Training Domain Specific Models for Energy-Efficient Object Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose an end-to-end framework for training domain specific models (DSMs) to obtain both high accuracy and computational efficiency for object detection tasks. DSMs are trained with distillation [1] and focus on achieving high accuracy at a limited domain (e.g. fixed view of an intersection). We argue that DSMs can capture essential features well even with a small model size, enabling higher accuracy and efficiency than traditional techniques. In addition, we improve the training efficiency by reducing the dataset size by culling easy to classify images from the training set. For the limited domain, we observed that compact DSMs significantly surpass the accuracy of COCO trained models of the same size. By training on a compact dataset, we show that with an accuracy drop of only 3.6%, the training time can be reduced by 93%.

1 Introduction

Implementing CNN based object detection on stationary surveillance cameras can lead to enhancing the safety of cities, homes, offices and factories by detecting unauthorized substances or discovering anomaly events (e.g. a collapsed person). However, the computation-efficiency is a key requirement since such devices demand battery operation to ease installation. Many successful approaches to improve the efficiency of image classification have been proposed, such as model compression[2] and model cascades with domain specific models [3]. However, object detection is more complex than image classification, and while these techniques are likely to remain effective, there is need for additional methods.

Instead of compressing large models, we target to train a computation-efficient model for *each specific* surveillance camera and a framework is proposed to train such domain specific models (DSM). The framework is based on knowledge distillation [1][4][5] but targets to reduce the accuracy gap between student and teacher models by training the student using a restricted class of domain specific images. Since such training may be conducted on edge-devices, we improve the training efficiency by culling easy-to-classify images with small accuracy penalty.

This paper's contribution is summarized below.

- We propose an end-to-end framework for training domain specific models (DSMs) to mitigate the tradeoff between object-detection accuracy and computational efficiency. To the best of our knowledge, this is the first successful demonstration of training DSMs for object detection tasks.
- By training resnet18-based Faster-RCNN DSMs, we observed a 19.7% accuracy (relative mAP) improvement compared to COCO trained model of the same size, tested on a customized YoutubeLive dataset.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

Algorithm 1 Training Domain Specific Models

Require: Domain Specific Model (DSM), Teacher Model

procedure 1. PREPARE DATASET
  Given domain images x_i for $i = \{0, \ldots, N_{\text{train}} - 1\}$
  for $i < N_{\text{train}}$ **do**
    $\text{label}(i) \leftarrow \text{Teacher.predict}(x_i)$.
    $\text{pred}(i) \leftarrow \text{DSM.predict}(x_i)$.
    compute $L_{\text{train}}(i)$ from $\text{label}(i)$ and $\text{pred}(i)$.
  Collect $[x_i, \text{label}(i)]$ pairs with n largest values of L_{train}.
Compile Difficult Dataset (DDS): $\Omega = ([x_0, \text{label}(0)]), \ldots, [(x_{n-1}, \text{label}(n-1)])$.
procedure 2. TRAIN DSM
  DSM.train (Ω)
procedure 3. INFERENCE
  Detection $\leftarrow$ DSM.predict(image)

Figure 1: Object detection results of the test image, before and after domain specific training.

- Since edge devices will have limited resources, we propose culling the training dataset to significantly reduce the computation resource required for the training. Only training data that has high utility in training is added. This filtering allows us to reduce training time by 93% with an accuracy loss of only 3.6%.

2 Training Domain Specific Models

Large scale object detection datasets such as COCO[6] contain a large and diverse set of natural images. Using a small model on such a large dataset would typically yield higher misdetections than a large model. Furthermore, [4] showed that misdetections usually occur between foreground and background (false positives + true negatives); rarely do misdetections occur as a result of inter-class errors. In video surveillance, because frames in a video stream share a stationary background, a compact model can be good enough to detect foreground and background. This motivates our DSM framework to train compact models with dataset constructed by domain-specific data.

As illustrated in Algorithm 1, our DSM framework consists of preparation of the data and training of the DSM. A large challenge when deploying models in surveillance is preparing the training data since manually labelling frames in videos is cumbersome. To overcome this, we label the dataset used to train the DSM by using the predictions of a much larger teacher model with higher accuracy and treating these predictions as ground truth labels. Furthermore, we compare the prediction on image x_i made by the teacher to that of the DSM; we determine whether to store the x_i and label Teacher.predict(x_i) in our compiled dataset Ω. After the training set is compiled, it is used to train the DSM.

Training a object detection model can take hours even with a GPU and can be challenging for applications requiring frequent retraining. We exploit the fact that when the DSM is pretrained on large-scale general dataset, it can already provide good predictions for a large chunk of the domain-specific data. This procedure develops a compact dataset Ω that is only composed of data that the DSM finds inconsistent with the prediction made by the teacher model. Keeping data x_j that both the DSM and teacher detections are consistent is computationally redundant because it does not contribute to gradient signals. We define L_{train} to quantify the consistency between teacher and DSM:

$$L_{train} = \frac{FP + TP}{TP + \epsilon} + \frac{FN + TP}{TP + \epsilon} \quad (1)$$

Table 1: Domain specific training results are summarized, where the mean accuracy result of 5 datasets are reported. Res101 results are used as ground truth, therefore accuracy is relative mAP (rmAP). Along the model name, parameters and inference time on GPU per image is reported.

	Teacher: Res101 [48M/68ms]					
	Res18 [12M/26ms}			Squeeze [6M/21ms]		
	COCO	DSM	Improvement	COCO	DSM	Improvement
mean accuracy	54.5	74.3	**+ 19.7**	41.5	63.3	**+ 21.7**

Table 2: Number of training samples versus accuracy with res18. For simple, we pick the first N training data and filter out the rest. For difficult dataset (DDS), N training data having highest L_{train} are chosen. The mean accuracy drop of the 5 datasets were computed, in respect to the model trained with all 3600 images. The training time does not include the time for teacher model labeling, which takes about 10 min. We utilize single TitanXp GPU to measure the training time.

Dataset	Classes	Strategy	Number of training samples n				
			64	128	256	512	All (3600)
coral	1	Simple	81	89.4	89.6	90	90
		DDS	89.6	89.6	89.6	89.8	
taipei	4	Simple	50.4	62	62.1	62.8	68.2
		DDS	60.7	61.7	62.2	64.2	
jackson	2	Simple	52.5	60.1	60.9	72.8	87.0
		DDS	71.6	76.7	78.3	80.6	
kentucky	2	Simple	35	38.7	44.2	54.8	67.2
		DDS	53.1	63.8	66.4	69.5	
castro	3	Simple	60.4	63	65.2	66	77.6
		DDS	67.2	68.35	75.0	77	
mean accuracy drop	-	Simple	23.6	20.8	13.8	11.3	0
		DDS	**9.5**	**5.9**	**3.6**	**1.7**	
Training Time [min]	-	-	**1.8**	**3.6**	**7.4**	**14.6**	110

where TP, FP, FN represents the number of true positive, false positive and false negative bounding-box (BB) detections of the image and $\epsilon = 0.5$. Significantly fewer training data and steps are required with only a minimal penalty in accuracy.

3 Experiments

Models. We develop Faster-RCNN object detection models on PyTorch pretrained on MS-COCO[7][8][6]. We use 3 models: resnet101(res101), resnet(res18), and squeezenet(squeeze) as the backbone region proposal network (RPN) [9][10]. Res18 and squeeze holds 10% and 19% TOP-5 Imagenet error, which is a common accuracy range for compact CNN models like MobileNet [11]. During training, res101 is used as the teacher, while res18 and squeeze are used as DSMs. While we chose Faster-RCNN for its accuracy on YoutubeLive, we can also use YOLO/SSD detectors for improved efficiency with this framework because the training requires the bounding box labels [12][13].

Dataset. We obtain 5 fixed-angle videos from `YouTubeLive`. The video is 2 hours each with 1 frames-per-second (fps), consisting of 7200 images. We split the images evenly: the first 3600 images are for training and the later 3600 images for testing.

Results. As shown in table 1, we first train our res18 DSM using the full $N_{\text{train}} = 3600$ training images for 10 epochs using stochastic-gradient descent with a learn rate of 10^{-4}. As compared to the res18 model pretrained on MS-COCO but without domain specific training, we achieved an average of 19.7% accuracy improvement.

Table 2 shows the effectiveness of DDS on res18. Using DDS, we were able to reduce the training time by 93% (256 images) with only 3.6% accuracy penalty. If we simply picked 256 training images sequentially (strategy simple on table), the accuracy worsens 10.2% compared to DDS.

References

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[2] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[3] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. Noscope: optimizing neural network queries over video at scale. *Proceedings of the VLDB Endowment*, 10(11):1586–1597, 2017.

[4] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 742–751, 2017.

[5] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4119–4128, 2018.

[6] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[7] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

Table 3: Accuracy improvement observed for multiple dataset settings. For PASCAL+YoutubeLive, we train the models with PASCAL-VOC2007 and YoutubeLive data. The accuracy improvement is rmAP improvement compared to the COCO trained model.

	PASCAL+YoutubeLive	YoutubeLive	Domain Specific
mean accuracy improvement Res18	+ 9.7	+ 12.9	+ 19.7
mean accuracy improvement Squeeze	+ 10.3	+ 13.5	+ 21.7

4 Appendix

4.1 Comparison against Data Distillation

Data Distillation [5] is fundamentally different from our application setting and methods. Models with large network capacities were shown to achieve higher accuracy by bootstrapping the dataset with [5]. On the other hand in our framework, in order to fully utilize the small network capacity, we aim to train the models with only the domain specific data.

We show on Table 3 that following a traditional method of data distillation (i.e. aggregating PASCAL with YoutubeLive data) yields lower rmAP improvement than with our approach of training with only domain specific data. In addition, training the small models with the entire YoutubeLive dataset also yields lower improvements as well. This is fundamentally because of the limited model capacity of the compact, but computationally-efficient model. We observe that for training small models, utilizing a larger dataset do not always obtain better results but restricting the data domain can do better.