for all CU sizes. The set of features is composed of the 19 features with the highest MI: *QP*, *VarPix*, *Grad* (2 features), *RatioGrad*, *MaxDiffMv*, *QuarterVarPix* (4 features), *InconsPix* (6 features) and *DiffInconsPix* (3 features).

4. CLASSIFIERS TRAINING PROCESS

The training process consists in building the classifier through maximizing the ratio of correct classification on the training dataset. In addition to correct classifications, the losses of RD performance introduced by misclassification are considered in this work.

4.1. Weight Training Instances with RD Cost Errors

To evaluate the impact of misclassification on the encoding efficiency, the RD errors ε_{RD} caused by a misclassification is introduced. RD error when the optimal partition mode A is chosen, while B is the partition mode selected by the encoder after a full RDO process, is defined by Equation (3).

$$\varepsilon_{RD}(A|B) = 100 \cdot \frac{RD_A - RD_B}{RD_B} \quad (3)$$

RD_A and RD_B are the RD costs resulting of full RDO process for partition modes A and B, respectively. In our case, $(A, B) \in \{(QT, BT), (BT, QT)\}$ and $RD_{BT} = min(RD_{BTH}, RD_{BTV})$.

Fig. 3 gives the average of RD error for the two types of misclassification, $\varepsilon_{RD}(QT|BT)$ and $\varepsilon_{RD}(BT|QT)$, according to the CU size. Results are averaged across 4 video sequences (*BasketballDrive*, *BQMall*, *Flowervase*, *Johnny*) and four QP.

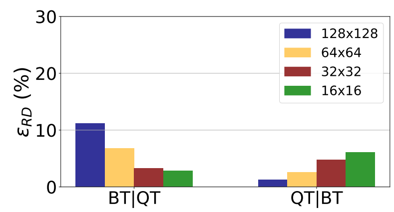


Fig. 3: Average RD misclassification error, by type of misclassification and by CU size.

Fig. 3 shows that $\varepsilon_{RD}(BT|QT)$ is higher for large CUs (128×128 and 64×64) than small CUs (32×32 and 16×16). In other words, $(BT|QT)$ misclassification has in average stronger impact on RD cost of large CUs compared to small CUs. Indeed, when BT partition mode is selected on large CUs, QT partition mode is no longer available, as detailed in Section 1. Combined with the limit of 3 successive BT partitions, fine grain partitioning is no longer achievable.

On the other hand, $\varepsilon_{RD}(QT|BT)$ is higher for small CUs than for large CUs. This is due to the fact that rectangular BT partition modes offer more partitioning shapes than square QT partition mode on small areas in the frame.

To conclude, the impact on RD cost according to misclassification type depends on CU size. From this observation, a ε_{RD} weighting of training instances is added in order to minimize the sum of ε_{RD} induced by misclassification. Note that the weights are needed only during the training process, and not when the trained model is used in encoding process.

4.2. Classification Rates

Let the classification rate be the percentage of correct classification given by the 4-fold cross-validation on the training dataset. **Table 1** describes the average classification rates of the $QTvsBT$ classifier, across the training sequences for the four considered QP and according to the CU size.

Table 1: Correct classification (in %), for every CU size category.

	128×128	64×64	32×32	16×16	Average
$QTvsBT$	69	70	67	60	67

The classification rates are between 60% and 70% according to the the CU size. In the literature, classification rates of techniques using ML to reduce the complexity of the QT partitioning in HEVC are close to 80% [23, 24].

In order to limit RD efficiency losses induced by misclassification, uncertainty zones of classification are introduced for each binary classifier. In the uncertainty zones both BT and QT partition modes are tested. A score value, deduced from the votes of individual decision trees, is used to build the uncertainty zones. The associated score $Score(A)$ corresponds to the percentage of decision trees that predict the class A and is defined as $Score(A) = \frac{N_{votes}(A)}{N_{trees}}$.
$N_{votes}(A)$ is the number of trees voting for class A and N_{trees} is the number of trees constituting the RF classifier. $Score(A)$ takes values between 0 and 1. The closer $Score(A)$ is to 1, the more predominantly the RF classifier selects class A.

For our specific case of classification between two classes QT and BT, as $N_{votes}(QT) + N_{votes}(BT) = N_{trees}$, then $Score(QT) + Score(BT) = 1$. Using this relation, the classification decision of $QTvsBT$ RF classifier is QT if $Score(QT) > 0.5$ and BT otherwise (see Section 2.1).

An example of uncertainty zone is illustrated in red on **Fig. 4**. Uncertainty zone is the range $[0.5 - dS(BT), 0.5 + dS(QT)]$ of $Score(QT)$, with $dS(BT)$ and $dS(QT)$ fixed thresholds included in $[0, 0.5]$. In uncertainty zone, both BT and QT partition modes are tested by the encoder.

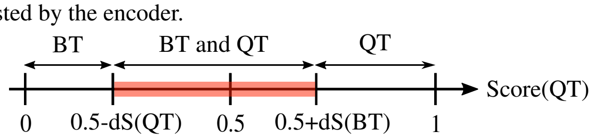


Fig. 4: Uncertainty Zone for $QTvsBT$ classification.

Based on an off-line study, $dS(QT)$ and $dS(BT)$ thresholds are fixed by resolution and CU size using training sequences to limit misclassification. Using uncertainty zones, 98% of training sequences CUs are either correctly classified or included in the uncertainty zone, limiting RD efficiency losses.

5. EXPERIMENTAL RESULTS

5.1. Experimental Setup

To conduct the experiments, 18 video sequences (3 by class) not included in the training sequences (see Section 3.1) are used: *CatRobot1*, *ParkRunning3*, *ToddlerFountain*, *Traffic*, *SteamLocomotiveTrain*, *NebutaFestival*, *Cactus*, *RitualDance*, *Kimono*, *RaceHorsesC*, *PartyScene*, *BasketballDrill*, *ParkScene*, *KristenAndSara*, *FourPeople*, *BlowingBubbles*, *RaceHorsesD* and *BQSquare*. The experiments are conducted under the CTC [6] for RA configuration at the four QP values. The proposed complexity reduction solution is implemented in JEM-v7.0. In order to limit the encoding time, JEM-v7.0 encoder compares the RD cost of the whole current CU with those of the BTH and BTV partition modes to prune the QT partition mode. As our solution does not compute all the RD costs of the BT partition mode, this condition is removed in the experiments. The performance of our complexity reduction solution is evaluated by measuring the trade-off between RD efficiency using the BD-BR [25] and encoding time reduction ΔT, defined by Equation 4.

$$\Delta T = \frac{1}{4} \sum_{QP_i \in \{22,27,32,37\}} \frac{T_O(QP_i) - T_R(QP_i)}{T_O(QP_i)} \quad (4)$$

$T_O(QP_i)$ and $T_R(QP_i)$ are the original (encoded with full RDO process) and reduced time spent to encode the video sequence with $QP = QP_i$, respectively.

5.2. Performance Evaluation of the Proposed Solution

Table 2 details the performance in terms of encoding degradation BD-BR(%), encoding time reduction $\Delta T(\%)$ and time overhead θ of the proposed complexity reduction solution with respect to the performance obtained by disabling the BT partitioning modes. The results are averaged by class across the 18 test sequences.

Table 2: BD-BR(%), $\Delta T(\%)$ and $\theta\%)$ averaged by video class.

	Proposed Solution			**Disabled BT partition modes**	
Video Class	BD-BR(%)	$\Delta T(\%)$	$\theta(\%)$	BD-BR(%)	$\Delta T(\%)$
A1	+ 0.85	30	0.07	+ 6.5	81
A2	+ 0.58	31	0.12	+ 3.5	80
B	+ 0.7	30	0.09	+ 6.9	80
C	+ 0.47	32	0.19	+ 4.5	79
D	+ 0.36	27	0.55	+ 3.3	75
E	+ 0.49	31	0.22	+ 3.6	78
Average	+ 0.57	30	0.21	+ 4.7	79

BT partition modes are added to QT partition mode defined in HEVC standard to improve compression efficiency at the cost of increased encoding time. Therefore, disabling BT partition modes sets an upper bound of complexity reduction. On the one hand, **Table 2** shows that the proposed solution is able to reduce the encoding time by 30% for a BD-BR increase of 0.57% in average. On the other hand, results show that disabling BT partition modes increase the BD-BR of 4.7% for a complexity reduction ΔT of 79% in average. Compared to disabling BT partitioning modes, the proposed solution is able to divide by 8 the BD-BR increase, dividing by 2.5 complexity reduction in average.

Related works [16], [17] and [18] presented in Section 1 reduce the encoding time of 42%, 32% and 50%, with BD-BR increase of 0.65%, 0.52% and 1.3%, respectively. However, [16, 17] techniques are based on CNNs to reduce the encoding time without always specifying their implementations, while CNNs are known to have high computational overhead. The complexity overhead θ of our proposed solution, including in the encoding time reduction, is between 0.07% and 0.55% of the encoding time. This performance confirms the lightweight of our approach and highlights that RFs classifiers consume few computing resources, which is a key point to use this solution in a real-time or embedded framework.

6. CONCLUSION

This paper presents a ML solution based on RF classifiers to speed up the QTBT partitioning scheme. RF classifiers are trained off-line to determine which partition modes between QT and BT is more likely. The features used by the RF classifiers are computed based on texture of pixel luminance samples and motion divergence. Experimental results have shown that the proposed solution reduces the encoding time by 30% on average for a BD-BR increase of 0.57%. The complexity overhead of our solution represents 0.21% of the encoding time in average, highlighting that RFs classifiers consume few computing resources. Future work will go further by adding classifiers to estimate which partition mode between BTH and BTV should be selected.

7. REFERENCES

[1] "N17055, Algorithm Description of Joint Exploration Test Model 7 (JEM 7)," vol. https://mpeg.chiariglione.org/standards/exploration/future-video-coding/n17055-algorithm-description-joint-exploration-test-model.

[2] "Joint Exploration Model (JEM) reference software 7.0," .

[3] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] Naty Sidaty, Wassim Hamidouche, Olivier Deforges, and Pierrick Philippe, "Emerging video coding performance: 4k quality monitoring," in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, Erfurt, Germany, May 2017, pp. 1–3, IEEE.

[5] J. An, Y.-W. Chen, K. Zhang, H. Huang, Y.-W. Huang, and S. Lei, "Block partitioning structure for next generation video coding," vol. https://www.itu.int/md/T13-SG16-C-0966/en, 2015.

[6] Jill Boyce, Karsten Suehring, and Xiang Li, "JVET-J1010: JVET common test conditions and software reference configurations," 2018.

[7] Zhaoqing Pan, Sam Kwong, Ming-Ting Sun, and Jianjun Lei, "Early MERGE Mode Decision Based on Motion Estimation and Hierarchical Depth Correlation for HEVC," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 405–412, June 2014.

[8] Guilherme Correa, Pedro Assuncao, Luciano Agostini, and Luis da Silva Cruz, "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1866–1874, Nov. 2011.

[9] Biao Min and Ray C. C. Cheung, "A Fast CU Size Decision Algorithm for the HEVC Intra Encoder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 892–896, May 2015.

[10] Alexandre Mercat, Florian Arrestier, Maxime Pelcat, Wassim Hamidouche, and Daniel Menard, "Prediction of quad-tree partitioning for budgeted energy HEVC encoding," in *2017 IEEE International Workshop on Signal Processing Systems (SiPS)*, Lorient, Oct. 2017, pp. 1–6, IEEE.

[11] Liquan Shen, Zhi Liu, Zhaoyang Zhang, and Xuli Shi, "Fast Inter Mode Decision Using Spatial Property of Motion Field," *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1208–1214, Oct. 2008.

[12] Jian Xiong, Hongliang Li, Qingbo Wu, and Fanman Meng, "A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 559–564, Feb. 2014.

[13] Zhenyu Liu, Xianyu Yu, Yuan Gao, Shaolin Chen, Xiangyang Ji, and Dongsheng Wang, "CU Partition Mode Decision for HEVC Hardwired Intra Encoder Using Convolution Neural Network," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5088–5103, Nov. 2016.

[14] Fanyi Duanmu, Zhan Ma, and Yao Wang, "Fast CU partition decision using machine learning for screen content compression," in *2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, QC, Canada, Sept. 2015, pp. 4972–4976, IEEE.

[15] Xiaolin Shen and Lu Yu, "CU splitting early termination based on weighted SVM," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, Dec. 2013.

[16] Zhipeng Jin, Ping An, Liquan Shen, and Chao Yang, "CNN oriented fast QTBT partition algorithm for JVET intra coding," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, St. Petersburg, FL, Dec. 2017, pp. 1–4, IEEE.

[17] Zhao Wang, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, and Siwei Ma, "Fast QTBT Partitioning Decision for Interframe Coding with Convolution Neural Network," *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2550–2554, 2018.

[18] Zhao Wang, Shiqi Wang, Jian Zhang, Shanshe Wang, and Siwei Ma, "Probabilistic Decision Based Block Partitioning for Future Video Coding," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1475–1486, Mar. 2018.

[19] Alexandre Mercat, Florian Arrestier, Maxime Pelcat, Wassim Hamidouche, and Daniel Menard, "On predicting the HEVC intra quad-tree partitioning with tunable energy and rate-distortion," *Journal of Real-Time Image Processing*, July 2018.

[20] Leo Breiman, "Random Forests," Jan. 2001.

[21] ITU, "Recommandation ITU-T P.910 : Subjective video quality assessment methods for multimedia applications," 1999.

[22] Xiaolin Shen, Lu Yu, and Jie Chen, "Fast coding unit size selection for HEVC based on Bayesian decision rule," in *2012 Picture Coding Symposium*, Krakow, May 2012, pp. 453–456, IEEE.

[23] Alexandre Mercat, Florian Arrestier, Maxime Pelcat, Wassim Hamidouche, Daniel Menard, INSA Rennes, and Institut Pascal, "Machine Learning Based Choice of Characteristics for the One-Shot Determination of the HEVC Intra Coding Tree," *2018 Picture Coding Symposium (PCS)*, pp. 263–267.

[24] Guilherme Correa, Pedro A. Assuncao, Luciano Volcan Agostini, and Luis A. da Silva Cruz, "Fast HEVC Encoding Decisions Using Data Mining," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 660–673, Apr. 2015.

[25] Gisle Bjontegaard, "Calculation of average PSNR differences between RD-Curves," Apr. 2001, VCEG-M33 ITU-T.