Domain-Specific Approximation for Object Detection

Ting-Wu Chin
National Chiao Tung University

Chia-Lin Yu
National Chiao Tung University

Matthew Halpern
University of Texas at Austin

Hasan Genc
The University of Texas at Austin

Shiao-Li Tsao
National Chiao Tung University

Vijay Janapa Reddi
University of Texas at Austin

There is growing interest in object detection in advanced driver assistance systems and autonomous robots and vehicles. To enable such innovative systems, we need faster object detection. In this article, we investigate the trade-off between accuracy and speed with domain-specific approximations (DSAs) for two state-of-the-art deep learning-based object detection meta-architectures. We study the effectiveness of applying approximation both statically and dynamically to understand their potential and applicability. By conducting experiments on the ImageNet VID dataset, we show that DSA has great potential to improve the speed of the system without deteriorating the accuracy of object detectors. To this end, we present our insights on harvesting DSA and devise a proof-of-concept runtime, AutoFocus, that exploits dynamic DSA.

With rapid progress being made in the field of computer vision and machine learning, there is growing interest in the practical deployment of intelligent algorithms in systems, such as autonomous vehicles, autonomous robots, and advanced driver assistance systems. For many of these "intelligent" systems, detection is one of the fundamental algorithms involved in developing end-to-end applications. Detection can lead to obstacle recognition, avoidance, and navigation. As a result, detection is becoming an important algorithm for developing cognitive visual agents.

Thus far, the majority of effort on object detection has been focused on achieving high accuracy. However, from a system's perspective, object detection speed also matters. For instance, how

Published by the IEEE Computer Society
0272-1732/18/$33.00 ©2018 IEEE

fast an autonomous car or drone can move depends on how fast the detection algorithms run on the underlying hardware and software substrate. Moreover, to build robust, intelligent agents, there will likely be more than one cognitive algorithm running on the system at the same time.

Therefore, the throughput of the system ought to be one of the top concerns for system designers in the near future. In our experiments, modern deep learning-based object detectors, without modification, could only provide up to 2 frames per second (FPS) on NVIDIA's Tegra X1 with 640x480 images. Typical real-time performance requires close to 30 FPS. We need faster hardware to ensure real-time performance while several tasks are running simultaneously and to cope with future high-resolution (such as 2048x1024) images. Alternatively, we need software techniques that can improve performance and energy efficiency on existing hardware.

In this article, we determine that category-awareness introduces new opportunities for domain-specific approximation (DSA) to augment hardware and software advancements. Some categories are almost always larger than others (for example, trains are usually larger than motorcycles), which results in different sensitivity toward approximation techniques. We explore the speed and accuracy trade-offs brought by two software-driven DSA optimization techniques: image size and the number of region proposals. Both of these techniques affect speed and accuracy.[1] Image size affects how much detail is covered within the receptive field of view—the region used to classify objects. Small images have more information crowded into a fixed-size receptive field; thus, there are fewer regions that need examination, resulting in less accuracy but faster detection. Region proposals are the regions that are likely to contain objects. In object detection, algorithms that propose object regions often act as a filter that let regions with high probability of containing objects pass through them for further processing (classifying where and what the object is within that region). Hence, reducing the number of region proposals reduces the total candidate regions that need examination, resulting in less accuracy but faster detection.

We analyze category-aware DSA along two settings: static and dynamic. In the static case, we use one approximation configuration for each category. An approximation configuration refers to a tuple involving an image size and a region-proposal size. We determine this tuple using offline profiling. In the dynamic case, we select an approximation configuration per image, albeit still within a category. We approximate the input frame dynamically by extracting useful features from previous frames and rely on a linear model to infer the next approximation configuration.

In both cases, our results indicate that category-awareness is promising in terms of speed improvement compared to the category-oblivious DSA discussed in prior work.[1] In an oracle scheme, category-aware static and dynamic DSA achieve 3.7× and 7.5× speedup without accuracy degradation for a certain object category. We also discuss the limitations and challenges of implementing both static and dynamic DSA in a runtime and provide our insights on harnessing the potential of both static and dynamic DSA. The runtime we design approximates the input frame on the fly by extracting useful features from previously seen frames and uses a linear model to infer the next setting. Our results show up to 51- and 211-percent speed improvement with 22- and 41-percent accuracy degradation for Faster R-CNN and R-FCN, respectively.

Our findings suggest that future system designers of cognitive visual agents can improve their speed and energy-efficiency by using knowledge of what target object categories matter most. For example, an autonomous vehicle's system designer might want the vehicle's object detector(s) to work well on several crucial categories, such as pedestrians, bicyclists, and cars. One way to improve the speed of such a system would be to try to find the image size that has the fastest speed but does not compromise accuracy across the three categories and deploy the system with the optimized image size. Alternatively, the designer could optimize the image size of the target categories and dynamically choose the corresponding image size when those objects show up in their perspective or field of view, leading to better performance.

In summary, our contributions are as follows:

- We investigate DSA and characterize the effectiveness of category-awareness.
- We conduct a limit study to understand the benefit of applying approximation in a per-frame manner with category-awareness (category-aware dynamic DSA).
- We present the challenges of harnessing DSA and a proof-of-concept runtime.

EXPERIMENTAL SETUP

We conduct our analysis and optimization on two modern meta-architectures, Faster R-CNN[2] and R-FCN[3], for deep learning-based object detection on a state-of-the-art embedded system.

Object Detectors

The feature extractors we use are the ones that come with the repository of the object detector, and the weights we use are the weights that were pre-trained on the PASCAL VOC dataset[4] by the original authors. We use the ZF network[5] for Faster R-CNN and ResNet-50[6] for R-FCN.

We also considered including both the SSD[7] and YOLO[8] network in our study. However, neither of them is amenable to input image scaling. The network demands a fixed size input. Both of them scale the image into a fixed size regardless of the original image size, and it performs poorly if the image resolution is large and contains a lot of small objects. As the camera resolutions continue to grow larger, we envision more and more applications will be based on more flexible meta-architectures, such as Faster R-CNN and R-FCN.

Dataset and Metric

Dataset. We picked 40 videos from ImageNet Object Detection from Videos (VID) task dataset for each of the categories we studied and divided them into 20 training videos and 20 testing videos. Specifically, among the categories that overlapped with the PASCAL VOC dataset, we randomly picked eight categories for our analysis, which include the following: airplane, bus, car, dog, horse, motorcycle, sheep, and train. Notice that we picked categories that overlapped with the PASCAL VOC dataset, so that we could use the pre-trained models provided by authors without re-training.

Metric. As in most of the object-detection evaluation, we counted on average precision or mean average precision to evaluate the accuracy of the object detector. We leveraged the MS-COCO toolkit[9] and modified it to calculate per-image average precision to determine the optimal approximation configuration in a per-image granularity.

Platform

Hardware. We conducted all of our experiments on the SoC Tegra X1, which, at the time of this writing, is a state-of-the-art embedded SoC that has a Maxwell GPU with 256 CUDA-core and a quad-core ARM A57 CPU. Unless stated otherwise, the frequency of the CPU and GPU are kept constant at the highest speed (1912.5 MHz for the CPU and 998.4 MHz for the GPU).

Software. We used CUDA 7.0 and cuDNN 4 for GPGPU processing. We used the Caffe[10] deep-learning framework to power all the algorithms we studied.

DSA

In this section, we first describe related work in DSA and the terms used throughout the paper. Then, we analyze the effectiveness of category-aware static and dynamic DSA. Though we focus on performance, our results can also be applied to improving energy efficiency.

State-of-the-Art

Approximation has been discussed in many recent works.[11–13] However, DSA is still an emerging concept[14] and bound to specific domains.[15] For object detection, the most relevant work is by Huang et al.[1] and discusses trade-offs between speed and accuracy for different image sizes, the number of region proposals, and feature extractors in the object detectors.

Our article focuses on category-aware static and dynamic DSA. We suggest that category-awareness introduces more opportunities in DSA than category-oblivious DSA.[1] Additionally, we investigate the applicability of harnessing those opportunities with both static and dynamic DSA.

Terminology and Definitions

Downsampling. To set a proper baseline, we first scale every image in the dataset to a fixed height of 480 pixels. For approximating the image size, we evaluate 11 different downsampling levels, starting from 480 pixels. At each level, we downsample the image by 40 pixels, progressively scaling the image all the way down until we reach 80 pixels in height.

Region Proposal. We choose 300 as the baseline. Both Faster R-CNN and R-FCN use the same value by default. Also, we explore a different number of proposals (300, 200, 100, 50, and 10) as a means to reduce processing requirements without impacting accuracy.

Region of Interest. We define a region of interest (ROI) as the bounding box surrounding the object of interest in an image.

Approximation Configuration. We denote an approximation level as a configuration tuple. For example, (160, 50) denotes 160 in image height and 50 region proposals. We denote marginal approximation by leaving the other as a hyphen. For example, (360, -) means 360 in image height while using the baseline for the number of proposals (300 in our study).

Safe Approximation. We focus only on approximations that do not deteriorate the average precision. When we present results for the oracle approximation scheme, we select the approximation level that achieves equal or greater average precision compared to the baseline.

Optimal Approximation. We define this as the fastest configuration (maximum FPS) without accuracy degradation (so it is still considered a safe approximation).

Category-Aware DSA Opportunity

Some categories, such as trains (see Figure 1(a)), are almost always larger than other categories, such as motorcycles (see Figure 1(b)). What this implies is that, in terms of accuracy, some categories almost always have lower sensitivity to approximation than others, which motivates us to investigate the benefits of bringing category-awareness to the DSA techniques we study.

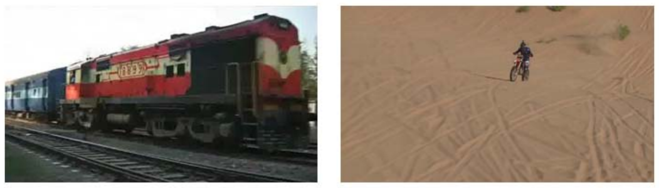
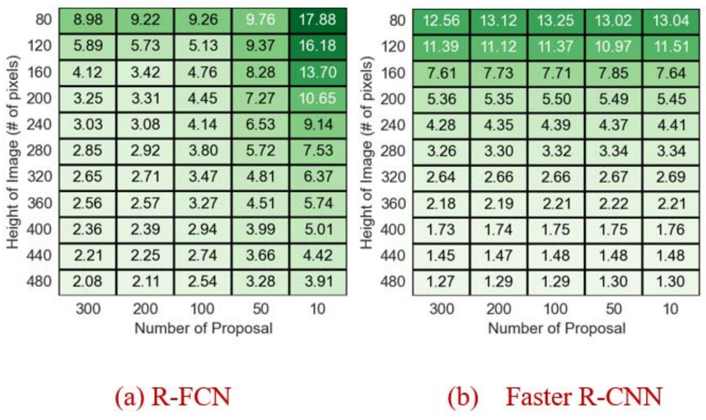

(a) Train (b) Motorcycle

Figure 1. Inter-category variation encourages category-aware DSA.

Performance/Throughput

Before we analyzed a category's sensitivity to the DSA techniques, we first investigated the performance/throughput benefit brought by the techniques. Figure 2 shows an evaluation of how the number of FPS (performance/throughput) is affected by the two techniques. First, we found that for Faster R-CNN, due to the fully connected layers applied to each of the region proposals, its speed is related to proposal scaling. For R-FCN, it is somewhat invariant to proposal scaling because the computation after ROI-pooling layer is small compared to the convolution layers before ROI-pooling. Second, both meta-architectures are sensitive to image scaling. Due to R-FCN's fully convolutional design, it is more sensitive than Faster R-CNN. By jointly investigating both techniques, we found that Faster R-CNN can have better speed gain leveraging image scaling when the number of proposals is small. This is reasonable because fully connected layers for each proposal imposes some computation overhead, and it is invariant to image size.



(a) R-FCN

Height of Image (# of pixels)	300	200	100	50	10
80	8.98	9.22	9.26	9.76	17.88
120	5.89	5.73	5.13	9.37	16.18
160	4.12	3.42	4.76	8.28	13.70
200	3.25	3.31	4.45	7.27	10.65
240	3.03	3.08	4.14	6.53	9.14
280	2.85	2.92	3.80	5.72	7.53
320	2.65	2.71	3.47	4.81	6.37
360	2.56	2.57	3.27	4.51	5.74
400	2.36	2.39	2.94	3.99	5.01
440	2.21	2.25	2.74	3.66	4.42
480	2.08	2.11	2.54	3.28	3.91

(b) Faster R-CNN

Height of Image (# of pixels)	300	200	100	50	10
80	12.56	13.12	13.25	13.02	13.04
120	11.39	11.12	11.37	10.97	11.51
160	7.61	7.73	7.71	7.85	7.64
200	5.36	5.35	5.50	5.49	5.45
240	4.28	4.35	4.39	4.37	4.41
280	3.26	3.30	3.32	3.34	3.34
320	2.64	2.66	2.66	2.67	2.69
360	2.18	2.19	2.21	2.22	2.21
400	1.73	1.74	1.75	1.75	1.76
440	1.45	1.47	1.48	1.48	1.48
480	1.27	1.29	1.29	1.30	1.30

Figure 2. Throughput in FPS for object detection algorithms under two kinds of DSA techniques.

Similar approximation strategies have been explored in the past, but in a more limited capacity. Huang et al.[1] investigate what knobs generally affect the speed and accuracy of the object detectors, while our analysis focuses more on image scaling and proposal scaling and is more comprehensive and insightful in the following manner. First, we analyze image sizes at a finer granularity, from 80 pixels to 480 pixels in height with a 40-pixel step. Second, our evaluation considers both techniques jointly, which introduces more approximation configurations with similar performance improvement for architectures that are sensitive to both techniques (Faster R-CNN). For example, Figure 2(b) shows that approximation configurations (80, 300), (160, 50), and (240, 10) share similar performance improvement, which enables the system to optimize for accuracy under the same performance improvement.

In terms of accuracy, Figure 3 indicates that Faster R-CNN is more sensitive to region proposal scaling compared to R-FCN. Moreover, R-FCN has a sharper boundary compared to Faster R-CNN, which suggests that Faster R-CNN is more robust when it comes to the two DSA techniques. Interestingly, a different category has different sensitivity toward both DSA techniques. Airplane (Figure 3(a)) has more "safe approximation" configurations than train (Figure 3(b)), which has more safe approximation configurations than motorcycle (Figure 3(c)).

Inter-Category Variance

Given the performance improvement motivation of the two DSA techniques, we next analyzed the accuracy of the object detector for each category we studied. We refer to this study as "inter-category variance" because here we are focused on the variance that stems from differences across categories, and not within different images that all correspond to the same category.

One of the reasons behind the variance is intuitive: that the larger the object, the smaller an approximation can go without accuracy degradation. In our results, the average object size in pixels in the dataset we use for airplane, train, and motorcycle are 107k, 126k, and 38k, respectively.

However, inter-category variation is not solely due to the size of the objects; it is also due to the noise in the background (for example, the background for the airplane class has less noise than others) and the fact that some categories are harder for the object detector than others.[16] This implies that the hard classes are more sensitive to information loss (approximation). The variation of the safe approximation across category encourages category-awareness for even faster system design.

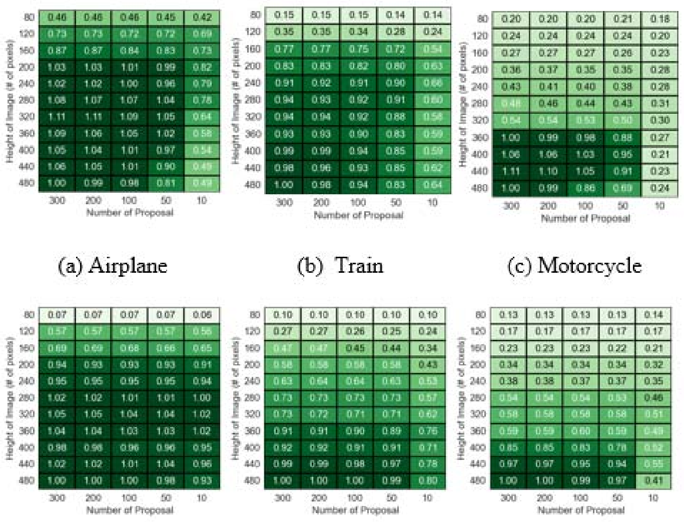


Figure 3. Normalized average precision of Faster R-CNN (top row) and R-FCN (bottom row) for three selected categories under two kinds of DSA techniques.

Intra-Category Variance

Beyond the benefit introduced by category-awareness, we observe that there is even greater intra-category variance for DSA techniques to exploit on a per-frame basis in an input stream.

In a dynamic input stream (a video, for example), the optimal approximation can change continuously. Objects in a dynamic input stream are constantly changing in perspective size (objects are sometimes near and large, and at other times small and far). Figure 4 shows a train passing by from left to right. As time goes by, the train gets larger; the level of safe approximation can become more aggressive over time. Therefore, dynamically picking an approximation configuration can sometimes be better than statically picking a one-time fixed DSA configuration.

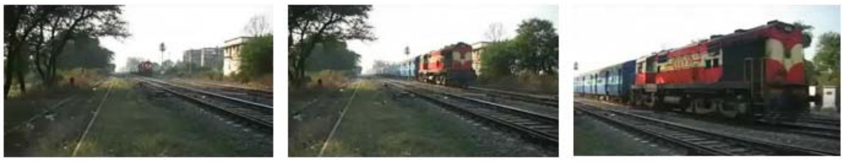
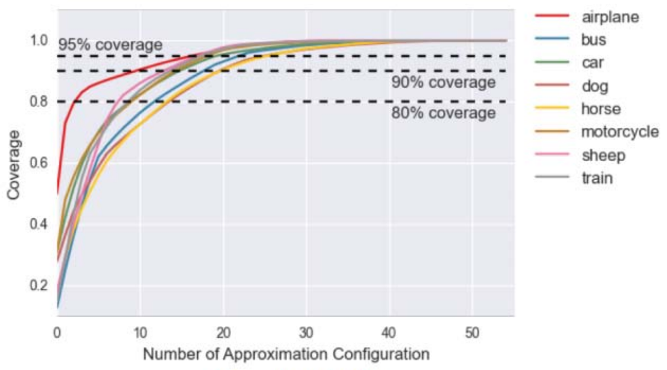


Figure 4. Intra-category variation encourages dynamic DSA.

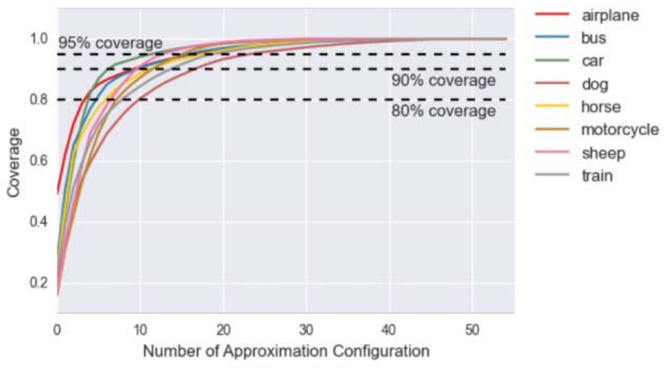
To motivate "dynamic DSA," Figure 5 shows the relationship between the portion of images of the training data that are approximated optimally and the corresponding number of approximation configurations considered for the eight categories that we mentioned previously. For example, we need 18 unique approximation configurations, as shown in Figure 5(a), to optimally approximate 95 percent of the airplane images from the training dataset.

The analysis hints at whether a category is worthy of dynamic DSA. With static DSA, the system applies only one approximation configuration for the entirety of each category. However, for the most extreme case, such as the dog category in Figure 5(b), we require 25 approximation configurations to optimally approximate 95 percent of the dogs. This means that static DSA loses significant opportunity to improve performance. On the other hand, categories such as sheep require the least number of approximation configurations to achieve 95 percent coverage, which might have better performance with static DSA.

Dynamic DSA can speed up Faster R-CNN by up to 7.5x and R-FCN up to 7x without accuracy loss for some categories. So, if we can model the optimal approximation correctly with a simple model, there is an opportunity to speed up object detection-based systems significantly.



(a) Faster R-CNN



(b) R-FCN

Figure 5. The relationship between the portion of images of the training data that are approximated optimally, and the corresponding number of approximation configurations considered.

DSA CHALLENGES TOWARD IMPLEMENTATION

Our study shows promise in category-aware DSA for object detectors. However, there are challenges that remain to be resolved to harness the benefits. Here, we first illustrate the challenges for implementing category-aware DSA and then provide our insights and implementation.

Challenges

We identify two major challenges that we believe to be the key to unlocking the potential for category-aware DSA.

Prior Knowledge. Category-awareness exposes extra "slack" for DSA techniques to exploit. However, without performing object detection in the first place, it is unclear how the system can obtain the information regarding the category of the object to determine the approximation configuration. So, a key challenge is how to approximate the category information to begin with.

Dynamism Modeling. To further harness dynamic DSA, it is essential to understand what causes different optimal approximation configurations. It is critical to understand what the good features are to predict the optimal approximation. Moreover, to benefit from speed improvement, the model for inferencing optimal approximation should be low-overhead.

Implementation

We implement dynamic DSA on streaming inputs, which is common in autonomous agents. In the streaming input case, there is temporal consistency (the difference between any two consecutive frames is small), which we can leverage to overcome the aforementioned challenges.

Prior Knowledge. With temporal consistency, we can resolve the first challenge approximately in the sense that we regard the category of the objects in the current frame to be the output of the object detector from the previous frames. Notice that by approximating the prior knowledge this way, we introduce two new problems. 1) How many incoming frames can we assume to be the same as the current frame? 2) How much can we believe in the output of the object detector? For the first problem, we try one, three, and five frames and find three to be better than the others in the dataset we use. For the second problem, we count on the output probability (the output of the softmax layer) to tell if we can trust the ROIs. We set a threshold to 0.6 in our implementation.

Dynamism Modeling. We assume the approximation decision is correlated to the size and the number of objects in the frame. Intuitively, when the object is large, we might be able to downsample it more than when it is small. On the other hand, the number of objects affect occlusion. If there are more objects in the frame, it is more likely that there will be a lot of overlapping. Additionally, given a fixed-size input, the more objects there are in the frame, the smaller those objects can be. Hence, we build an ordered four polynomial regressor with the height and width of both the smallest and largest ROI in the frame and the number of ROIs in the frame.

Evaluation

We refer to our runtime system implementation as AutoFocus. We first evaluate the overhead introduced by AutoFocus. According to our measurements, on average, the inferencing approximation configuration using the ordered four polynomial regressor takes 3.2 ms, which is only 0.6 percent of the original Faster R-CNN object detector's performance overhead.

We compare AutoFocus against static DSA with prior knowledge and category-oblivious DSA (from prior work). To obtain results for the static DSA with prior knowledge, we obtain the optimal configuration for each category on the training set and use that configuration on the test set, given that we know the category in advance. For AutoFocus, we train the polynomial model on the training set and apply the model on the test set. Figure 6 shows our evaluation for category-oblivious DSA, static DSA, and AutoFocus. We first focus on the benefits brought by static DSA compared to category-oblivious DSA. For categories like buses, cars, and motorcycles, static DSA improves the speed by 20 percent on average with merely 3 percent accuracy degradation, which exhibits the good part of static DSA. On the other hand, for categories like airplanes and dogs, there is a large margin in speed improvement (60 percent) with relatively larger accuracy degradation (22 percent) on average for Faster R-CNN. In general, static DSA introduces a large performance benefit with little accuracy degradation. However, as stated previously, it requires prior category knowledge.

For AutoFocus, there is a general trend across categories that AutoFocus brings larger speedup improvement but with relatively larger accuracy degradation compared to static DSA. It introduces 51- and 211-percent speed improvement with 22-percent (0.09) and 41-percent (0.22) accuracy (average precision) degradation on average for Faster R-CNN and R-FCN, respectively. The primary reason for the accuracy degradation is our runtime's inefficiency. It is failing to accurately model the intra-category dynamism. As mentioned previously, Faster R-CNN is more robust to both DSA techniques, which results in the accuracy degradation difference. However, if we compare speedup and accuracy degradation equally, our runtime, on average, still outperforms static DSA with prior knowledge.

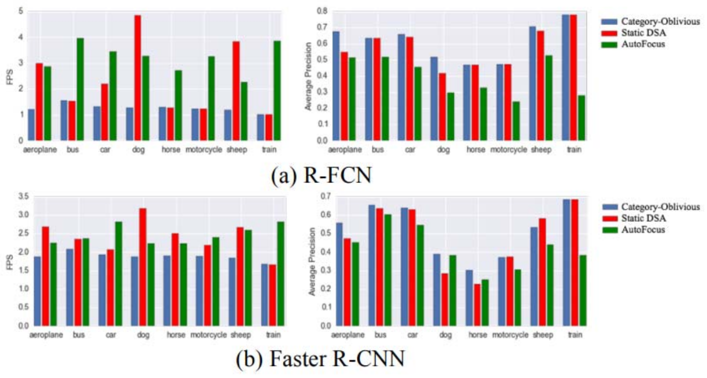


Figure 6. The performance and accuracy comparison among category-oblivious DSA, static DSA, and AutoFocus.

CONCLUSION

We present a limit study on category-aware static and dynamic DSA. In addition, we present the challenges in implementing DSA and present our insights from our proof-of-concept system. Our results show that category-aware DSA opens new opportunity for better speed or energy efficiency and takes a step toward its realization. However, to harness the benefit, we think more research effort should better addressed the challenges. In the long run, as visual systems become increasingly more intelligent, we believe DSA will offer significant improvements to system designers that are tasked with the trade-offs between performance, power, and accuracy.

REFERENCES

1. J. Huang et al., “Speed/accuracy trade-offs for modern convolutional object detectors,” *arXiv*, 2016; doi.org/1611.10012.
2. S. Ren et al., “Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, Advances in neural information processing systems, 2015, pp. 91–99.
3. J. Dai et al., “Object detection via region-based fully convolutional networks,” *In Advances in neural information processing systems*, 2016, pp. 379–387.
4. M. Everingham et al., “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, 2015, pp. 98–136.
5. M.D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *European conference on computer vision*, 2014, pp. 818–833.
6. K He et al., “Deep residual learning for image recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
7. W. Liu et al., “Ssd: Single shot multibox detector,” *European conference on computer vision*, 2016, pp. 21–37.
8. J. Redmon et al., “You only look once: Unified, real-time object detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
9. T.-Y. Lin et al., “Microsoft coco: Common objects in context,” *European conference on computer vision*, 2014, pp. 740–755.
10. Y. Jia et al., “Caffe: Convolutional architecture for fast feature embedding,” *arXive Preprint*, 2014; doi.org/1408.5093.
11. M.A. Laurenzano et al., “Input responsiveness: using canary inputs to dynamically steer approximation,” *ACM SIGPLAN Notices*, vol. 51, no. 6, 2016, pp. 161–176.
12. D. Mahajan et al., “Towards statistical guarantees in controlling quality tradeoffs for approximate acceleration,” *Proceedings of the 43rd International Symposium on Computer Architecture*, 2016, pp. 66–77.

13. S. Sidiroglou-Douskos et al., “Managing performance vs. accuracy trade-offs with oop perforation,” *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering* (ACM), 2011, pp. 124–134.
14. A. Sampson, “The case for compulsory approximation,” *Workshop on Approximate Computing Across the Stack*, 2016.
15. M. Buckler, S. Jayasuriya, and A. Sampson, “Reconfiguring the imaging pipeline for computer vision,” *arXiv*, 2017; https://arxiv.org/abs/1705.04352.
16. Z. Yan et al., “Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition,” *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2740–2748.

ABOUT THE AUTHORS

Ting-Wu (Rudy) Chin is a PhD student working with Dr. Diana Marculescu in the Department of Electrical and Computer Engineering at Carnegie Mellon University. His research interests broadly cover machine learning, computer vision, AI systems, energy-aware computing, and mobile cloud computing. He previously studied computer science at National Chiao Tung University. Contact him at tingwuc@andrew.cmu.edu.

Chai-Lin Yu is a software engineer at MediaTek Inc., Taiwan. His research interests include high-performance heterogeneous computing, automatic performance tuning, and performance modeling. He previously studied computer science at National Chiao Tung University. Contact him at tony1223yu.cs00@g2.nctu.edu.tw.

Matthew Halpern is a PhD student in the Department of Electrical and Computer Engineering at the University of Texas at Austin. His research interests include runtime systems and computer architecture for mobile computing. Contact him at matthalp@utexas.edu.

Hasan Genc is an undergraduate student in the Department of Electrical and Computer Engineering at the University of Texas at Austin under the supervision of Dr. Vijay Janapa Reddi. His research interests include computer architecture, operating systems, and software-hardware co-design. Contact him at hngenc@utexas.edu.

Shiao-Li (Charles) Tsao is a professor in the Department of Computer Science and the director of the Institute of Computer Science and Engineering at National Chiao Tung University. His research interests include energy-aware computing, embedded software and system, and mobile communication and wireless network. He has a PhD in engineering science from National Cheng Kung University. Contact him at sltsao@cs.nctu.edu.tw.

Vijay Janapa Reddi is an associate professor in the Department of Electrical and Computer Engineering at the University of Texas at Austin. His research interests include architecture and software design to enhance system performance, user experience, energy efficiency, and reliability for consumer devices and autonomous systems. Janapa Reddi has a PhD in computer science from Harvard University. He previously studied at the University of Colorado at Boulder and Santa Clara University. Contact him at vj@ece.utexas.edu.