Interactive Adaptation of Real-Time Object Detectors

Daniel Goehring[1], Judy Hoffman[2], Erik Rodner[3], Kate Saenko[4] and Trevor Darrell[1,2]

***Abstract*— In the following paper, we present a framework for quickly training 2D object detectors for robotic perception. Our method can be used by robotics practitioners to quickly (under 30 seconds per object) build a large-scale real-time perception system. In particular, we show how to create new detectors on the fly using large-scale internet image databases, thus allowing a user to choose among thousands of available categories to build a detection system suitable for the particular robotic application. Furthermore, we show how to adapt these models to the current environment with just a few in-situ images. Experiments on existing 2D benchmarks evaluate the speed, accuracy, and flexibility of our system.**

I. Introduction

The ability to quickly program an interactive robotic system to recognize large numbers of object categories is desirable for numerous applications including eldercare, inventory management, and assembly operations. However, robust real-time training and detection of large numbers of object models remains a key challenge problem in machine vision.

In recent years, remarkable progress has been made towards large scale object recognition, exploiting web-based annotated datasets including ImageNet [1], PASCAL [2], LabelMe [3], and SUN [4]; recognition of thousands of categories has been demonstrated in the ILSVRC challenge [1]. While bottom-up segmentation schemes are sometimes viable, operation in cluttered real world conditions calls for category-level detectors that perform multi-scale sub-window scans over the image to detect a category of interest [5], [6].

Deformable Part Models (DPM) [5] are among the best performing methods in challenges that rigorously test detection performance in difficult conditions, e.g., PASCAL VOC Challenge [2]. Implementations with efficient inference schemes exist [7] but are limited to models trained offline using a computationally expensive training process and a fixed set of categories (e.g., the 20 PASCAL objects). At the extreme, large numbers of such a priori models could be pre-computed for all of ImageNet, or for typical search phrases [8]. In this paper, we show how to train and adapt detection models quickly and on-demand, allowing the robotics user to customize the perception system to the particular needs of the application.

[1]International Computer Science Institute (ICSI), Berkeley, CA, USA goehring@icsi.berkeley.edu
[2]EECS, University of California at Berkeley, Berkeley, CA, USA judyhoffman@berkeley.edu, trevor@eecs.berkeley.edu
[3]Friedrich Schiller University of Jena, Germany erik.rodner@gmail.com
[4]EECS, University of Massachusetts at Lowell, Lowell, MA, USA saenko@eecs.uml.edu

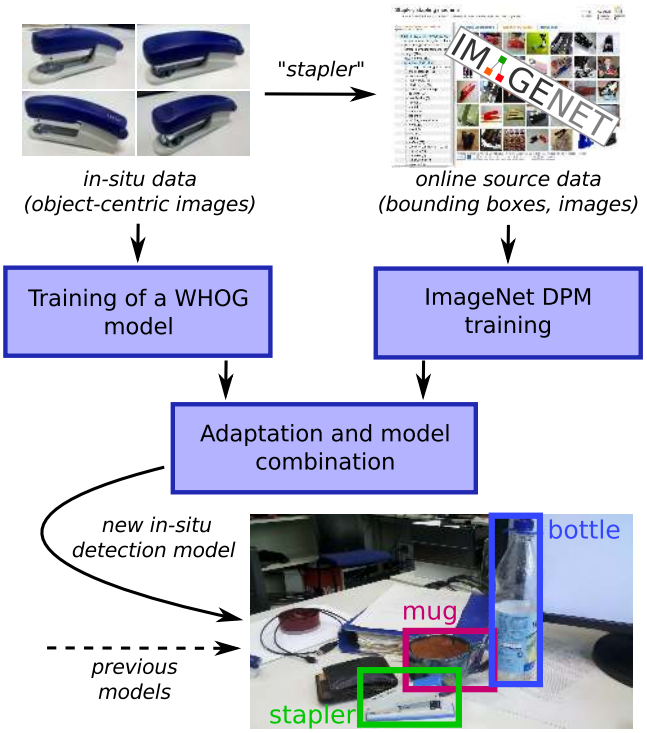


Fig. 1. Overview of our interactive object category learning and detection approach.

Unfortunately, models trained on large-scale datasets collected from the web often suffer in comparison to models trained from in-situ data in many domains [9]. The conventional alternative—requiring exhaustively labeled training instances in an environment—is overly burdensome and not necessary. Techniques for domain adaptation [10], [11] combine examples from a source domain with a small number of examples from the actual test environment, but require an expensive training step. In this paper, we develop a near real-time solution for adapting models from web sources to the test environment.

Our key innovation is the use of a fast approximate training scheme, based on the Whitened Histogram of Oriented gradients (WHOG) classifier model recently presented in [12]. This method provides orders of magnitude faster training of scanning window models than previously possible with conventional SVM-based training schemes, and facilitates training of models on-the-fly in target environments where there is insufficient labeled data to train a part-based model[1].

[1]But see [13] for a method that can train a WHOG-based model with parts, albeit more slowly than the model used in this paper.

Our method also efficiently combines elements of the DPM trained on different data sources into a single adapted model.

Figure 1 illustrates an example use of our overall approach. The user provides the name of a category of interest (in this case, "stapler") and, optionally, a few image examples (in-situ data). The system then downloads online source data from ImageNet (other repositories could be used) and trains a WHOG model on the fly, or retrieves a pre-trained DPM model if one is available. This model is then adapted to the provided in-situ data. The entire online training and adaptation process is very efficient, and takes less than 30 seconds on an average processor. The final model is then added to an existing set of object models and real-time inference proceeds at approximately 5 Hz with 10-20 categories on a conventional machine, using available real-time inference methods [14], [7], [15].

We briefly summarize the key contributions of our paper:

- a fast method for training detectors on demand from large-scale repositories such as ImageNet,
- a mechanism to interactively adapt such models with in-situ examples to lessen the effects of domain shift,
- ability to add in-situ categories, subcategories, or instances missing from the repository using the fast training scheme.

Below we report the performance of our method both in laboratory experiments and using a robotics-geared 2D object recognition benchmark. We will release an open source end-to-end CPU/ROS implementation of our toolbox for research use under http://raptor.berkeleyvision.org . We also provide a video demonstration of the full system in supplementary materials.

II. Related work

a) Object detection in Robotics: Several approaches enable a robot to acquire object models autonomously from automatically segmented 3D point clouds [16], [17], or RGB-D descriptors [18], [19], [20]. In this work, we focus on supervised training with 2D images, which utilizes existing large-scale online repositories as well as (minimal) labels interactively provided by the user. Lai *et al.* [19] present a semantic segmentation approach that tries to label each voxel in an RGB-D image with a semantic label. They use sliding window detection together with an additional MRF inference step to enforce consistency among neighboring voxels and also learn detection models using ImageNet. In contrast to our approach, they rely on expensive hard negative mining for learning and do not allow models to be refined or adapted. Spinello and Arras [21] present an adaptation technique for fusing RGB and depth data using a Gaussian process model. The adaptation technique presented in this paper is complementary to that approach, as it focuses on combining the strength of two datasets of the same modality (color RGB images). Saenko *et al.* [22] combine category-level DPM with an instance-level local feature method in a single detector, however all images of categories and instances come from the target test environment. Our method frees the user from having to collect all of the training data by utilizing the ImageNet repository, yet improves accuracy by allowing the user to augment the model with in-situ data.

b) Interactive training: Interactive visual learning of object categories has been studied by Fritz *et al.* [23] with a scheme based on scale-invariant patterns, a variant of a spatial bag-of-words model. Although our focus is also on interactive learning, our approach is based on state-of-the-art category detection models [5], [12] and the incorporation of additional training data from large-scale internet sources.

c) Adaptation: Domain adaptation and knowledge transfer have been mostly studied in the area of image categorization [24], [25], where the goal is to label an entire image instead of performing object localization. An exception is the Projective Model-Transfer SVM algorithm (PMT-SVM) proposed by [10], which adds another regularization term to the SVM optimization problem. Whereas PMT-SVM tries to find a model close to the source domain model, our mixture adaptation technique combines the strength of a target and source domain model directly without tuning regularization terms, and avoids the computational expense of re-training. The work of [26] shows adaptation techniques applicable to Exemplar SVM detectors [27], where a separate SVM classifier needs to be trained for each training example, and also evaluated during testing. Our method is based on the more efficient whitened HOG method [12], which allows for learning detection models with several thousands examples in a few seconds.

III. Real-time deformable part detectors

In the following, we review the detection framework used in our approach.

A. Deformable part models

One of the most common approaches to object detection in cluttered 2D images is a linear sliding-window detector, which filters a d-dimensional feature representation $\phi(I) \in \mathbb{R}^{w \times h \times d}$ of an image $I \in \mathbb{R}^{w \times h}$ with a filter vector $\mathbf{w} \in \mathbb{R}^{w' \times h' \times d}$ learned with a linear classifier such as an SVM, and considers the locations $\mathbf{x}$ with the highest output as detection candidates:

$$\text{argmax}_{\mathbf{x}}\, f_{\mathbf{w}}(\mathbf{x}) = \text{argmax}_{\mathbf{x}}\, [\mathbf{w} * \phi(I)]\,(\mathbf{x}) \qquad (1)$$

where $*$ denotes convolution in all d feature channels of $\phi(I)$ and we skipped the maximization with respect to different scales to simplify notation. In practice, all locations with values of $f_{\mathbf{w}}$ above a predetermined threshold are considered positive detections. It has been shown in several papers that histograms of oriented gradients (HOG) provide a suitable feature representation invariant to illumination changes and small shifts [5].

A deformable part model M is meant to be invariant to larger object deformations by allowing parts of the objects to move. It consists of a set of filters $\mathcal{F} = \{\mathbf{w}, \mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and a model for their spatial layout expressed as a deformation cost model $\mathbf{d}$. The root filter $\mathbf{w}$ is intended to cover the whole

Fig. 2. Examples of models learned from a single in-situ image and from ImageNet

object and the remaining filters cover parts of the object. The combined detection score is calculated by:

$$f_M(\mathbf{x}) = \max_{\mathbf{z}} \left(f_{\mathbf{w}}(\mathbf{x}) - \mathcal{D}(\mathbf{z};\mathbf{d}) + \sum_{j=1}^{k} f_{\mathbf{w}_j}(\mathbf{x}+\mathbf{z}_j) \right)$$

where $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_k)$ are the latent part locations, $\mathcal{D}$ is the cost of $\mathbf{z}$ with respect to a learned deformation model, and b is a bias term. The entire model is usually learned with a latent SVM scheme and is described in detail in [5].

Different views/aspect ratios can be handled by training a mixture of components, $\mathcal{M} = \{M_0, M_1, ...\}$, in which case z is augmented to add the latent component label that the example belongs to. The score of a mixture of components at a particular position in the image is defined as the maximum score of any component. In section IV, we take advantage of the component mixture formulation for fast online adaptation.

B. Fast inference with Fourier transformation

The main bottleneck of the detection is the convolution of the learned filters with the HOG feature map of the image and many possible directions to speed up this part of the algorithm have been proposed and studied [7], [15], [8], [14]. In our approach, we make use of the method presented in [14], which exploits the convolution theorem and computes the pointwise product of the respective Fourier transforms of the filter and the feature representation. This technique is orthogonal to [7], [15], [8] and provides speed-ups independent of the number of models used during detection. The public implementation of [14] is based on C and the `fftw` library and leads to a detection speed of around 2 Hz with 20 models on a 2.5 GHz machine using 320x240 pixel images.

C. Fast learning of detection models

Learning a complete DPM with latent SVM as proposed in [5] takes several hours on a standard machine, which is impractical for our purposes. As shown by [12], training a single sliding-window root filter, like that in eq. (1), can be done efficiently using simple Gaussian assumptions for the HOG features. Let us consider a single root filter $\mathbf{w}$ that must learn to differentiate between positive (sub-images showing an instance of the object category) and negative (sub-images of other categories or background) examples. Following [12], we now assume that both positive and negative examples are Gaussian distributed with the same covariance matrix $\mathbf{S}_0$ and mean vector μ_1 and μ_0, respectively. It can be shown that in this case, the optimal hyperplane separating both sets can be calculated as [12]:

$$\mathbf{w} = \mathbf{S}_0^{-1} (\mu_0 - \mu_1) \quad . \tag{2}$$

Although the underlying assumptions leading to this equation might be unrealistic in practice, the resulting simple learning step is at a closer look, a common feature whitening step. It implicitly decorrelates all the HOG features using statistics of a large set of (negative) examples. An important step necessary to deal with the high correlations naturally arising between neighboring HOG cells.

The covariance matrix as well as the mean μ_0 of negative examples can be estimated from an arbitrary set of negative images and HOG features calculated therein. Therefore, we can easily pre-compute it and re-use it for every new category model. Thus learning a new category only involves averaging the HOG features of the positive images and finding a suitable detection threshold.

In our case, we optimize the detection thresholds on the set of in-situ images with object bounding boxes interactively provided by the user. Specifically, we look at the scores of each user provided bounding box and choose a threshold value that would maximize a function of precision and recall (for example f1 measure), where a detection is considered correct if it overlaps the true bounding box by at least 50%.

IV. Interactive Learning and Adaptation

In the following, we show how to make use of large-scale image datasets within an interactive learning system for in-situ object detection.

A. ImageNet

The computer vision field has been studying recognition of common categories and has accumulated plenty of labeled image data from various internet sources, of which the ImageNet project [1] is just one example. ImageNet consists of over 14 million images and over 21k semantic concepts (synsets), many with annotated bounding boxes; e.g., the category *stapler* has over 1400 images with available bounding boxes. An important part of our approach is to make use of this data.

When the user teaches the system a new category, we search for a synset in ImageNet that matches the category name given by the user. The matching scheme searches the descriptions of the synsets for the terms given by the user. Finally, we select the most general synset (lowest level node in the ImageNet hierarchy) from the list of matching synsets and download image data and bounding box annotations

Fig. 3. Top: Demo setup with PR2 and objects to detect in front of it; Bottom Left: Training user interface, object to learn in region of interest (red square); Right: Detected objects in bounding boxes.

from a local copy of the repository, which only takes a few seconds for most of the categories used in our experiments.

We can then directly learn an object detector from the given training data using the scheme above, and additionally adapt it to the current environment, as shown in the next section. The whole process takes under 30 seconds on a standard computer and the main computational burden is the data extraction and not the learning part itself. In contrast to previous learning schemes for object detection used in robotics, this is a speed-up of orders of magnitude; boosting based training [6] or standard DPM training [5] usually takes several hours to complete.

B. Adaptation of detection model mixtures

When learning from two different data sources, problems often arise due to a possible domain shift [24], [25], *i.e.* a difference in the underlying data distributions. Several papers already showed that there is indeed a domain shift between ImageNet and object images taken in a robotics environment [28]. This is due to the fact that ImageNet is collected using internet search engines, which creates a bias to images found on `flickr` or shopping websites such as `amazon`, where objects appear in canonical poses and specific lighting conditions.

Our adaptation strategy is to incorporate the models trained on in-situ objects with max fusion, *i.e.* we add them as additional components in the ImageNet mixture model. Let $\mathcal{M}_I$ be the mixture model obtained from the

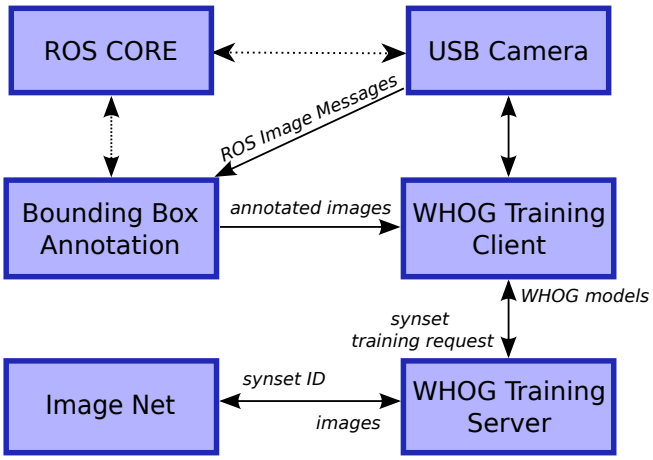


Fig. 4. Modules and data flow chart.

ImageNet dataset and let $\mathcal{M}_O$ be the mixture model learned from the examples supplied by the user. The combined model $\mathcal{M}$ performs detection by maximizing over all mixture components:

$$f_{\mathcal{M}}(\mathbf{x}) = \max_{M \in \mathcal{M}_I \cup \mathcal{M}_O} f_M(\mathbf{x}) \tag{3}$$

One issue with combining models trained separately in this way is that a single detection threshold may not be optimal for both models. We therefore optimize two separate thresholds: one for $\mathcal{M}_O$ and one for $\mathcal{M}_I$. These thresholds should ideally be optimized on the in-situ data. However, if there is very little training data in-situ (as in our experiments) you can limit your parameter space by using only one weighting parameter α which is multiplied by the in-situ scores and $1-\alpha$ is multiplied by the ImageNet scores. Then the top detection is chosen by maximizing over all weighted scores.

This simple adaptation strategy has the following advantages compared to classifier adaptation [5]: (1) it is much faster, as it only requires estimating the optimal weighting of the components; (2) it allows for adding different views/subtypes of the object category learned by ImageNet. Fig. 2 illustrates this fact by showing a model learned from a single in-situ image and a single mixture component learned from ImageNet. When acquiring training examples, users often choose only a specific view of the object, which would limit the detector to nearly the same views during detection. Incorporating the model learned from ImageNet adds several new views of the object to the model (like the frontal view depicted in the right image in Fig. 2), increasing the robustness of the detector with respect to view changes.

C. Interactive learning interface

In case additional in-situ images are required, we try to simplify the annotation step for the user as much as possible. Therefore, the user only needs to point the camera to the new object and needs to ensure that it is displayed in a predefined area projected into the camera image. After starting the image recording, an image is taken every five seconds. The user can see how much time is left until the next image

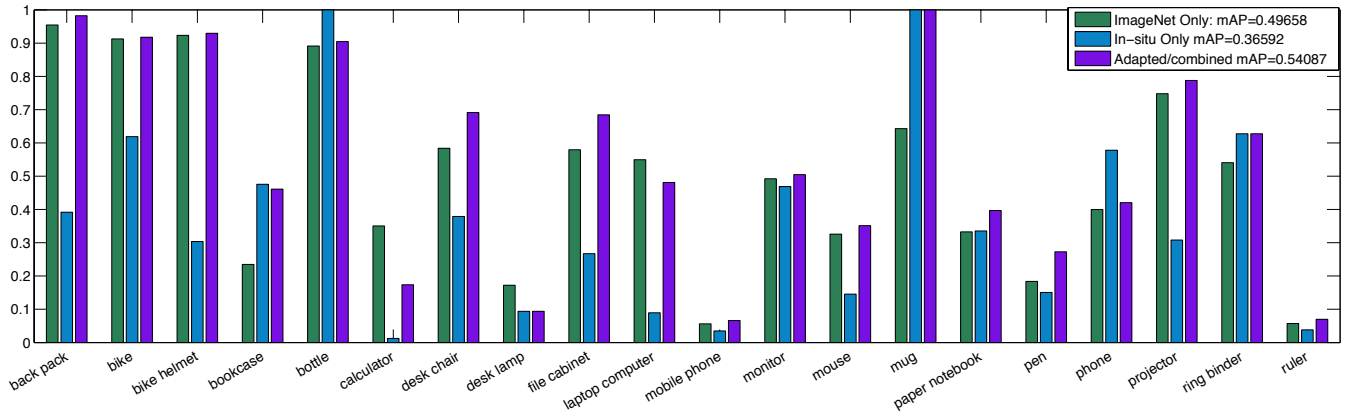


Fig. 5. The combined model has best quantitative performance for 12/20 categories. Here we plot average precision for each category for the ImageNet only model, ImageNet + in-situ tuned model, the in-situ only model, and our final adapted/combined model. Experiments were performed using the Office dataset.

is taken. This is important to let the user hold the camera still when an image is taken and reduce motion artifacts in the training images (see screenshot on the bottom left of Fig. 3). The current setup records up to five images in this manner, which is sufficient for robust detection models as shown in our quantitative experiments in the next section.

D. Overview of the system

Figure 4 depicts the software modules and data flow of our overall system. Our interactive learning and detection framework is divided into two parts: the first part takes care of training and adaptation and the second part is the real-time detection code. During training, we collect images via a camera ROS-node and annotate a bounding box in the image via a second ROS-node. The WHOG training client sends a training request to the WHOG training server (usually a different machine), which has access to ImageNet data. After training is finished, the training client receives a WHOG model from the training server, and provides it for use by the real time detection code in the core module.

The source code of our approach is implemented as a module for the ROS library and will be made publicly available under http://raptor.berkeleyvision.org .

V. Experiments

A. Datasets and experimental setup

Our goal is to learn object detectors suitable in new environments. For a quantitative analysis, we selected an office environment and annotated the publicly available Office dataset of [9] (webcam domain) to obtain labeled in-situ images suitable for testing our detection models. We also downloaded ImageNet synsets for those categories for which they were available with bounding boxes. This resulted in 20 categories, which we use for our quantitative experiments. The Office images were annotated with a single bounding box around the object of interest using Amazon Mechanical Turk.

For each category 3 positive and 9 negative detection examples are chosen from Office for training. However, please note that our adapted algorithms also have additional access to a large number of positive training examples obtained from ImageNet (see Table I for details). For testing, a set of roughly 25 images per category[2] is chosen randomly from the Office dataset for testing.

Performance is measured using average precision (AP) and we compare the following methods:

1) **ImageNet model only:** a WHOG model learned from all category images in ImageNet
2) **In-situ model only:** a WHOG model only learned for each category using 3 positive detection example images and 9 negative detection images from the Office dataset
3) **Adapted/combined model:** model combination with ImageNet and the in-situ data as in Section IV-B

B. Evaluation

Table II shows the mean average precision across all categories for the different methods. The average precision for a category is the integral of the precision-recall curve. We see that the adapted model, which combines training data from ImageNet and the in-situ images, achieves the highest performance. The large gap in performance between the baseline models (ImageNet only and in-situ only) and the final adapted model highlights the advantage of model adaptation to overcome the issue of dataset bias. Note that while the mean AP may seem low, this reflects the challenging nature of the task: accurately (to within 50% overlap) localizing multiple categories of objects in cluttered images.

To further analyze the results, we consider per category performance in Fig. 5. The adapted/combined model has highest performance for 12/20 of the categories and is tied with ImageNet Only for 2/20 categories, which demonstrates a consistent improvement. For example, the combined/adapted model is more than 0.1 AP higher than the ImageNet only model and is more than 0.3 AP higher than

[2]We evaluate on all 31 categories, not just the 20 with matching positive synsets in ImageNet.

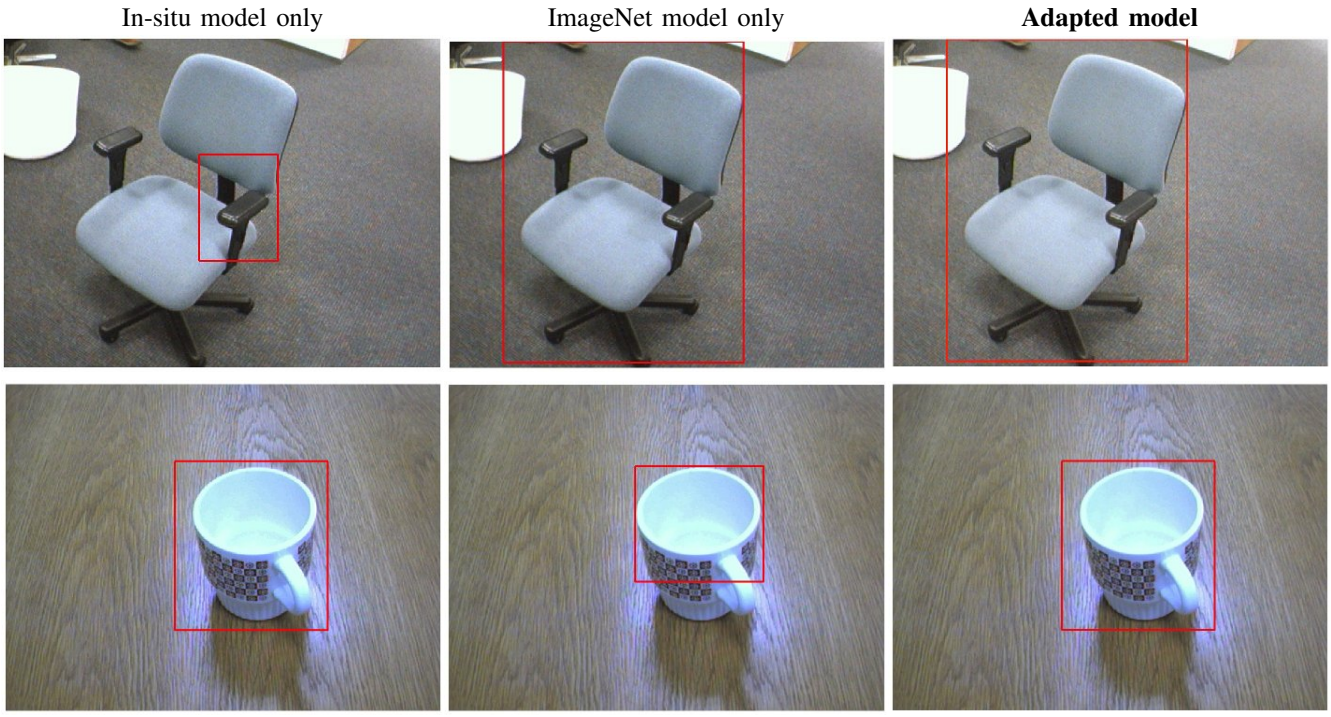


Fig. 6. The combined adapted model successfully learns from both the ImageNet and in-situ models. Here we show some example detections obtained with models learned from (Left) in-situ images, (Center) ImageNet images, and (Right) a combined adapted model. While the in-situ model fails on the chair example and the ImageNet model fails on the mug example, the combined model correctly detects both chair and mug.

category name	synset	number of ImageNet training examples
back pack	n02769748	639
bike	n02834778	275
bike helmet	n03127747	625
bookcase	n02871439	315
bottle	n04557648	478
calculator	n03483823	510
desk chair	n03001627	2248
desk lamp	n04380533	509
file cabinet	n03337140	490
laptop computer	n03642806	820
mobile phone	n02992529	532
monitor	n03085219	637
mouse	n03793489	830
mug	n03797390	652
paper notebook	n02840619	157
pen	n03906997	641
phone	n03179910	171
projector	n04009552	578
ring binder	n02840245	499
ruler	n04118776	578

TABLE I

NUMBER OF LABELED BOUNDING BOXES WE USED AS TRAINING EXAMPLES FROM IMAGENET FOR EACH CATEGORY.

the in-situ only model. This is likely due to some types/views of chairs not being available in ImageNet, despite its large size. We also show some detection examples on the Office test set in Fig. 6 to showcase the ability of the adapted model to effectively combine the strengths of both the in-situ and the ImageNet data sources. Additionally, screenshots of the system running live is shown on the bottom right of Fig. 3 and Fig. 7 (refer to the supplementary material for a video demonstrating real time detection in a lab setting).

Our results also reveal large differences between categories: while overall AP is high for certain objects, like bike, bottle and mug, it is very low for a few objects, such as pen and ruler. The smaller objects are hard to detect in our environment primarily because the training and test in-situ data is not pose-normalized. In other words, while a bike will almost always captured in one pose (upright, wheels at the bottom etc), a pen, calculator, ruler, or phone, can be captured in any orientation or pose. This creates a problem for our basic models which attempt to learn a single representation per category choice. Fig. 8 shows example in-situ calculator images as well as the calculator models learned both from ImageNet and in-situ data. The ImageNet model learns a clean box-like structure, but is limited in that it has to be in a particular orientation. The in-situ model learns essentially nothing since the training data is not pose-

Method	Mean average precision
ImageNet model only	0.4966
In-situ model only	0.3659
Adapted/combined model	**0.5409**

TABLE II

DETECTION PERFORMANCE OF OUR METHOD TESTED ON IN-SITU IMAGES FOR DIFFERENT LEARNING SCENARIOS.

Fig. 7. Example images during detection experiments.

normalized. In the future, we would like to improve our detector by adding some rotation invariance to our models.

VI. Conclusions

We presented an approach to learning detection models on-the-fly while combining training data from internet sources with a few images from the enviroment. Our results demonstrate that the time consuming and tedious step of collecting hundreds of images of each object category of interest for robotic perception can be in many cases successfully avoided. However, we also show that it is important to adapt models learned from internet sources to the target enviroment. We proposed a simple adaptation scheme whereby the internet model is quickly combined with an in-situ model with appropriate threshold tuning. The advantages of this scheme were demonstrated by showing the significant performance gains in our experiments when using the adaptation technique.

In future work, we would like to extend the current system towards object discovery, such as proposing object hypotheses to the user to further reduce the amount of supervision necessary [29]. Furthermore, active learning techniques [30] could be used to guide the acquisition step during learning to examples with a significant impact on the classification model.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[3] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

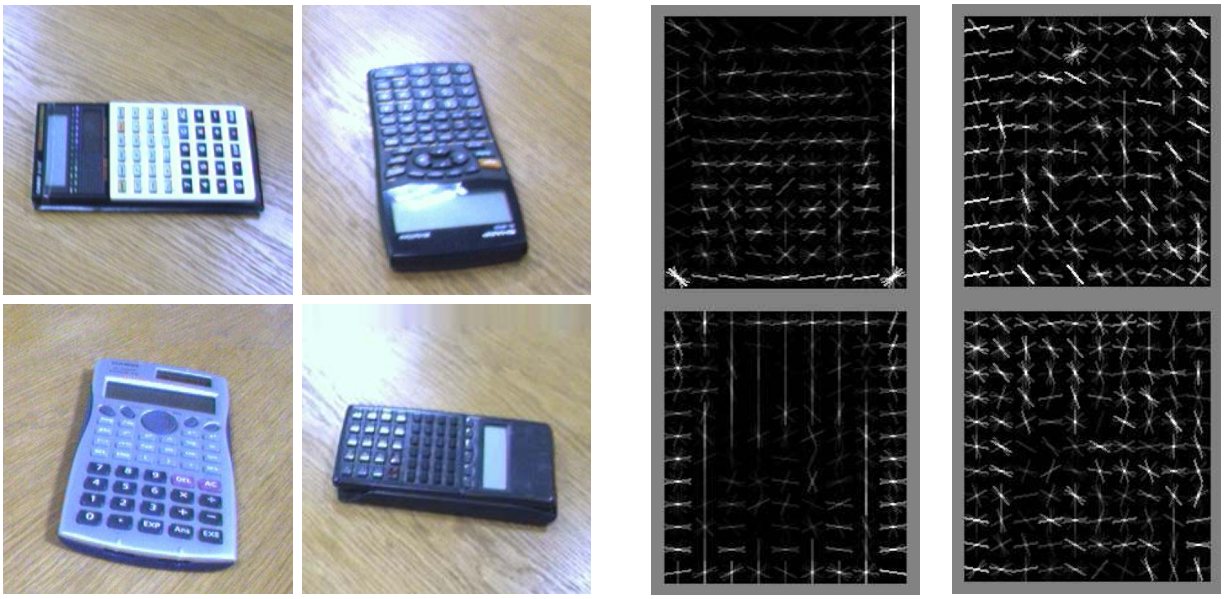
[4] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*. IEEE, 2010, pp. 3485–3492.

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–511.

[7] H. O. Song, S. Zickler, T. Althoff, R. Girshick, M. Fritz, C. Geyer, P. Felzenszwalb, and T. Darrell, "Sparselet models for efficient multiclass object detection," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 802–815.

[8] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.



(a) (b) (c)

Fig. 8. Calculator class: (a) Example in-situ calculator images, (b) Imagenet Calculator Model, (c) In-situ Calculator Model. The models are depicted with both a likely positive structure (top) and a likely negative structure (bottom). A discriminative model would look very have distinct positive and negative structures. The ImageNet model learns that a calculator is roughly represented by a box with horizontal lines near the top (screen viewing area) and a mix of horizontal and vertical lines on the bottom (button area). This model misclassifies any in-situ calculator image that in not in this canonical pose. Similarly, the model learned from in-situ data was unable to generalize from the calculator data seen since it was not pose normalized. Hence, neither the positive nor negative model depictions have any resemblance to a calculator and final detection performance is low.

[9] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 213–226.

[10] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *IEEE International Conference on Computer Vision*, 2011.

[11] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[12] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 459–472.

[13] R. Girshick and J. Malik, "Training deformable part models with decorrelated features," in *ICCV 2013, to appear*, 2013.

[14] C. Dubout and F. Fleuret, "Exact acceleration of linear object detectors," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 301–311.

[15] H. O. Song, T. Darrell, and R. B. Girshick, "Discriminatively activated sparselets," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 196–204.

[16] J. Behley, V. Steinhage, and A. B. Cremers, "Performance of histogram descriptors for the classification of 3d laser range data in urban environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012.

[17] M. Liu and R. Siegwart, "Information theory based validation for point-cloud segmentation aided by tensor voting," in *IEEE International Conference on Information and Automation (ICIA)*, 2013.

[18] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse distance learning for object recognition combining rgb and depth information," in *in IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[19] ——, "Detection-based object labeling in 3d scenes," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1330–1337.

[20] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann, "Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot," in *in IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[21] L. Spinello and K. O. Arras, "Leveraging rgb-d data: Adaptive fusion and domain adaptation for object detection," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4469–4474.

[22] K. Saenko, S. Karayev, Y. Jia, A. Shyr, A. Janoch, J. Long, M. Fritz, and T. Darrell, "Practical 3-d object detection using category and instance-level appearance models," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 793–800.

[23] M. Fritz, G.-J. M. Kruijff, and B. Schiele, "Tutor-based learning of visual categories using different levels of supervision," *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 564 – 573, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314210000238

[24] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1785–1792.

[25] J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell, "Efficient learning of domain-invariant image representations," in *International Conference on Representation Learning, arXiv:1301.3224*, 2013.

[26] Y. Aytar and A. Zisserman, "Enhancing exemplar svms using part level transfer regularization," in *British Machine Vision Conference*, 2012.

[27] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 89–96.

[28] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1521–1528.

[29] M. Bjorkman and D. Kragic, "Active 3d scene segmentation and detection of unknown objects," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3114–3120.

[30] A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler, "Labeling examples that matter: Relevance-based active learning with gaussian processes," in *German Conference on Pattern Recognition (GCPR)*, 2013, pp. 282–291.