VIDEO-ANALYTICS TASK-AWARE QUAD-TREE PARTITIONING AND QUANTIZATION FOR HEVC

Praneet Singh, Edward J. Delp, Amy R. Reibman

Elmore School of Electrical and Computer Engineering, Purdue University

ABSTRACT

Video analytics systems designed for computer vision tasks use deep learning models that rely on high-quality input data to maximize performance. However, in a real-world system, these inputs are often compressed using video codecs such as HEVC. Video compression degrades the quality of the inputs, thereby degrading the performance of these models. Region-of-interest (ROI) coding enables bits to be allocated to improve performance; however, the method to select regions should be computationally simple since it must occur during or before the video is compressed and transmitted for further processing. In this paper, we propose a task-aware quad-tree (TA-QT) partitioning and quantization method to achieve ROI coding for HEVC and other video coding standards. TA-QT uses a lightweight edge-based model to guide task-aware video encoding to improve end-stage video analytics (ESVA) performance while reducing both bit-rate and encoding time. We demonstrate the effectiveness of our approach in terms of (a) the performance of the ESVA on compressed inputs, (b) transmission bit-rates, and (c) encoding time.

***Index Terms*—** Video analytics, computer vision, deep learning, HEVC, HM, video compression, task-aware

1. INTRODUCTION

With the advancements in machine learning, many video analytics systems employ deep neural networks for computer vision tasks. These approaches perform extremely well in most scenarios provided a large number of high-quality data samples (ground-truth) are available for training. When these models are deployed in real-world systems with bandwidth constraints, they must operate on compressed inputs. Video codecs like HEVC [1] are used to compress the input video data on the edge which is then transmitted, decoded, and finally used by the end-stage video analytics (ESVA). While compression is required in a video analytics system to reduce the required transmission bandwidth, it can degrade the performance of the ESVA and incur significant processing time.

The bandwidth utilized in a practical video analytics system depends on the degree of compression used. In an ideal case, the ESVA of such a system (e.g., object detection, segmentation, or tracking) would perform perfectly with minimal bandwidth requirements. Although, these analytics are robust to some degree of compression, severe compression can hamper the performance of these task. Video analytics performance has been shown to degrade as the compression on the input data increases, for tasks such as object detection, segmentation, depth estimation [2], pedestrian detection [3], and person re-identification [4]. We also demonstrate here similar results for pedestrian and face detection by compressing the inputs using the reference implementation of HEVC, HM [5].

Deep learning models deployed in video analytics systems are task-specific. They learn to focus on task-related regions in the input which can be verified using class activation heatmaps [6]. Hence, *task-aware video encoding* approaches are well-suited for practical video analytics. These video encoding procedures ensure task-specific regions of the input have higher quality, do not degrade the ESVA performance, and reduce bit-rate by severely compressing other regions unrelated to the task. Approaches proposed in [7] and [8] generate task-aware encodings using perceptual quality metrics like PSNR and SSIM. More recent approaches [9–11] use class-activation heat maps [6] from complex fully-trained deep neural networks to achieve the same.

Unlike previous approaches to task-aware video encoding, modifying the frame partitioning used during encoding can lead to a simpler yet effective solution. For example, in HM, a recursive quad-tree (QT) partitioning scheme is used to decompose the frame into smaller blocks called Coding Units (CUs) until a QT that optimizes rate-distortion is obtained. Due to its recursive nature, it constitutes a major portion of the encoding time. Each CU is then quantized using a fixed Quantization Parameter (QP) that determines the degree of compression and also affects the partitioning structure. In our work, we refer to HM's frame partitioning and quantization scheme as *HM-QT*. HM-QT affects the ESVA performance and encoding bit-rate, as demonstrated in our experiments.

In this paper, we propose a Task-Aware Quad-Tree (*TA-QT*) partitioning and quantization strategy that replaces HM-QT. Figure 1 demonstrates our overall approach in the context of a practical video analytics system. TA-QT uses predictions from lightweight edge-based task-specific networks [12], [13] to generate task-aware QTs in a non-recursive manner. The CU partitioning structure in TA-QT is task-aware i.e., CUs in regions relevant to the task undergo finer partitioning while CUs in other regions remain unpartitioned. In addition to this, TA-QT varies the quantization for CUs to en-

THIS WORK HAS BEEN SUPPORTED BY THE FORD MOTOR COMPANY UNDER THE FORD-PURDUE ALLIANCE PROGRAM.

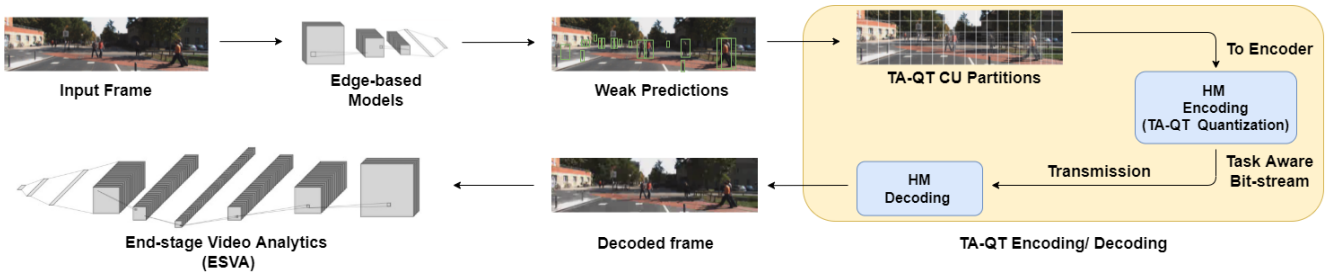


Fig. 1. Overview of a practical video analytics system; Our proposed TA-QT component is highlighted in yellow.

sure that task-specific regions always have higher quality and the other task unrelated regions are compressed more. TA-QT's non-recursive task-aware frame partitioning and quantization helps improve the ESVA performance, save transmission bit-rate and reduce the video encoding time.

In comparison to the task-aware approaches mentioned earlier, TA-QT has the following advantages: it relies on a lightweight network that is computationally less expensive than using class activation heat maps produced using a sophisticated fully trained network; its partitioning does not require recursion and is independent of QP; it can be used for both intra and inter-frame encoding; it can easily be incorporated into systems performing other ESVA tasks like segmentation and for upcoming video encoding standards like VVC [14].

In the upcoming sections, we first describe our approach and its efficacy. Next, we discuss the impact of frame partitioning and quantization during encoding on the ESVA performance. Finally, we present experiments for face and pedestrian detection that demonstrate TA-QT outperforms HM-QT in terms of ESVA performance, bit-rate, and encoding time.

2. BACKGROUND

HEVC prescribes an exhaustive and recursive QT frame partitioning procedure to optimize the rate distortion trade-off. The detailed procedure of partitioning the frame into QTs of Coding Units (CUs), Prediction Units (PUs) and Transform Units (TUs) is described in [1].

In the HM implementation, CUs are recursively partitioned and a QT that best optimizes the rate distortion is selected. Next, each CU is quantized based on a fixed QP (range $0-51$). We infer from [1] that larger CUs allow for better compression while smaller CUs improve quality. Thus, in HM, the recursive partitioning is repeated for each frame such that at lower QPs, CUs are partitioned finely all the way down to a size of 8x8 to obtain better quality. As the QP increases, the larger 64x64 CUs that are not partitioned become more abundant to achieve better compression. HM supports adapting the QP across frames, but not across individual CUs.

Several approaches [15–17] have been designed with the goal to replicate HM-QT's partitioning more efficiently. However, they focus solely on reducing the complexity of the partitioning procedure and encoding time, and they do not consider a subsequent ESVA task after the encoding process.

3. TA-QT PARTITIONING AND QUANTIZATION

Our proposed TA-QT approach generates a recursion-free task-aware QT for each frame using predictions from lightweight edge-based networks. These networks are selected such that they perform the same task as the ESVA with significantly fewer parameters. TA-QT follows the partitioning principles prescribed by HEVC [1] to generate task-dependent QTs. It also assigns task-aware quantization. With these changes, TA-QT creates task-aware bit-streams that improve ESVA performance, save transmission bit-rate, and reduce overall encoding time in video analytics systems.

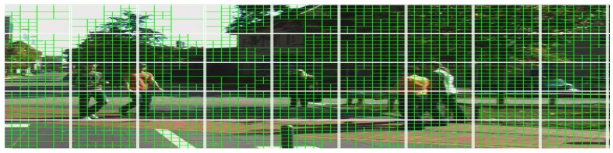
The CU partitioning for TA-QT is designed to allocate the most bits to the regions that contain objects, the least number of bits to background regions that contain no objects, and an intermediate number of bits to the contextual regions immediately surrounding an object. The latter is motivated by the observation that these contextual regions improve ESVA performance [18]. As such, TA-QT assigns small CU sizes in task-dependent regions, medium CU sizes in contextual regions, and large CU sizes in the background regions. Similarly, QPs are assigned based on CU size. The small CUs in task-specific regions are assigned lower QPs, and larger CUs are assigned larger QPs. Table 1 summarizes the assignment of CU size and QP based on content inside each region.

Region	Object	Context	Background
CU Size	8×8	$16 \times 16,\ 32 \times 32$	64×64
QP Range	(18-26)	(26-38)	(42-51)

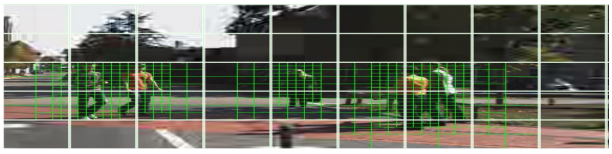
Table 1. Assignment of CU size and QP based on region

TA-QT also uses the task-aware CU partitions to create task-aware PU and TU partitions by leveraging the fact that they have the same root node [1]. Hence, it can also be applied for inter-frame encoding where the motion vectors and prediction residuals are estimated only in task-aware regions.

Figure 2 illustrates the partitioning and quantization of TA-QT and HM-QT on a frame from the KITTI test set. It is clear from these figures that TA-QT's partitioning and quantization is task-aware unlike HM-QT's i.e., finer CU partitions only lie in pedestrian regions and these are encoded with a much higher quality. The other regions not relevant to the task experience severe blocking artifacts due to unpartitioned CUs that experience a higher degree of compression.



(a) HEVC-QT



(b) TA-QT

Fig. 2. Comparison of HM-QT and TA-QT on a cropped frame from the KITTI dataset.

Thus, in practical video analytics systems, TA-QT achieves better compression in comparison to HM-QT as follows:

- Finer 8x8 CUs occur in task-aware regions and are quantized with lower QPs (18-26) for better quality. This improves the ESVA performance;
- 16x16 and 32x32 CUs occur in regions with contextual information. These are not as important as the task-specific regions and are quantized with slightly higher QPs (26-38) to achieve a balance between quality and compression;
- 64x64 CUs occur in regions not relevant to the task. They are quantized with the highest QPs (42-51) to achieve a significant reduction in bit-rate;
- TA-QT produces more 64x64 CUs; these do not require partitioning, which also reduces encoding time.

4. EXPERIMENTS

In this section, we first explore the impact of CU partitioning and quantization on the ESVA performance. Next, we compare TA-QT and HM-QT in terms of Mean Average Precision (mAP) of the ESVA, encoding bit-rate, and encoding time.

4.1. Experimental Setup

4.1.1. Datasets

We consider pedestrian and face detection as ESVA tasks to demonstrate TA-QT. For pedestrian detection we use the KITTI dataset [19], and for face detection we use IJB-C [20]. The KITTI test set containing 30 video sequences is used to evaluate inter-frame encoding while the IJB-C dataset with $272,366$ images is used to evaluate intra-frame encoding.

4.1.2. Edge-Based Models

TA-QT generates ROI partitions using predictions from edge-based models. These models are selected to be computationally lighter, faster and perform the same task as that of the ESVA networks. For pedestrian detection, we use a Lightweight Pedestrian Detector (LPD) from [13]. For face detection, we use RetinaFace-MobileNet0.25 from [12].

The models used have fewer than 3 million parameters compared to over 25 million parameters for their ESVA counterparts. With fewer parameters, their performance suffers at higher prediction confidence thresholds; difficult ROIs are missed which results in them being compressed more than desired. This degrades the ESVA performance. To ensure no ROIs are missed, we lower the prediction thresholds for these models. This does increase the required transmission bit-rate due to false positives, but it improves overall performance.

4.1.3. TA-QT and HM-QT Encoding

We encode the KITTI and IJB-C dataset with TA-QT and HM-QT at different bit-rates (QPs). In both cases, we use the HM reference software. For HM-QT, we encode the datasets with HM at fixed QPs in the range of 18-51. For TA-QT, we modify the HM encoder to accept task-aware partitions generated from the lightweight network predictions. In addition, four fixed QPs for each CU size are specified while encoding. With these changes, TA-QT creates a task-aware video encoding using the procedure specified in Section 3.

4.1.4. End-Stage Video Analytics (ESVA)

We apply Yolov4 [21] ($27.3M$ parameters) for pedestrian detection and RetinaFace - ResNet50 [22] ($29.3M$ parameters) for face detection. Yolov4 is trained on the uncompressed KITTI training set, and tested on compressed versions of the KITTI test set. Similarly, RetinaFace-ResNet50 is trained on WiderFace [23] and tested on compressed versions of IJB-C.

4.2. Effects of Video Compression and Frame Partitions

Although previous results have been presented on the effects of video compression on learning tasks, the effects of frame partitioning on these tasks has not been evaluated. Here, we demonstrate how both compression and frame partitioning affect the performance of the ESVA.

In this experiment, KITTI sequences are encoded with HM at different fixed QPs and different fixed CU sizes. Yolov4's mAP% on these encoded KITTI test sets is shown in Figure 3. We see that as compression increases with larger QP, the ESVA performance decreases because the frame quality degrades. Comparing the different points at each QP in Figure 3 shows the effect of varying the CU partition size. At lower QPs, we can reduce the bit-rate without affecting the mAP%, while at the higher QPs, the mAP% can be markedly improved by modifying the CU partitioning.

4.3. TA-QT vs HM-QT

Here, we provide experimental evidence that TA-QT outperforms HM-QT in terms of ESVA mAP%, transmission bit-rate, and encoding time, making it more viable in practice.

Figure 4 shows the ESVA performance on the TA-QT and HM-QT encoded test sets at different bit-rates. Inter-frame

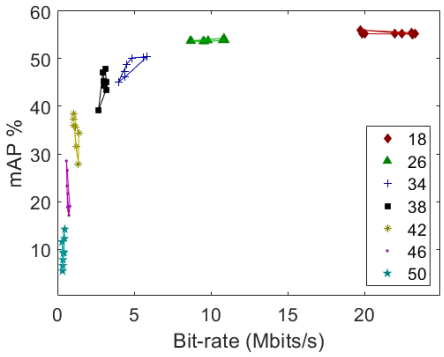
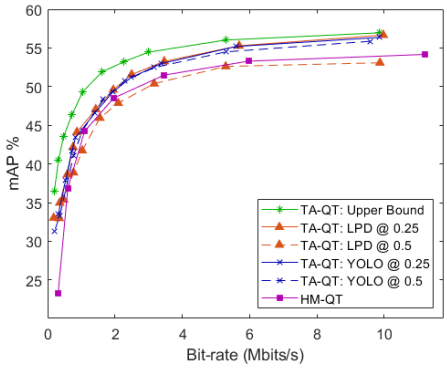
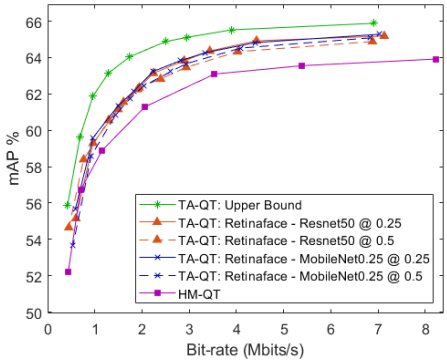


Fig. 3. Pedestrian detection performance of Yolov4 on KITTI for encodings with fixed QPs and fixed CU partition sizes.

encoding results are shown in Figure 4(a) for pedestrian detection on the KITTI test set, while Figure 4(b) shows the intra-frame encoding results for face detection on IJB-C.



(a) TA-QT vs. HM-QT; inter-frame encoding on KITTI.



(b) TA-QT vs. HM-QT; intra-frame encoding on IJB-C.

Fig. 4. mAP% vs Bit-rate comparison

The green curve indicates a upper-bound on the performance that could be achieved by the ESVA if TA-QT were to use ground-truth regions from the testsets instead of relying on network predictions. Compared to HM-QT (pink curve), there is significant room for ROI-encoding to improve.

The orange curves indicate performance when a lightweight detector is used identify task-specific regions for TA-QT, and the blue curves correspond to a more computationally-complex detector. The dashed lines indicate when these models operate at a typical detection threshold of 0.5 Intersection Over Union (IOU), and the solid lines indicate a lower threshold of 0.25. Particularly for the inter-frame encoding on KITTI, using a lower threshold improves performance. This is because at a higher threshold, difficult ROIs are missed and therefore compressed heavily, affecting overall performance. At a lower threshold, more ROIs are identified and overall ESVA performance is improved. Note that both the detectors perform noticeably better than HM-QT, but neither achieve the upper-bound performance, particularly for lower bit-rates. Using a computationally-complex detector is similar to previous work in this domain, but is impractical due to the increased complexity at the encoder. Therefore, our approach of using lightweight models at the edge is preferred.

Tables 2(a) and 2(b) compare the approaches in terms of the minimum bit-rate required to achieve a specific mAP % for each task and the corresponding average encoding time. We see that TA-QT achieves significant bit-rate savings while reducing encoding time even with the added requirement of extracting predictions from a lightweight network.

KITTI, Inter-Encoding				
map %	Min. Bit-rate (Mbps)		Avg. Encoding Time (s)	
	TA-QT	HM-QT	TA-QT	HM-QT
55	5.38	>11.21	868.24	1421.36
50	2.05	2.71	736.23	1260.33
45	1.02	1.26	656.66	1145.32
40	0.64	0.82	608.43	995.43
35	0.36	0.57	574.33	961.12

(a) TA-QT (LPD @ 0.25) vs HM-QT

IJB-C, Intra-Encoding				
mAP %	Min. Bit-rate (Mbps)		Avg. Encoding Time (s)	
	TA-QT	HM-QT	TA-QT	HM-QT
65	5.42	>8.3	7.63	15
62.5	1.95	3.12	6.48	12.42
60	1.01	1.58	5.72	9.86
57.5	0.76	0.87	4.46	7.63
55	0.57	0.61	3.98	6.02

(b) TA-QT (MobileNet0.25 @ 0.25) vs HM-QT

Table 2. Comparison of Min. Bit-rate and Avg. Encoding time to achieve specific mAP % on ESVA.

5. CONCLUSIONS

In this paper, we proposed a novel task-aware partitioning and quantization scheme in TA-QT that can be incorporated in practical video analytics systems. TA-QT is free from recursion and independent of QP, unlike HM-QT. Furthermore, it can be easily adapted for newer video coding standards like VVC and for other ESVA tasks like tracking. In our experiments, we showed that TA-QT outperforms HM-QT in terms of ESVA performance, bit-rate savings, and encoding time. However, improvements are possible in ROI selection, as seen by the fact that using the ground-truths for partitioning further improves the bit-rate versus mAP trade-off. Predicting CU QPs based on bandwidth availability and utilizing inter-frame CU partitions are other areas where TA-QT can be improved.

6. REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. 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, 2012.

[2] M. Poyser, A. Atapour-Abarghouei, and T. Breckon, "On the impact of lossy image and video compression on the performance of deep convolutional neural network architectures." in *25th International Conference on Pattern Recognition (ICPR2020)*, September 2020.

[3] K. Tahboub, A. R. Reibman, and E. J. Delp, "Accuracy Prediction for Pedestrian Detection," in *International Conference on Image Processing (ICIP)*, 2017.

[4] H. Chen, E. Delp, and A. Reibman, "Estimating Image Quality for Person Re-Identification," in *International Workshop on Multimedia Signal Processing*, 2021.

[5] Joint Video Exploration Team, "HEVC HM reference software," https://vcgit.hhi.fraunhofer.de/jvet/HM, 2013.

[6] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "GRAD-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 839–847.

[7] Y. Gitman, M. Erofeev, D. Vatolin, B. Andrey, and F. Alexey, "Semiautomatic visual-attention modeling and its application to video compression," in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 1105–1109.

[8] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 19–33, 2013.

[9] Q. Cai, Z. Chen, D. Wu, S. Liu, and X. Li, "A novel video coding strategy in HEVC for object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[10] Y. Mei, F. Li, L. Li, and Z. Li, "Activation Map Saliency Guided Filtering for Efficient Image Compression for Vision Tasks," in *54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 1117–1121.

[11] J. Shi and Z. Chen, "Reinforced bit allocation under task-driven semantic distortion metrics," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020.

[12] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[13] Y. He, D. Xu, L. Wu, M. Jian, S. Xiang, and C. Pan, "LFFD: A light and fast face detector for edge devices," *arXiv preprint arXiv:1904.10633*, 2019.

[14] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the Versatile Video Coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[15] K. Kim and W. W. Ro, "Fast CU depth decision for hevc using neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1462–1473, 2018.

[16] 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*, vol. 29, no. 1, pp. 144–155, 2019.

[17] S. Bouaafia, R. Khemiri, F. E. Sayadi, and M. Atri, "Fast CU partition-based machine learning approach for reducing HEVC complexity," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 185–196, 2020.

[18] P. Hu and D. Ramanan, "Finding tiny faces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 951–959.

[19] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[20] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney *et al.*, "IARPA Janus benchmark-C: Face dataset and protocol," in *IEEE International Conference on Biometrics*, 2018, pp. 158–165.

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[22] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5203–5212.

[23] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A Face Detection Benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5525–5533.