A Camera with Brain – Embedding Machine Learning in 3D Sensors

Burhan Ahmad Mudassar, Priyabrata Saha, Yun Long, Muhammad Faisal Amir, Evan Gebhardt,
Taesik Na, Jong Hwan Ko, Marilyn Wolf and Saibal Mukhopadhyay
School of ECE, Georgia Institute of Technology, Atlanta, Georgia, USA
{burhan.mudassar, priyabratasaha, yunlong, mfamir, egebhardt6, taesik.na, jonghwan.ko}@gatech.edu,
{wolf, saibal}@ece.gatech.edu

Abstract—**The cameras today are designed to capture signals with highest possible accuracy to most faithfully represent what it sees. However, many mission-critical autonomous applications ranging from traffic monitoring to disaster recovery to defense requires quality of information, where useful information depends on the tasks and is defined using complex features, rather than only changes in captured signal. Such applications require cameras that capture useful information from a scene with highest quality while meeting system constraints such as power, performance, and bandwidth. This paper will discuss the feasibility of a camera that learns how to capture task-dependent information with highest quality, paving the pathway to design a camera with brain. 3D integration of digital pixel sensors with massively parallel computing platform for machine learning creates a hardware architecture for such a camera. The paper will discuss embedded machine learning algorithms that can run on such platform to enhance quality of useful information by real-time control of the sensor parameters. We conclude by identifying critical challenges as well as opportunities for hardware and algorithmic innovations to enable machine learning in the feedback loop of a 3D image sensor based camera.**

Index Terms—**Smart cameras, Embedded vision, Machine learning based Feedback, 3D-stacked sensor, Digital Pixel Sensor**

I. Introduction

The design of high quality cameras have always been at the forefront of electronics industry. The advent of digital pixel technologies and 3D integration promises unprecedented gains in resolution and frame rates of cameras [1]–[7]. The 3D die-stacking have shown major performance gains in traditional cameras [6], [7]. Unlike an analog pixel, a digital pixel has dedicated A/D conversion and memory buffers for each pixel, for example, using pulsed frequency counters. This close coupling allows higher dynamic range (>100 dB) and faster readouts (> 10k fps). But the individual pixel size can be larger (and fill factor can be lower) due to the in-pixel peripheral circuits. Fine-grain 3D integration can alleviate this problem by vertical stacking the photosensors and in-pixel read-out circuits [8]. An additional advantage is the feasibility of heterogeneous integration of photosensors and ROICs in different stacks, for example, coupling CMOS-based read out circuitry (ROIC) with non-silicon based photosensors [5] for hyperspectral sensing.

The cameras are designed to always capture signals at the highest possible quality to most faithfully represent what it sees. However, the amount of data a camera can transmit to an user depends on the available bandwidth of the transmission channel. The signal entropy-driven image and video encoding techniques within the camera have been used to reduce transmitted data. The smarter cameras often augment basic encoding techniques to improve quality, for example, by delivering an "abstracted output" that combines the output of a semantic task with the raw video stream through color coding or by sending as a separate data stream [9]; or by using the output of semantic tasks to enable Region-of-Interest (ROI)-based processing of video using low-level tasks such as edge detection or motion detection [10]. More recently, research efforts have been directed to integrate higher level deep learning [11] tasks within a camera, for example, classification, object detection, or activity recognition, leading to conceptual architectures of 3D sensors with integrated deep learning [7], [8], [12].

We are at a confluence in camera technology where advances in 3D stacking, digital pixel sensors and deep learning based vision can come together to realize the goal of a smart camera in the truest sense of the word [1], [3], [4], [6], [8], [11], [12]. However, even in emerging cameras with integrated learning [8], [12] the image processing pipeline has remained almost the same i.e. an image sensor captures the image followed by low-level signal processing to clean, encode and store/transmit the data followed by high level tasks. As cameras are being increasingly used to drive many mission-critical autonomous applications ranging from traffic monitoring to disaster recovery to defense, we argue that a uni-directional processing pipeline misses the opportunity to create a 'true' smart camera. In such applications useful information depends on the tasks and is defined using complex features, rather than only changes in captured signal. For example, an autonomous car needs to only focus on areas of the image that directly impact the quality of its driving such as road segmentations, proximal objects etc. Likewise, in a surveillance scenario where motion is not the only factor in determining objects/activities of interest such as a suitcase of a certain color left idle at a terminal can be the object of interest. Such applications require cameras that capture such useful information from a scene with highest quality while meeting system constraints such as power, performance, and bandwidth.

978-3-9819263-2-3/DATE19/©2019 EDAA

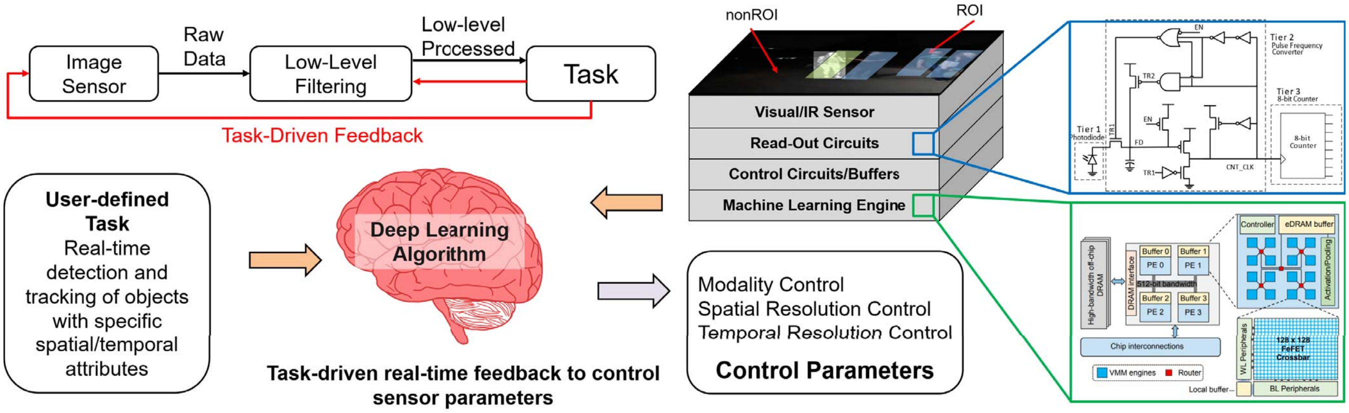


Fig. 1. Our proposed approach integrates feedback directly from a high level task to control acquisition for higher quality and delivery of information. In this paper, we assume the task of object detection and tracking. Our system design of a 3D-stacked image sensor with a broadband photodiode array [5]. The combination of near-sensor processing and per-pixel control exposes various control knobs that can be manipulated by a high level semantic task. Blue Inset: Digital Pixel Circuit consisting of the photodiode, A/D conversion using PFC and Digital counters [8]. Green Inset: Accelerator within the logic layer. An example FeFET accelerator is shown [13].

This paper presents the vision of a ’true’ smart camera that learns how to capture ’useful information’ as defined by a task with highest quality. This is enabled by embedding feedback control within the camera where a deep learning module performing high-level tasks directly controls the sampling characteristics of the pixels. The presented camera model consists of a digital pixel sensor in a 3D-stacked die with a dedicated neural network accelerator (Figure 1). The digital pixel exposes fine-grained control schemes as opposed to coarse-grained or uniform control schemes offered by analog pixel image sensors. A dedicated neural network accelerator tier allows for task-based control of the sensor for higher task performance at the end user with a lower consumption of bandwidth. We discuss camera models that control the spatial and temporal resolution of the camera as well as input modality (visual or infrared) of the pixel. All the presented control schemes are local and thus we only transmit the useful parts of the video feed. It is imperative that ROIC technologies necessary to enable such fine-grain control of the spatial/temporal resolutions are important aspects of designing an adaptive camera, however, this paper focus on overall system architecture and feedback control principles. As an use case, we demonstrate the efficacy of our models on the task of object detection and tracking. It is important to note that, there is a precedence for a vision system where higher level information processing is used to control how a scene is captured; in biology the brain provides feedback signals to the retina to perform task modulation of input singals [14].

II. System Architecture

We base the design of our smart camera with embedded feedback on a 3D stacked sensor topology with a processing-in-memory (PIM) based acceleration tier as shown in Figure 1. The top layer is the digital pixel sensor array which is divided into three tiers with inter-tier communication carried out using high-throughput TSVs and Cu-Cu interconnects. The tiers consist of the photodiode tier, the photocurrent to frequency converter (PFC) tier and the counter tier. We assume a broadband photodiode array presented by Goosens et al. [5] to enable hyperspectral input.

Specialized read-out circuits control various parameters of the sensor including choosing which modality to fetch for each pixel (modality control), perform pixel binning for non-ROI regions (spatial resolution control) and sampling the pixels at different frame rates (temporal resolution control). A logic layer within the stack performs the task processing that guides this per-pixel control. The type of accelerator in the logic layer has implications on the performance of the feedback loop and system performance including power and energy consumption. Software solutions such as mobile-GPUs and dedicated digital or analog accelerators can be used to implement the logic layer due to the heteregenous integration opportunities offered by 3D integration. We explore the impact of the accelerator on the feedback loop further in section IV.

III. Feedback Control

In this section, we will present camera models with controllable parameters driven by DNN algorithms. The close coupling between the controllable pixel and the logic layer allows the DNN to specifically tune the inputs generated by the sensor that results in a higher task performance and a lower consumption of bandwidth. First, we present a camera model that controls the spatial and temporal resolution of RoI regions within an image based on the specifications of the end user. It uses the output of an object detection algorithm to localize and classify objects of interest. The feedback loop then increases/decreases the spatial and temporal resolution of RoI/nonROI regions respectively. The second model that we present combines inputs from multiple modalities (in this case infrared). This model is useful for scenes with varying illumination e.g. low-light scenes or light-cluttered scenes. A periodic switching control policy learns to assign the required modality for specific parts of the image based on the output of an object detection DNN. Early fusion of both modalities is performed and then provided to the object detection DNN for detecting objects of interest. In addition to the control, there are two important components that contribute to the robustness of the feedback loop i.e. Mixed-mode training and ROI prediction. Mixed-mode training of the network performing control is necessary as the input space is variably sampled or composed of mixed modalities. ROI prediction accounts for the evolution of the scene. It offsets and smooths the detected ROIs using a Kalman-Filter based approach.

Spatio-Temporal Resolution Control: We present the concept of spatio-temporal resolution control using an object detection DNN (Figure 2)(a). Our approach is rooted in the concept of region of interest (ROI) based processing. The determined RoIs are fetched at a higher spatial and temporal resolution thatn non-ROI regions of the image. The spatial resolution control scheme is modeled in simulation by averaging out the values of non-ROI regions. In hardware, this process can be implemented by pixel binning. Neighbouring pixel values are averaged out before being fetched by the ROIC. The temporal resolution scheme is similar. We fetch ROI regions at a higher frame rate than non-ROI regions. This is made possible in hardware due to the localized structure of the individual pixel readouts.

Determining the ROI regions for spatio-temporal resolution control depends on the end task or application. We develop a use case for the problem of object detection. The objective of object detection is to classify and localize all objects within an image frame. It is a challenging problem that has only recently made headway with convolutional neural network based detectors [15]. We couple the output of an object detection algorithm within the camera to the ROIC. For each input frame, the algorithm generates a set of detections for objects of interest. These detections are then passed to a ROI prediction algorithm that compensates for the delay between the generated feedback and the acquisition of the next frame. The output of the ROI prediction is then fed to the resolution control module. ROI regions of subsequent frames are then fetched at a higher spatio-temporal resolution. If new objects of interest enter non-ROI regions of the frame then they may not be detected. We fine-tune our networks on subsampled inputs at various resolutions to solve this.

Applied to the task of object detection on a running sequence from the NFS dataset which contains high frame rate videos of actions sampled at 240 fps [16]. They also generate sub-sampled versions of the videos at 30 fps and manually add motion blur for the sub-sampled videos. We designate the person class as the object of interest. Qualitative results are shown in Figure 3. Sampling the ROI regions at a higher spatio-temporal resolution results in a higher detection accuracy and reduction of motion blur. In addition to the accuracy increase we achieve 27x reduction in the bandwidth consumed (from 1592 Mbps to 57.6 Mbps). Further reduction can be achieved if it is coupled with a MJPEG compression engine for off-camera transmission.

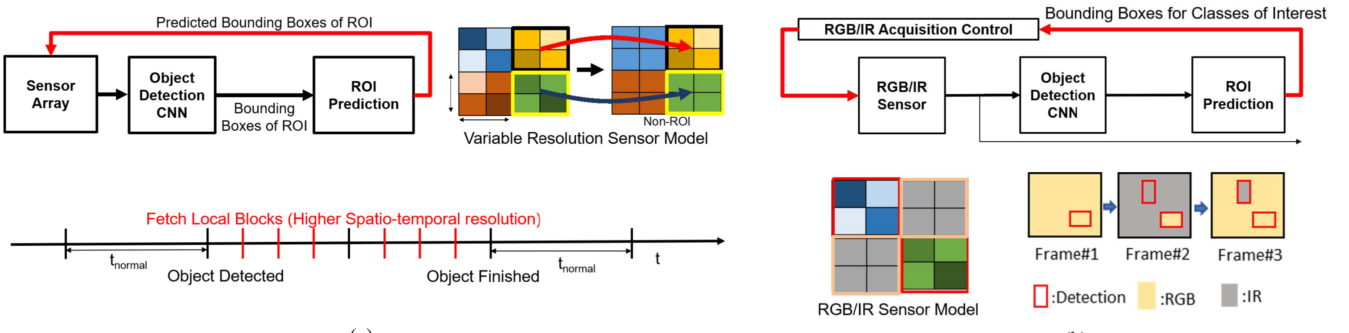
TABLE I
DETECTION AND TRACKING ACCURACY AND BANDWIDTH CONSUMED ON THE CAMEL DATASET [18] WITH FEEDBACK CONTROL

	Object Detection (mAP)	Object Tracking (MOTA)	Bandwidth (Mbps)
Visual	0.509	0.384	62.09
SpatioTemporal	0.512	0.403	20.58
Mixed-modality	**0.586**	**0.480**	**17.88**

Multi-Modality Control: Similar to human eyes, visual spectrum image sensors are only capable to capture relevant information in well-illuminated environment. Resultantly, object detection system based on input from visible spectrum sensor shows poor performance under inadequate lighting condition particularly at night time. Combining information from visual and IR imaging sensors is a widely used solution for round-the-clock applications. Following the success of deep neural network (DNN) in object detection task, several multi-modal DNN architectures have been proposed. These architectures fuse the individual visual pipeline and IR pipeline at different layers of the network. These feature level fusion architectures significantly improve the object detection accuracy compared to a single pipeline, but complexity of the network increases exponentially as the number of modalities (for different spectra) increases. Saha et al., investigate a different approach where the inputs from different modalities are fused to create a mixed-modality image that can be operated by a single DNN [17]. Their experiment involves only visual and far-IR images, however, the proposed method can be extended easily to incorporate images from other spectra.

Data-driven feedback from object detection network is used to control the spatial modality of the scene (Figure 2(b)). Detections on current frame determine the local modality of the next frame. For the very first frame of a sequence, we pass the visual modality image through the object detection network and mark the detected bounding boxes with confidence score above certain threshold as RoIs. Assuming objects are not moving very fast, we expect that retaining the modality of RoIs will ensure their detectability in the next frame. Therefore, we keep the modality of RoIs same (i.e. visual) in the next frame and switch the modality of nonRoIs to IR (Figure. 5) to find out any new object. In this fashion, we keep track of modality for each RoI and ensure they get detected in the next frame while altering nonRoI modality in search of new objects. Mixed-modality image improves detection accuracy (mean average precision) by 7.7% on the CAMEL Dataset [18] and tracking accuracy by 9.6%. (Table I).

Mixed Mode Training: A key factor for robust operation of the above mentioned control schemes is that the input space is either variably sampled (in case of spatiotemporal resolution control) or is composed of various modalities (multi-modality control). This poses problems for new objects that enter the scene (birth) and may go undetected if the detection networks are not fine-tuned. We solve this problem with data augmentation by partitioning our dataset equally with images sampled at



(a)

(b)

Fig. 2. (a) Block diagram for control of spatiotemporal resolution using the output of an object detection algorithm with a sensor model used to simulate the variable resolution control and the Spatio-temporal resolution control policy. Local ROI blocks are fetched at a higher spatial and temporal resolution (b) Block diagram for multi-modality feedback control. Task information is used to decide the modality for each individual pixel with a Multi-modality Sensor Model and the Mixed-modality image creation algorithm. Inputs to the object detection network for three consecutive frames are presented.

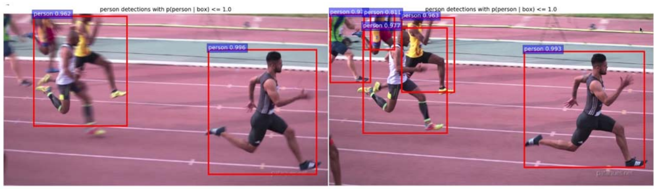


Fig. 3. Spatiotemporal Resolution Detection applied to the problem of Object Detection. Motion blurring is diminished with the feedback control and object detection accuracy is increased. **Left**: Normal detections at 30 fps **Right**: ROI detections at 240 fps.

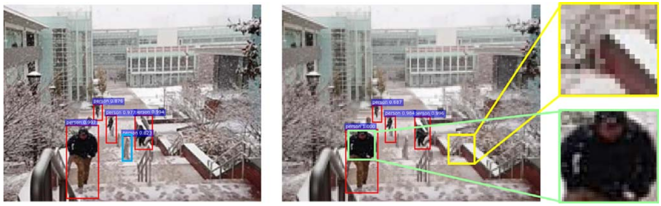


Fig. 4. Application of spatial resolution control to the problem of object detection. Note the elimination of the false positive highlighted in blue when feedback control is applied.. In the right panel the insets show the difference between non-ROI and ROI regions. Yellow Inset: Non-ROI region, Green Inset: ROI Region

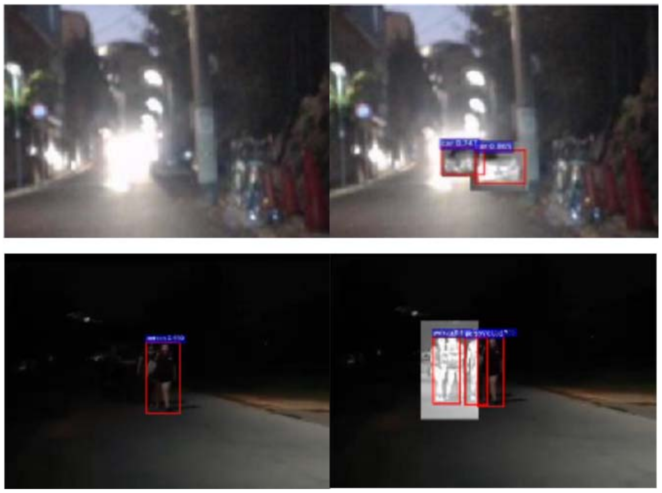


Fig. 5. Use cases for multiple visual modalities and application of modality control. Application of multi-modality control to **Top**: Light-cluttered scenes where a light sources completely saturates the visible spectrum reducing visibility of the scene and **Bottom**: Insufficient Illumination within the scene. Exposure time and gain has to be increased which leads to motion blur and noisy images. Multi-modality control allows us to switch the pixel modality while keeping track of the objects of interest.

various resolutions. 3 splits are created from the MS COCO dataset with varying resolutions (Original/2x/4x). In case of mixed-modality control we add IR images to the training set in addition to the original RGB images. We also experimented with using ground-truth boxes to create mixed-mode images manually for the training process but that did not yield any significant improvement over plain data augmentation.

ROI Prediction: Evolution of the scene due to object movements and finite latency between feedback generation and application can potentially cause feedback to go ”stale”. Additionally, frequently occurring mis-detections can cause unstable feedback performance. A ROI prediction module is inserted to address these challenges. We adopt concepts from object tracking literature and apply the SORT approach for ROI prediction [19]. The SORT algorithm uses an Hungarian algorithm-based association module to associate new and past detections. Kalman Filter-based predictors estimate and predict the coordinates of objects using a linear motion model. Any predictors that are not updated with new detections are removed after a certain number of frames. For completeley new detections, a new predictor is created.

IV. Computational Complexity

In this section, we will discuss the challenges created as a result of integrating feedback in to the camera pipeline. Computational complexity of the control algorithms is a critical challenge due to the finite latency between the generation of feedback and its application. State of the art object detectors rely on complex convolutional backbones composed of many

TABLE II
SYSTEM ACCURACY WITH HARDWARE CONSTRAINTS AND SOFTWARE TECHNIQUES FOR OPTIMIZATION

	Optimization	Latency (s)	Detection (mAP)	Tracking (MOTA)
GPU (300 W)	-	-	0.586	0.480
mGPU (5 W)	-	0.504	0.512	0.092
	Quantization (FP16)	0.252	0.518	0.294
	Pruning	0.403	0.259	0.005
	Simpler Detector	0.032	0.327	0.253

convolution layers to perform feature extraction. For the feedback models presented in the preceding section, we used a state of the art detector RFCN with a complex convolutional backbone ResNet-101. The RFCN detector is a two-stage detector with a backbone, a region proposal network (RPN) followed by ROI pooling and a few layers for classification. The RPN generates class-agnostic object candidates. In total, 129 GFLOPs of computation are required to perform one forward pass.

As mentioned in Section II, different types of acceleration can be implemented in the logic layer. For our initial analysis we estimate the latency of feedback application using peak throughput offered by a low-power mobile GPU platform (Tegra X1). The Tegra X1 can deliver about 256 GFLOPS/s of peak throughput at 5 W. We compare the performance with a high-performance GPU (GTX 1080Ti, 11 TFLOPS/s). Applying hardware constraints to the feedback models presented earlier we observe that the performance of the closed loop feedback system drops considerably as shown in Table II. We extend our analysis by breaking down the complexity of the RFCN ResNet-101 detector on a per-module basis. Our analysis shows that the backbone (95.49 FLOPS) comprises over 75% of the compute complexity. To address this challenge, a multitude of hardware and software optimizations can be applied. Software approaches consist mainly of model reduction techniques and low bit precision inference. Hardware approaches include specific domain-based accelerator ASICs or neuromorphic architectures making use of novel devices such as ReRAM [20] or FeFET [13].

A. *Software Approaches to Complexity Reduction*

In this work, we explore three different approaches to reduce the network complexity. Our experiments indicate that compared with image classification, object detection networks tends to be more sensitive to these complexity reduction techniques and require more efforts for fine-tuning.

Simpler Detector: An alternative is to use a simpler detector that is optimized for mobile applications. We choose the SSD Mobilenet v1 architecture that has 13 layers of convolution (1.2 GFLOPS of computation; 6.54 M parameters). Shifting to the simpler network allows for real-time processing at the cost of task performance. On the CAMEL dataset we are able to recover 16.1% of the tracking accuracy. Note that open loop performance (mAP) is lower due to the simpler detector.

mAP@0.5	Float	32-bit	16-bit	8-bit	4-bit
Average	61.3	61.3	61.3	54.2	0.001
Car	63.5	63.5	62.5	58.0	0.0
Truck	55.5	55.5	59.9	51.0	0.001
Person	75.4	75.4	75.2	72.1	0.004
Stop sign	86.7	86.7	81.8	84.1	0.0

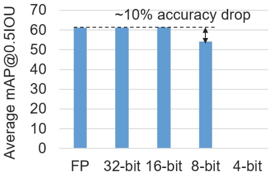


Fig. 6. Accuracy (mAP@0.5) with different bit-precision.

Quantization: By reducing the bit-precision of weights, activations, and gradients, both the computing complexity as well as the memory footprint can be reduced. Image classification networks show remarkable resilience to half precision or even ternary/binary precision [21]. However, interestingly, our experiment indicates that object detection DNN models (such as RFCN and Faster-RCNN) show less robustness towards quantization. In our experiment, we quantize the first-stage feature extractor of RFCN while keeping the back-end RPN/ROI predictor in floating point precision. As shown in Figure 6, after re-training, the accuracy (in terms of mAP) drops $\sim 10\%$ with 8-bit precision. With 16-bit quantization, object detection performance is similar to full precision. The Tegra platform allows for packing of FP16 operations effectively doubling the peak throughput. We are able to recover 20.2% of the tracking accuracy using this approach.

Network pruning: With network pruning, a portion of synapses and neurons can be eliminated from the model based on a ranking criteria. Ranking criteria are diverse including magnitude-based rankings [22], error sensitivity based [23] or a combination of both [24]. We explore to prune the network with a user-defined specialized class set according to the requirements of the end user. The intuition behind this approach being that the multiple features being learnt for a diverse set of classes (1000 in ImageNet, 90 in MS COCO) may not be necessary for a specialized class set. We apply this approach in conjunction with pruning. A new classifier is inserted with specialized set of classes and retrained on the target dataset. During training, gradient and activation probabilities are used to create a ranking [24]. All filters over the entire backbone are ranked and 20% of the total filters are removed. The performance of the detector drops by more than 50% even without the feedback loop ($51.2\% \rightarrow 25.9\%$). *Well known pruning techniques for image classification do not seem to apply well to object detection but further investigation is necessary.*

B. *Domain-Specific Accelerators*

With the rise of deep learning techniques, dedicated hardware accelerators have been extensively researched to take advantage of the specific structure of DNNs to perform efficient inference/training in a low power budget. Deeptrain uses the Processing in Memory (PIM) paradigm in a Hybrid Memory Cube (HMC) for accelerated training and inference [25]. DeepTrain has dedicated DRAM layers within the 3D stack eliminating off-chip DRAM accesses. DNPU uses a LUT-based multiplier fabric with on-chip quantization table (Q-

TABLE III
SYSTEM ACCURACY ON DIFFERENT HARDWARE PLATFORMS

Platform	Latency (s)	Frame Rate	AP (person)	MOTA
GPU (300 W)	0.032	30	0.586	0.480
Tegra X1 (5 W)	0.504	1.9	0.512	0.092
ReRAM [27](2.5 W)	0.135	7.3	0.586	0.446
FeFET [13] (2.27 W)	0.128	7.9	0.586	0.446
DNPU [26] (2.5 W)	0.065	15.5	0.586	0.480
DeepTrain [25] (2.6 W)	0.032	30	0.586	0.480

table) to speed up mulitplications [26]. Beside ASIC, there are efforts in exploiting emerging non-volatile memory (NVM), in particular, resistive random-access memory (ReRAM) and ferroelectric FET (FeFET), for DNN acceleration [13], [20], [27]. The key idea behind the ReRAM/FeFET based accelerator is utilizing crossbar array to perform vector-matrix multiplication (VMM), which is the major type of computation for DNN. It has been demonstrated that ReRAM/FeFET based DNN accelerators promise much higher computing efficiency than the CPUs/GPUs [13], [20]. We again project the performance of our feedback loop by computing the latency for different hardware platforms as shown in Table III. All projections are made assuming RFCN ResNet-101 as the detector.

Under similar power budgets, the specialized accelerators are able to recover the performance of the feedback loop compared to a general purpose mGPU accelerator. The peak throughput offered by the Deeptrain (4 TOPS) accelerator allows for a maximum processing throughput of 62 fps. For videos containing high-speed targets (such as the NFS videos), this processing speed is 4x short of the required target. Hence, additional innovations are required at the accelerator level as well to fully realize our adaptive camera.

V. Conclusions and Future Work

This paper represents a step in the direction of realizing a smart camera with task guided parameter control but some issues of computational complexity and energy efficiency remain. Accurate networks are needed to realize the full potential of the feedback loop but are limited by their computational complexity. We have explored software model reduction techniques and domain-specific accelerators to address this limitation in Section IV.

From a task and learning perspective, a prospective direction would be dealing with uncertainty and out-of-distribution inputs. This will ultimately require integrating bayesian approaches, in-field learning and fast reconfiguration of the camera. Integration of in-camera network training will require further hardware/software innovations to address the computational requirements of training (floating-point compute, 3x memory to store gradients and activations).

VI. Acknowledgement

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] A. El Gamal, "Digital pixel image sensors," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 2049–59, 2001.

[2] O. Skorka and D. Joseph, "Cmos digital pixel sensors: technology and applications," in *SPIE*, 2014.

[3] K. I. Schultz, M. W. Kelly *et al.*, "Digital-pixel focal plane array technology," *Lincoln Laboratory Journal*, vol. 20, no. 2, pp. 36–51, 2014.

[4] D. Bol, G. de Streel *et al.*, "A 65-nm 0.5-v 17-pj/frame. pixel dps cmos image sensor for ultra-low-power socs achieving 40-db dynamic range," in *VLSI Circuits Digest of Technical Papers, 2014 Symposium on*. IEEE, 2014, pp. 1–2.

[5] S. Goossens, G. Navickaite *et al.*, "Broadband image sensor array based on graphene–cmos integration," *Nature Photonics*, vol. 11, no. 6, p. 366, 2017.

[6] T. Haruta, T. Nakajima *et al.*, "4.6 a 1/2.3 inch 20mpixel 3-layer stacked cmos image sensor with dram," in *ISSCC*, 2017.

[7] S. Mukhopodhyay, M. Wolf *et al.*, "The camel approach to stacked sensor smart cameras," in *DATE*, 2018.

[8] M. Amir and S. Mukhopadhyay, "3d stacked high throughput pixel parallel image sensor with integrated reram based neural accelerator," in *S3S*, 2016.

[9] B. Rinner and W. Wolf, "An introduction to distributed smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1565–1575, 2008.

[10] J. H. Ko, B. A. Mudassar, and S. Mukhopadhyay, "An energy-efficient wireless video sensor node for moving object surveillance," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 1, no. 1, pp. 7–18, 2015.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[12] M. Amir, D. Kim *et al.*, "Neurosensor: A 3d image sensor with integrated neural accelerator," in *S3S*, 2016.

[13] Y. Long, T. Na *et al.*, "A ferroelectric fet based power-efficient architecture for data-intensive computing," in *ICCAD*, 2018.

[14] J. Bullier, "What is fed back?" *23 Problems in Systems Neuroscience*, p. 103, 2005.

[15] J. Huang, V. Rathod *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *CVPR*, 2017.

[16] H. K. Galoogahi, A. Fagg *et al.*, "Need for speed: A benchmark for higher frame rate object tracking," in *ICCV*, 2017.

[17] P. Saha, B. Mudassar, and S. Mukhopadhyay, "Adaptive control of camera modality with deep neural network-based feedback for efficient object tracking," in *AVSS*, 2018.

[18] E. Gebhardt and M. Wolf, "Camel dataset for visual and thermal infrared multiple object detection and tracking," in *AVSS*, 2018.

[19] A. Bewley, Z. Ge *et al.*, "Simple online and realtime tracking," in *ICIP*, 2016.

[20] Y. Long, T. Na, and S. Mukhopadhyay, "Reram-based processing-in-memory architecture for recurrent neural network acceleration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, no. 99, pp. 1–14, 2018.

[21] I. Hubara, M. Courbariaux *et al.*, "Binarized neural networks," in *NeurIPS*, 2016.

[22] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[23] J. H. Ko, D. Kim *et al.*, "Adaptive weight compression for memory-efficient neural networks," in *DATE*, 2017.

[24] P. Molchanov, S. Tyree *et al.*, "Pruning convolutional neural networks for resource efficient inference," in *ICLR*, 2017.

[25] D. Kim, T. Na *et al.*, "Deeptrain: A programmable embedded platform for training deep neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2360–2370, 2018.

[26] D. Shin, J. Lee *et al.*, "14.2 dnpu: An 8.1 tops/w reconfigurable cnn-rnn processor for general-purpose deep neural networks," in *ISSCC*, 2017.

[27] A. Shafiee, A. Nag *et al.*, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *ISCA*, 2016.