Machine Learning Approaches for Intra-Prediction in HEVC

Buddhiprabha Erabadda, Thanuja Mallikarachchi, Gosala Kulupana and Anil Fernando
Centre for Vision, Speech and Signal Processing, University of Surrey, United Kingdom
Email: {e.harshani, d.mallikarachchi, g.kulupana, w.fernando}@surrey.ac.uk

***Abstract*—The use of machine learning techniques for encoding complexity reduction in recent video coding standards such as High Efficiency Video Coding (HEVC) has received prominent attention in the recent past. Yet, the dynamically changing nature of the video contents makes it evermore challenging to use rigid traditional inference models for predicting the encoding decisions for a given content. In this context, this paper investigates the resulting implications on the coding efficiency and the encoding complexity, when using offline trained and online trained machine-learning models for coding unit size selection in the HEVC intra-prediction. The experimental results demonstrate that the ground truth encoding statistics of the content being encoded, is crucial to the efficient encoding decision prediction when using machine learning based prediction models.**

***Index Terms*—HEVC, intra-prediction, machine learning, support vector machines, coding unit**

I. Introduction

The High Efficiency Video Coding (HEVC) standard, which was introduced in 2013, has shown a significant coding efficiency improvement (≈50%) compared to its immediate predecessor H.264/AVC [1]. This is attributed to the assortment of novel coding modes and features, and the quad-tree based hierarchical partitioning structure that encompass the new standard. However, the brute force approach of analyzing every possible combination of coding modes and partitioning structure using Rate-Distortion (RD) optimization in order to select the optimum coding modes for a given content, adversely affects the encoding complexity of HEVC encoders.

Therefore, the recent literature proposes numerous approaches to reduce the HEVC's compelling encoding complexity. These include knowledge based methods that utilize statistical [2], [3], texture and motion analysis information [4], [5], and learning based methods that utilize machine learning based inference models [6]–[8], to predict the coding information for a given content thereby skipping the brute-force RD optimization. However, recent advancements in data science and deep learning frameworks have popularized the use of machine learning based inference models to early predict the coding modes and coding structure for a given content and reduce the exorbitant complexity of the encoders.

That being said, the properties of the contents are subjected to continuous and dynamic changes, which make the offline trained machine learning based prediction models less flexible for complex video sequences. Therefore, it is vital to investigate the possibilities of extending the offline trained models to dynamically adapt with the changing video contents, such that coding decisions predicted from the models are relevant to the content being encoded. In this context, this paper investigates the impact on the coding efficiency and the coding complexity in the HEVC intra-prediction, when using offline trained and online trained Support Vector Machine (SVM) models for Coding Unit (CU) size selection.

The remainder of the paper is organized as follows. Sec. II provides a detailed overview of the state-of-the-art encoding complexity reduction methods that use machine learning techniques. Sec. III briefly describes the implication of using offline and online trained models for CU size selection. Sec. IV presents the experimental results followed by a discussion and finally, Sec. V states the concluding remarks and future work.

This work was supported by the CONTENT4ALL project, which is funded under European Commission's H2020 Framework Program (Grant number: 762021).

II. Background and related work

The complexity reduction using machine learning based models is achieved by intelligently predicting the optimal coding parameter combinations for a given content. This includes the prediction of coding parameters ranging from prediction mode, reference pictures, and quad-tree hierarchy to filtering parameters, etc. However, the selection of quad-tree hierarchy for a given content (in particular the CU size) is considered the major source of the increased encoding complexity in HEVC encoders. For example, the CU size decision (i.e., 8×8 to 64×64) in the quad-tree structure is traditionally decided at CU depth level (i.e., 0, 1, 2, 3), where a decision is taken whether to split or not split the current CU into sub-CUs based on the RD cost at each level [1]. Thus, the objective of using a prediction model is to skip the brute-force evaluation of all possible CU sizes and to predict the optimal CU size that minimizes the RD cost for a given content.

The following sub-sections introduce some of the state-of-the-art methods that fall under two major supervised machine learning techniques; Support Vector Machines (SVM) and Neural Networks (NN) that have been widely adopted in encoder complexity reduction.

A. SVM-based approaches

Zhang *et al.* [4] propose a two-stage SVM approach to determine the CU size. In the first stage, eight SVMs are used, with two SVMs for each of the four CU depth levels

(i.e., 0, 1, 2, 3). These SVMs are built offline using a large amount of training data. During the encoding process, one of these two SVMs takes the decision for *early split* (ES) and the other SVM takes the decision for *early terminate* (ET). The decision of the first stage is a combined outcome of these two SVMs. When the predictions of the two models disagree, the decision making is passed down to the second stage, where a single SVM is used at each depth level. The SVM classifiers of the second stage are online classifiers, that are built after encoding the current depth level. The training data for these models are calculated during the encoding process. Here, the binary SVM classifier at each depth level decides either to stop at the current CU depth level or to continue to evaluate the sub-CUs.

On the other hand, the fast CU size decision using algorithm proposed by Liu *et al.* [9] uses two SVMs (one for the split decision and one for the non-split decision) to categorize CUs in to 3 classes; CUs with high texture complexity (split), CUs with less texture complexity (do not split), and CUs with average texture complexity (difficult to predict). In this approach, the CUs that are categorized into latter are subjected to traditional RD optimization to determine their split decision.

Similar approaches that utilize SVMs to reduce the HEVC's encoding complexity in intra coding are reported in [6], [8], [10].

B. NN-based approaches

Liu *et at.* [11] propose a Convolutional Neural Network (CNN) based approach that predicts the CU split decision. The CNN architecture has two convolution layers and two fully-connected layers. The CU is classified into one of three categories; *homo* (terminate at the current depth level), *split* (skip the current depth level and evaluate the next depth level), and *comb* (evaluate with traditional RD optimization).

A similar approach has been adopted in [12] with a CNN architecture that constitutes three convolution layers and three fully-connected layers. Here, one CNN per each CU depth level has been used, as opposed to using a single CNN for all CU depth levels.

The algorithms presented in [13], [14] also use similar CNN architectures for predicting the CU size for a given image block, thereby reducing the encoding complexity in HEVC intra-prediction.

C. Other machine learning based approaches

In addition to the widely adopted SVM and NN based approaches, several other machine learning techniques have been utilized to develop fast encoding algorithms for HEVC. For example, logistic regression [15], decision trees [16], [17], random forest [18], and Bayesian classification [19] are some of the state-of-the-art learning based approaches that have been considered in the literature for reducing HEVC's encoding complexity.

III. Evaluating the implications of offline and online trained models

In order to investigate the impact on the coding efficiency and encoding complexity, when using offline and online trained SVM models for CU size selection, we utilize two different encoding algorithms; one that predicts the CU sizes based on decisions of offline trained models and another that predicts CU size decisions using online trained models which are created at runtime during the encoding process.

The CU split decision can be modeled as a binary classification problem, with classes *split* and *non-split*. Here we use SVMs as the machine learning technique because it can handle binary classification with significant computational advantages [20]. Furthermore, SVMs are widely been used in the context of CU split decision prediction, thus, the proposed analysis will be beneficial for a large number of algorithms. However, we also believe that similar implications are expected for any other machine learning technique since the data collection and training will become common factors regardless of the learning technique been adopted.

SVMs can utilize a range of features available in the encoding loop as input features which are mapped to the output that decides whether the CU is split or not. In this case, the feature set proposed in [4] has been adopted for both offline and online trained SVM models which have been utilized for the following evaluation. The CU level feature set proposed in [4] constitute context features from the neighbouring CUs, CU complexity information and coding information for the current CU depth level, thus, covers every aspect of the CU.

A. Online-only training

1) Data collection: In the case of online training, the data set for SVM training is accrued from the content being encoded, during the encoding process. For instance, the video sequence is initially encoded using RD optimization until 2000 data samples are collected for each CU depth level. The data constitutes the feature vector described in (1) and CU split decision obtained from the RD optimization. Once the number of expected data points are gathered, two models (*split* and *non-split*) are created for each depth level.

The model pair, *split* and *non-split* for a given CU depth level, is independent from the models in other depth levels. Therefore, data collection takes place independently resulting in the models being created at different points in time, depending on the video resolution and Quantization Parameter (QP). For example, when encoding a low resolution video (e.g., 416×240), models for lower CU depths (i.e., 0, 1) are likely to be generated after encoding several frames. On the contrary, models for higher CU depths (i.e., 2, 3) may become available much sooner.

Each data sample includes a set of features F^i defined as,

$$F^i := \{\theta^i, \pi^i, \tau^i\}, \tag{1}$$

where θ^i, π^i, and τ^i refer to texture, pre-analysis (encoding with PLANAR mode), and context information, respectively. The features collected during this process are described in detail in [4]. Here, $\mathrm{i} \in \{0, 1, 2, 3\}$ denotes the CU depth level.

2) Weight calculation: The optimal weights for the weighted SVMs are determined during runtime for each depth level independently. This is done by testing a range of weight value pairs ranging from 1:5 and 5:1, for the two SVMs (split

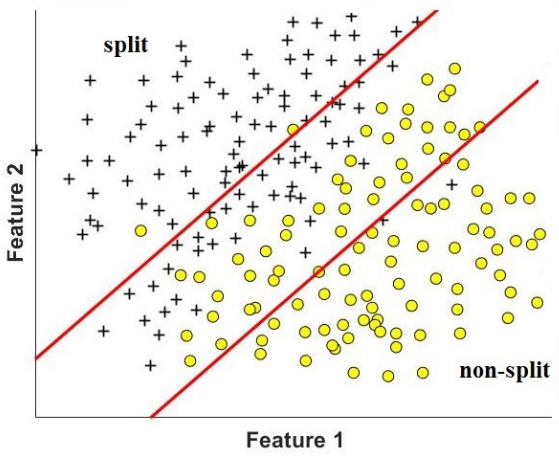

Fig. 1. Sample hyperplane positioning of the SVMs for two features

and non-split) in a given depth level. The best model pair is selected by evaluating the precision, which is calculated for split model (φ_s) and non-split model (φ_{ns}) as,

$$\varphi_s = \frac{T_p}{T_p + F_p} \tag{2}$$

and

$$\varphi_{ns} = \frac{T_n}{T_n + F_n}, \tag{3}$$

respectively. Here, T_p, T_n are the numbers of True Positive and True Negative samples, whereas F_p and F_n are the numbers of False Positive and False Negative samples, respectively.

Maximization of the precision for both models results in SVM *split* model minimizing the categorization of *non-split* samples as *split* while SVM *non-split* model minimizing the categorization of *split* samples as *non_split*.

Fig. 1 for example, depicts logical separation for the CU size decisions made by the corresponding SVMs at a particular depth level for two sample features. The data samples that fall in the region between the hyperplanes are sent to RD optimization to determine the CU split decision.

3) Updating the models: After a certain number of predictions are made, the SVM models are discarded and re-created in order to ensure that the models are content-adaptive. The ratio between the number of training samples and the number of predicted samples in this paper is maintained at 1:200.

B. Offline-only training

1) Data collection: The SVM models generated in this approach use the data collected offline prior to the current encoding process. The process followed is similar to that of the online-only approach, however, the whole process is carried offline.

The number of training samples used to create the offline models was also limited to 2000, in order to match the numbers in the online approach. Using higher number of training samples increases the model accuracy, but it also increases the number of support vectors resulting in an increased prediction complexity. This can drive away the implications of differences between offline and online model building.

2) Weight calculation: An approach similar to that of online training is carried out to determine the optimal weights for the offline weighted SVM models. However, neither the sample data nor the weights are updated for the offline trained models, thus, the initial models are continuously maintained throughout the encoding process.

IV. Experimental results and discussion

The two SVM models (online trained and offline trained), are implemented within the HM16 [21] reference encoder. For SVM implementation, the optimized *libSVM* [22] library has been used with Radial Basis Function(RBF) kernel and C set at 100 in both online and offline training. RBF is chosen given the fact that it can handle non-linear decision boundaries and it performs well with smaller number of features [22]. Having $C = 100$ enables generalization of the models to avoid overfitting.

Both algorithms are compared against HM16 to measure the encoding complexity reduction and coding efficiency impact. In this case, the encoding time performance $\Delta T(\%)$ is evaluated using,

$$\Delta \mathrm{T}(\%) = \frac{T_{HM} - T_\rho}{T_{HM}} \times 100, \tag{4}$$

where T_{HM}, and T_ρ are the encoding times of the HM reference encoder and the evaluating algorithm, respectively, for $\mathrm{QP} \in \{22, 27, 32, 37\}$. The impact on coding efficiency for the two algorithms is measured using the Bjøntegaard Delta Bit Rate (BDBR) increase [23]. The Table I illustrates the experimental results of the two approaches.

The experimental results show that offline-only model achieves a significant complexity reduction in the range of 74.34% on average, yet at the same time resulting in significant coding losses, 9.74 % BDBR increase on average. On the other hand, the online-only model demonstrates negligible coding losses, i.e., 0.96 % BDBR increase, yet at the same time results in less encoding complexity reductions (56.36%), compared to the offline-only model.

The relatively inferior encoding time saving of the online-only SVM model approach compared to the offline-only SVM model approach can be intuitively explained as follows.

- Online-only models gather training data during the encoding process, while using the RD optimization to determine the CU split decision
- Selecting the optimal weights for the SVM models require analysis of a range of weight value pairs in runtime

However, the online-only approach demonstrates a negligible BDBR increase due to the following reasons.

- Training and testing data (i.e., encoding data in this case) for the SVM models come from the same dynamic data distribution
- SVM models are being discarded and re-created periodically, which ensures that the CU split decisions more relevant to the content being encoded

TABLE I
COMPARISON BETWEEN STATIC AND CONTENT-ADAPTIVE MODELS (ALL INTRA MAIN).

Sequence	Online vs HM			Offline vs HM		
	ΔT (%)	BD-Rate (%)	BD-PSNR (dB)	ΔT (%)	BD-Rate (%)	BD-PSNR (dB)
BasketBallDrillText	36.6	0.45	-0.02	64.36	15.82	-0.84
Kimono	70.2	2.32	-0.08	81.23	5.8	-0.19
BasketBallPass	52.68	0.56	-0.03	72.33	10.71	-0.65
BQTerrace	74.57	0.91	-0.05	73.94	7.01	-0.38
Traffic	47.73	0.56	-0.03	79.84	9.38	-0.46
Average	56.36	0.96	-0.04	74.34	9.74	-0.5

V. CONCLUSION

Reducing the encoding complexity while keeping the coding efficiency of HEVC intact is a compelling challenge for resource constrained consumer electronic devices. Therefore, utilizing ground truth data in a given video sequence for training the decision making models results in more accurate CU split decisions. However, depending only on ground truth information can adversely affect the encoding complexity reduction due to the time consumed for online model training. In this context, future work will focus on developing hybrid decision making models that utilize both offline and online trained inference models that can make fast decisions while being adapted to the dynamic nature of the video contents.

REFERENCES

[1] G.J. Sullivan, J. Ohm, W.J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] I. Kim, J. Min, T. Lee, W. Han, and J. Park, "Block Partitioning Structure in the HEVC Standard," *IEEE transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1697–1706, 2012.

[3] S. Cho and M. Kim, "Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 23, no. 9, pp. 1555–1564, 2013.

[4] Y. Zhang, Z. Pan, N. Li, X. Wang, G. Jiang, and S. Kwong, "Effective Data Driven Coding Unit Size Decision Approaches for HEVC INTRA Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[5] M.U.K. Khan, M. Shafique, and J. Henkel, "An adaptive complexity reduction scheme with fast prediction unit decision for HEVC intra encoding," in *Proc. IEEE International Conference on Image Processing (ICIP)*. IEEE, 2013, pp. 1578–1582.

[6] Y. Zhang, S. Kwong, X. Wang, H. Yuan, Z. Pan, and L. Xu, "Machine Learning-Based Coding Unit Depth Decisions for Flexible Complexity Allocation in High Efficiency Video Coding," *IEEE Transactions on Image Processing*, vol. 24, no. 7, pp. 2225–2238, 2015.

[7] G. Correa, P. Assuncao, L. V. Agostini, and L. A. C. da Silva, "Fast HEVC encoding decisions using data mining," *IEEE trans. on circuits and systems for video technology*, vol. 25, no. 4, pp. 660–673, 2015.

[8] X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, pp. 4, 2013.

[9] X. Liu, Y. Li, D. Liu, P. Wang, and L.T. Yang, "An Adaptive CU Size Decision Algorithm for HEVC Intra Prediction based on Complexity Classification using Machine Learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[10] Y.F. Cen, W.L. Wang, and X.W. Yao, "A fast CU depth decision mechanism for HEVC," *Information Processing Letters*, vol. 115, no. 9, pp. 719–724, 2015.

[11] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, and D. 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, 2016.

[12] T. Li, M. Xu, and X. Deng, "A deep convolutional neural network approach for complexity reduction on intra-mode HEVC," in *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1255–1260.

[13] T. Laude and J. Ostermann, "Deep learning-based intra prediction mode decision for HEVC," in *Picture Coding Symposium (PCS), 2016*. IEEE, 2016, pp. 1–5.

[14] M. Xu, T. Li, Z. Wang, X. Deng, and Z. Guan, "Reducing Complexity of HEVC: A Deep Learning Approach," *arXiv preprint arXiv:1710.01218*, 2017.

[15] Q. Hu, Z. Shi, X. Zhang, and Z. Gao, "Fast HEVC intra mode decision based on logistic regression classification," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1–4.

[16] D. Ruiz-Coll, V. Adzic, G. Fernandez-Escribano, H. Kalva, J.L. Martinez, and P. Cuenca, "Fast partitioning algorithm for HEVC Intra frame coding using machine learning," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4112–4116.

[17] S. Zhu and C. Zhang, "A fast algorithm of intra prediction modes pruning for HEVC based on decision trees and a new three-step search," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21707–21728, 2017.

[18] B. Du, W. Siu, and X. Yang, "Fast CU partition strategy for HEVC intra-frame coding using learning approach via random forests," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*. IEEE, 2015, pp. 1085–1090.

[19] J. Chen and L. Yu, "Effective HEVC intra coding unit size decision based on online progressive Bayesian classification," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[20] T. Zhang, "An introduction to support vector machines and other kernel-based learning methods," *AI Magazine*, vol. 22, no. 2, pp. 103, 2001.

[21] "HM software manual [online] Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware," .

[22] C.C. Chang and C.J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[23] G. Bjontegarrd, "Calculation of average PSNR differences between RD-curves," *ITU - Telecommunications Standardization Sector STUDY GROUP 16 Video Coding Experts Group (VCEG)*, 2001.