Edge-Cloud Collaborative Processing for Intelligent Internet of Things: A Case Study on Smart Surveillance

Burhan A. Mudassar
Georgia Institute of Technology
Atlanta, GA
burhan.mudassar@gatech.edu

Jong Hwan Ko
Georgia Institute of Technology
Atlanta, GA
jonghwan.ko@gatech.edu

Saibal Mukhopadhyay
Georgia Institute of Technology
Atlanta, GA
saibal@ece.gatech.edu

ABSTRACT

Limited processing power and memory prevent realization of state of the art algorithms on the edge level. Offloading computations to the cloud comes with tradeoffs as compression techniques employed to conserve transmission bandwidth and energy adversely impact accuracy of the algorithm. In this paper, we propose collaborative processing to actively guide the output of the sensor to improve performance on the end application. We apply this methodology to smart surveillance specifically the task of object detection from video. Perceptual quality and object detection performance is characterized and improved under a variety of channel conditions.

KEYWORDS

Edge-Cloud Partitioned Processing, Content Aware Feedback, Object Detection

1 INTRODUCTION

The internet-of-things (IoTs) platforms for visual data analytics find applications in many areas such as home, commercial, and military surveillance, robotics, autonomous vehicles, to name a few [1, 2]. A major challenge to scaling number of edge devices in these platforms is the high data volume of image sensors which strains the shared channel bandwidth. For example, a surveillance camera transmitting a 1920 x 1080 video at 30 fps will require a bandwidth of 1492 mega bits per second (Mbps). A second challenge is increased human cost and decreased operator efficiency due to many cameras[1]. Consequently, smart surveillance requires automation of subtasks such as detection, classification, or recognition of objects of interest or anomalous behavior.

The common approaches to solve the preceding challenges are often at odds with each other. For example, common compression techniques such as MJPEG are employed at the sensor to reduce data volume. Likewise Convolutional Neural Networks (CNNs) can be used at the cloud to enable machine intelligence with near human level accuracy and generalization performance [3–6]. However, object detection accuracy of a CNN is adversely impacted due to video compression artifacts from MJPEG in bandwidth constrained

Figure 1: System Architecture of IoT with centralized processing at host. We propose active host to sensor feedback to improve task performance.

channel as illustrated in Figure 2(a). Further, video compression also reduces confidence in true positive (and true negative) thereby degrading the separability between positive and false events, and hence, increasing the need for human intervention. The computation complexity and memory demand of the object detection CNN makes it difficult to deploy such network at the edge devices.

This paper presents edge-host collaborating processing to improve the quality of DNN processing at the host by providing feedback to the edge (sensor and conventional compression engine) and guide the device to transmit useful information for the end task with highest quality (Figure 1). We assume conventional CNN processing algorithm such as object detection network at the host, and augment that with computation kernels to generate desired feedback supported by multi-quality compression algorithms at the edge device [7].

Our approach is rooted in the concept of Region-Of-Interest (ROI) based processing where ROIs are transmitted with higher quality. The ROI based processing is used many camera, where simple ROI based on motion, is detected at the edge using low-cost algorithms. However, computationally intensive nature of the object detection algorithms prevents ROI detection at the edge. For example, even a simple object detector, tiny YOLO [4] consumes 860.7 ms of latency and 2152 mJ energy for a frame on a mobile GPU (mGPU).

Instead, we propose to provide the ROI information through a feedback path from the host, for example, bounding box coordinates containing objects of interest, generated by the object detection CNN from frame ' i ' to the sensor node. A multi-QF MJPEG then encodes ROI and non-ROI regions of frame ' $i + 1$ ' with different Q_F . We demonstrate the application of rate controllers that selects the optimal Q_F values for ROI and non-ROI regions to compress the input image to meet the transmission bandwidth of the channel. As illustrated in Figure 2(b) preserving the quality of the ROI regions increases object detection accuracy (new detection and higher confidence) at the host.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DAC '18, June 24–29, 2018, San Francisco, CA, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5700-5/18/06...\$15.00

<https://doi.org/10.1145/3195970.3196036>

ROI SSIM : 0.972

Size : 82 kB

(a)

ROI SSIM : 0.984

Size : 81 kB

(b)

Figure 2: Object Detection Results for Person Class (a) Uniformly degraded frame at QF = 8 (b) Multi-QF frame with ROI QF = 20 and non-ROI QF = 0

The effectiveness of the host to edge feedback faces two interesting challenges. First, feedback compute latency at the host and transmission delay through the channel may cause the feedback information to be "stale" on arrival. We solve the first problem by creating a prediction module at the host that offsets the ROI information to account for feedback latency. Second, new objects of interest entering the frame in non-ROI regions can be misdetected due to lower quality. We address the second challenge by training the object detector with augmented image data containing multiple quality levels.

We characterize the performance of our approach quantitatively on the MOT Multi-Object Tracking Dataset [8] and achieve an improvement of 5% in Average Precision for the Pedestrian Detection task while operating at a bandwidth of 2 Mbps. Moreover, we show good separability between positive and negative detections by improving the Equal Error Rate (EER) by 3%. We synthesize the edge system in 28nm ASIC and estimate latency and energy consumption. Our approach adds a negligible energy overhead (19nJ) due to multi-QF implementation of MJPEG engine compared to baseline single QF design.

2 RELATED WORK

Object Detection: Object Detection is a challenging computer vision task as it involves both classification and accurate localization of all objects within an image. Deep learning based architectures have made huge gains possible in this task. Of these the two stage detectors first popularized by R-CNN have been widely adopted due to their competitive accuracy [5, 6] where class agnostic object proposals are generated followed by classification.

Even complex DNNs are prone to recognition errors when faced with distorted inputs. S. Dodge et. al estimate the effect of JPEG compression on the performance of CNN for the image classification task [9]. At lower quality factors, performance significantly decreases. Kang et. al observe that frame based object detection approaches such as R-CNN and its variants, when applied to object detection in video yield noisy detections even with minimal variations across successive frames. [10].

Embedded implementations of object detectors comprised of DNNs are in their infancy right now. Nevertheless, there is a growing interest in transferring this capability to the edge. In recent years, the Low Power Image Recognition Challenge (LPIRC) has acted as a catalyst for research into this domain [11]. Winning approaches have used variants of Fast or Faster R-CNN with frame rates in the 1-2 fps mark implemented on the Jetson TK1 mobile SoB.

Host To Sensor Feedback Extensive literature exists on Adaptive Quality of Service Video Streaming with adoptions in H.263 and H.264. These approaches are primarily centered around entropy coding, error sensitivity based feedback, motion-based ROI coding [12]. We adopt an orthogonal approach wherein the ROI itself is defined as the metric for information and is encoded at a higher quality.

3 PROPOSED APPROACH

Figure 3 shows the overall flow of the proposed approach. We create a feedback path from the output of the complex task at the host to the sensor. At the host side, an object detection module processes incoming frames and generates bounding box coordinates and confidence scores for classes of interest. To address the challenge of finite latency between the time the frame was sampled to the time the feedback arrives at the sensor, we create a prediction module at the host. The goal of the prediction module is to maintain a motion model for each detected object. Using this motion model, the predictor compensates for the feedback latency by translating the bounding box coordinates.

Object detection is performed on a frame by frame basis and thus its output does not maintain object identities across successive detections. As identified by Kang et. al. [10], detection performance is also noisy and prone to mis-detections even with minimal scene variations across frames. We adopt approaches from object tracking literature to (a) associate detection across frames and (b) smooth predictions when faced with noisy detections or misdetections.

At the sensor side, a multi-QF JPEG encoder encodes the frame block by block. The encoder control module uses the ROI map to

Figure 3: Block Diagram of Proposed System with ROI Feedback from Host to Sensor

Figure 4: Architecture of R-FCN Object Detector with RPN[6]

Figure 5: Fine-tuning of R-FCN with augmented data containing images at various QF

assign ROI/nonROI status to the block. We adopt rate controller approach of Ko et. al. [13] and use two PID controllers for controlling the QF of ROI and non-ROI regions to meet data rate constraints. Given a target frame size T , we partition the target frame size into $targetframesize_{ROI}$ and $targetframesize_{nonROI}$. A value of 50 percent of total size T is chosen for ROI size empirically. Note that an equal distribution of size targets does not necessarily mean equal QF as ROI blocks are more sparsely distributed.

3.1 Object Detection

We adopt a two stage detector R-FCN [6] for implementation at the host due to its competitive accuracy and speed. The R-FCN network consists of a convolutional feature extractor followed by a region proposal network (RPN) which generates class agnostic proposals at multiple scales and locations (Figure 4). A classification layer generates per class probabilities for each proposal. A regression layer generates bounding box coordinates for each proposal location. Multiple overlapping proposals are then eliminated by non-maximum suppression (NMS) both after the RPN and after the classifier. ResNet-101 [14], a residual deep network with 100 convolutional layers is used as the backbone feature extractor. The backbone is pre-trained for the Image Classification task on ImageNet then trained for the Object Detection task on PASCAL VOC and MS COCO.

3.2 Multi-QF Training

As identified earlier, transmitting at variable quality poses problems for detection of new objects in non-ROI regions of the frame. We adopt a data-driven approach to solve this problem (Figure 5). The R-FCN network is fine-tuned for images generated at multiple at JPEG quality factors. Images from the MS COCO Dataset [15] train split are augmented and used for training the R-FCN network. We divide the MS COCO Dataset into 5 partitions with each partition representing quality factors ranging from 10 to 100. The R-FCN network then is trained with a low learning rate (fine-tuned) on these images using the Caffe framework.

3.3 ROI Prediction

A flow diagram of the multi-object ROI prediction algorithm is shown in Fig. 6. The ROI prediction module consists of two components i.e. an association module and the motion predictor. We adopt the SORT algorithm [16] for multiple object association and prediction. Association is necessary due to the fact that detector operates on a frame by frame basis. Given a set of detections D , we have predictions from multiple predictors P .

An optimal pairing can be found by casting this as a bi-partite graph matching problem and optimizing a cost function i.e. in this case the overlap between the predicted and detected bounding boxes. The Hungarian algorithm provides an optimal solution to the graph matching problem. The output of the association algorithm will result in some matched and unmatched detections. For each unmatched detection, a new predictor is created. If a certain predictor is not matched with a detection for a significant amount of time, then it is deleted. Detections and Predictions for a sample case are shown in Fig. 7

At the core of each predictor is the Kalman filter. The Kalman filter is a state prediction model that assumes Gaussian distributed state variables and a linear dynamical motion model. In our case, the visible state variables are the 4 coordinates of the bounding box while the latent state variables are the velocities of each coordinate. Detections at every frame are used to update the corresponding state variables. A missing detection at any point in time increases the uncertainty of the prediction made by the filter. This is advantageous for us in cases where a mis-detection occurs. Figure 8 compares two cases where ROI information is applied with and

Figure 6: Information Flow for Multi-Object Prediction

Figure 7: Detections and Predictions coming from the Neural Network (red) and ROI Prediction (blue)

without prediction. The object under consideration falls partially in a non-ROI region when prediction is not applied.

4 METRICS FOR EVALUATION

The accuracy of the object detection task is evaluated by considering two metrics i.e. Precision and Recall. These metrics are designed to penalize wrong classifications as well as incorrect localizations. For each bounding box generated by the object detector a confidence score per class is returned. A detection is considered correct when the confidence for the correct class is greater than a specific threshold and has sufficient overlap with the ground truth. Overlap is evaluated with the PASCAL VOC Criteria of Intersection over Union (IoU). An IoU ratio of greater than 0.5 is considered to be sufficient overlap.

Recall. The Recall Metric evaluates the coverage of the detector. It is defined as the ratio of correct detections to the sum of correct detections and missed detections and is given in equation (1). A high value of the Recall Metric ensures that the maximum amount of objects are detected. Depending on the application, a

Figure 8: Impact of ROI prediction on quality of ROI regions. Top: Without ROI prediction. Bottom: With ROI Prediction

Table 1: Host Platform Specifications and Latencies

Module	Latency (ms)	Platform
Object Detector	128	GTX 1080 Ti (GPU)
ROI Prediction	0.127	i7-7900k (CPU)

lower detection threshold can be set to increase the value of Recall.

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (1)$$

Precision. The Precision Metric evaluates the correctness of the detector. It is defined as the ratio of correct detections to the sum of correct detections and false detections and is given in equation (2). A high value of the Precision Metric ensures that the majority of the detections are correct. Depending on the application, a higher detection threshold can be set to increase the value of Precision.

$$\text{Precision} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}} \quad (2)$$

Separability. Separability of positive and negative detections indicates the strength of the classifier. Given a set of confidence scores for positive and negative detections, a detection threshold can be set that will adjust the False Positive Rate (FPR) and True Positive Rate (TPR) of the system. A common metric for selection of detection threshold is the Equal Error Rate (EER) which gives a detection threshold at which probability of mis-classifying a positive example is equal to a mis-classifying negative example and is given by equation (3).

$$\text{EER} : 1 - \text{TPR} = \text{FPR} \quad (3)$$

Perceptual Quality. SSIM is a similarity metric for images to capture perceptual quality. In the case of objection detection, we compare the SSIM of the ground-truth objects with the detected objects, not the overall image, to understand how well the perceptual quality of the ROI are preserved.

Figure 9: Distribution of confidence scores for (a) Uniform and (b) Adaptive Approach

Figure 10: Results on MOT17-04 for fixed bandwidth of 2 Mbps (a) Recall (b) Precision and (c) SSIM

5 EXPERIMENTAL SETUP

Simulation Model for Feedback Feedback is simulated by incorporating network latencies in our simulation. At a fixed bandwidth B , the transmission latency from the edge to the host consists of the serialization latency from the edge $t_{seredge}$ and the propagation latency t_{prop} . The incoming frame is processed by the detector and generates bounding boxes and class probabilities after t_{CNN} .

The ROI predictor offsets the bounding boxes to account for delays with latency t_{pred} . Note that $t_{pred} \ll t_{CNN}$. The bounding box coordinates are then sent back to the host with serialization latency $t_{serhost}$ and propagation latency t_{prop} . Note $t_{serhost} \ll$

Figure 11: Results on MOT17-04 for variable bandwidth (a) Recall (b) Precision and (c) SSIM

Table 2: Energy Consumption and Latency for Sensor Modules

Module	Latency (ms)	Energy Consumption (uJ)
Uniform JPEG	42.1	152.28
Multi-QF JPEG	42.1	171.72
Rate Controller	0.001	0.02

$t_{seredge}$ due to the obviously smaller size of coordinate information compared to image information. The combined latency is then given by equation (4).

$$t_{feedback} = t_{seredge} + 2 * t_{prop} + t_{CNN} + t_{pred} + t_{serhost} \quad (4)$$

Given a particular sampling latency t_{samp} , determining the frame at which feedback will arrive becomes simple with the above formulation and is given by equation (5). A value of $offset_{feedback} < 1$ implies feedback arrives before the next frame is sampled. High values are tolerable for static or relatively slow moving objects but are not suitable for higher degrees of motion and the introduction of new objects into non-ROI regions of the frame

$$offset_{feedback} = \frac{t_{feedback}}{t_{samp}} \quad (5)$$

Host Configuration At the host side, a GPU implementation of the R-FCN detector is used. For ROI prediction, we use the CPU implementation of the SORT algorithm [16]. Latency for both modules is given in Table 1. For ROI prediction, the latency number corresponds to an average of 20 predictors.

6 RESULTS

Evaluation is performed on subsequence 4 of the MOT17 Multi-Object Tracking Dataset. We define the ROI in this sequence as pedestrians. This sequence contains videos sampled at 30 fps and at a resolution of 1920 x 1080. At a fixed bandwidth, experiments

Figure 12: Required Bandwidth per Node and Number of Operational Nodes for a shared bandwidth of 20 Mbps with a performance target of Recall = 0.5

are conducted by varying the target frame rate and sub-sampling the sequence accordingly. Normalized frame rate is calculated by dividing the target frame rate by the sampling rate. Figure 9 shows that proposed approach with adaptive feedback can increase the separability between the true positive and true negative detection by increasing the confidence of the true positives. EER is improved over baseline by 3%.

Performance Analysis-Fixed Bandwidth As the target frame rate decreases, frames can be sent at a higher quality. As shown in Figure 10, Precision is improved at multiple target frame rates with our proposed approach. SSIM performance for the Adaptive approach is improved over the Uniform approach as target frame rate is increased indicating higher quality transmission of ROI. Precision is also improved by a margin of 5%.

Performance Analysis-Variable Bandwidth Performance of the adaptive approach is characterized against variable bandwidth. Bandwidth is varied from 2 Mbps to 20 Mbps. A target frame rate of 3 fps is chosen. As shown in Figure 11, the advantage of the adaptive approach is apparent for all three metrics at 2 Mbps with improvements of 5% and 6% in Precision and Recall respectively. At higher bandwidths, quality of ROI/nonROI regions becomes the same for both approaches hence no difference in performance is observed.

Energy Consumption Analysis We synthesize the multi-QF JPEG module and PID Rate Controllers in 28 nm ASIC and estimate the latencies and energy consumed per frame. For the baseline case, we synthesize a uniform JPEG encoder. Latency and energy consumed per frame is given in Table 2. A minimal energy overhead of 19.44 nJ is incurred at the sensor side for the multi-QF case.

Scaling Number of Edge Devices We use the measurements from the variable bit rate bandwidth experiment to project operational capacity of the network at a shared bandwidth of 20 Mbps with a performance target defined by the maximum of the evaluated metrics for the smallest bandwidth. As shown in Fig. 12, A larger number of nodes can operate at a fixed performance target for the adaptive approach showing better scalability to large IoT platforms.

Bandwidth Overhead Bandwidth overhead is incurred as a result of sending ROI information from the host to the sensor in form of bounding box coordinates. A maximum of 50 predictors

are maintained in the MOT17-04 subsequence. Using 16 bits per coordinate, maximum bandwidth overhead is calculated to be 3.12 kB per frame. For a target frame rate of 3 fps, the overhead is 74 kbps.

7 CONCLUSIONS

We have presented a framework that integrates task information produced from a computationally expensive algorithm at the host to guide the data collection and transmission from a resource constrained sensor under limited channel bandwidth. The proposed approach shows that collaborative processing between host and edge can enhance the quality of information processing in an IoT platform even with basic edge devices with negligible energy overhead. The future work needs to consider how such collaborative processing can be extended to future platforms with smart edge devices.

8 ACKNOWLEDGEMENTS

The research reported here was supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-17-2-0045. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA.

REFERENCES

- [1] Honghai Liu, Shengyong Chen, and Naoyuki Kubota. Intelligent video systems and analytics: A survey. *IEEE Transactions on Industrial Informatics*, 9(3):1222–1233, 2013.
- [2] Mario Gerla, Eun-Kyu Lee, Giovanni Pau, and Uichin Lee. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In *WF-IoT*, 2014.
- [3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.
- [4] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *CVPR*, 2017.
- [5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.
- [6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016.
- [7] Y. Hu, F. Meng, and Y. Wang. Improved jpeg compression algorithm based on saliency maps. In *ICISP*, 2012.
- [8] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, March 2016. arXiv: 1603.00831.
- [9] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *QoMEX*, 2016.
- [10] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. Early Access.
- [11] Huizi Mao, Song Yao, Tianqi Tang, Boxun Li, Jun Yao, and Yu Wang. Towards real-time object detection on embedded systems. *IEEE Transactions on Emerging Topics in Computing*, 2016. Early Access.
- [12] Satyajayant Misra, Martin Reisslein, and Guoliang Xue. A survey of multimedia streaming in wireless sensor networks. *IEEE communications surveys & tutorials*, 10(4):18–39, 2008.
- [13] Jong Hwan Ko, Burhan Ahmad Mudassar, and Saibal Mukhopadhyay. An energy-efficient wireless video sensor node for moving object surveillance. *IEEE Transactions on Multi-Scale Computing Systems*, 1(1):7–18, 2015.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- [16] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016.