Runtime Model Recommendation for Exemplar-based Object Detection

Fanyi Xiao[1], Martial Hebert[1], Yaser Sheikh[1], Yair Movshovitz-Attias[1], Mei Chen[2] and Denver Dash[2]

[1]School of Computer Science, Carnegie Mellon University
[2]Intel Science and Technology Center, Pittsburgh

Abstract

We present an approach for object instance detection that uses model recommendation to predict a subset of relevant exemplar models for object detection based on an testing image at runtime. An initial subset of randomly selected exemplar models, the probe set*, is first applied to the testing image, and its responses are used, in conjunction with a* rating matrix*, to predict the responses of all the exemplar models. The subset of exemplar models predicted to score the highest is then applied to the testing image to generate the final detections. This method enables* scaling up *the number of exemplar models to capture large object appearance variability, while maintaining computational efficiency. We present a novel max-selection scheme that allows us to build the rating matrix in a weakly-supervised fashion, allowing us to leverage large amounts of data easily. In addition to computational efficiency, we present experimental results which demonstrate that this model recommendation approach can outperform a baseline in which all the exemplar models are evaluated on the testing image.*

1. Introduction

Detecting an object instance is challenging because of the large variations in the visual world. If we think of each image as a data point, the variations among images due to viewpoint, deformation, illumination, scale, as well as intra-class variation produce a large *image space*. Conceptually, recognizing an object in an image is to partition the image space into different categories or instances. Given a limited number of training images, a common approach for object detection is to train a single *global* model that captures the entire variance within an object category [3]. However, training an effective global model is hard both in theory and practice due to the aforementioned variations. Subsequent research has presented *piecewise* approaches that break down the task of capturing the variation by using a collection of models [7, 2, 11]; these approaches have been demonstrated to capture a greater degree of variability but require larger training datasets. With the increasing access to larger datasets, methods that take this divide-and-conquer approach to the limit by training one detector per training sample have also been proposed. This *exemplar-based* approach lets the data speak for itself [18, 21, 19, 13], with each exemplar model representing a local region in the image space. This approach is particularly well suited for object *instance* detection rather than broader *category* detection problem. We present object instance detection as our application in this paper.

A critical issue of exemplar-based approaches is the ability to scale up the number of models (and therefore the degree of variability that can be captured), while keeping the computation tractable. Achieving this goal is not trivial as increasing the number of exemplar models directly leads to the increase in computation; clustering or averaging the exemplar-models negates the data-driven appeal of the approach. A common class of approaches is to share model parameters by clustering visually similar components or discovering common dictionaries offline in the training stage [5, 11, 23, 22]. There are some other works addressing this scalability issue at runtime [24, 8, 9]. Gao and Koller [9] suggest using a value-theoretic approach for selecting classifiers at run-time. Each classifier response is considered an observation that potentially holds information about the classification task, and has a computational cost associated with it. They balance classification gain and computational cost to dynamically select classifiers. Cascades of classifiers [24, 8] are another example of runtime selection of classifiers. An instance is passed through a series of classifiers, and at each point in the process a decision is made based on the previous responses whether to apply another classifier or make a prediction.

The runtime model selection is appealing since the models selected for each particular testing image could be different according to partial information we obtain at runtime. This input-sensitive approach helps us to make better decision on what models would be appropriate for each testing image, in contrast to the offline approaches which do not distinguish between two different testing images. The cascaded method in [24] builds the cascade in a input-insensitive manner whereas the approaches described in [9] select models in a input-sensitive way. However, their application scenario and approach is very different from ours. Note that they do inference at every step with the computation scales cubically with the number of evaluated models.

Our idea in this paper is to exploit the correlation among models to do runtime model recommendation in a input-sensitive manner. The key insight about exemplar models is that the models of an object category/instance are not unorganized or uncorrelated. Instead, there are strong correlation patterns exist among them. In this paper, rather than looking for correlations among models directly in image space, we look for correlations among models by checking their *responses* on images (see Figure 2a). Knowing the structure among models enables us to make predictions based on the responses of any exemplar model given the responses of other exemplar models. Therefore, it is possible to apply a small set of exemplar models (which we refer to as *probes*) on a testing image and to predict the responses of all the models. The exemplar models that are predicted to score highly on the testing image are then recommended for detection on the image. If the prediction can be computed efficiently, this saves significant computation as it can drastically reduce the number of models that we need to apply at run-time. Even when scaled up to a large number of models, it is still possible to detect objects in a tractable manner. Specifically, we use *model recommendation* [15, 20] to exploit the structure in the responses of different models on different images. Empirically we observe that model recommendation often outperforms the direct approach of applying all the exemplar models, which is surprising at first glance. This seemingly counter-intuitive result comes from the fact that model recommendation leverages information from different model responses and makes its prediction on which model responses would be likely to *fire*, i.e., exceed a fixed threshold. This helps suppress false positive detection, as can be seen in Figure 1.

To the best of our knowledge, this paper is the first to do runtime model recommendation for object instance detection task.

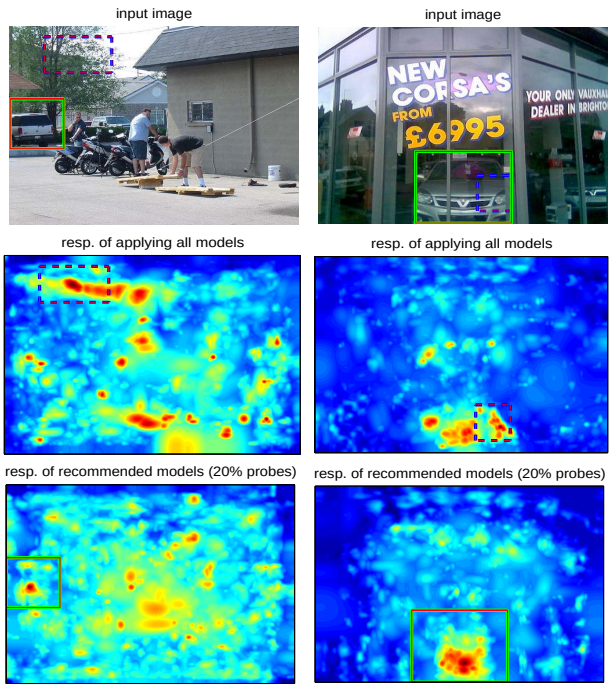


Figure 1: The first row shows the testing images, the second and third rows are responses generated by applying all the models and just models from model recommendation, respectively. In both cases, the color of each pixel encodes the maximum response of detectors at those pixels. The dotted boxes show the detection with the maximum score obtained by applying all the models; the solid boxes are the detections from the recommended models using 20% probes. The left example shows that the car is detected at wrong location by taking the maximum over all the detectors, whereas the correct location is recovered by using only 20% of the models as probes in our recommendation setting. Similarly, The right example shows that using all detectors, the object is detected with the wrong scale, and the recommended models give us correct scale of the detection.

2. Approach

2.1. Exemplar Models for Detection

Exemplar methods for object detection use an ensemble of models, each member of which is applied to an testing image using a *sliding-window*. Each exemplar model $\mathbf{w}_i, i = 1, 2, 3, ...M$ is thus used to capture one data point in the image space belonging to this object category/instance. The response $f_i(\mathbf{x})$ of an exemplar model $\mathbf{w}_i$ on image patch $\mathbf{x}$ is given by:

$$f_i(\mathbf{x}) = \mathbf{w}_i^T \phi(\mathbf{x}). \quad (1)$$

We use $\mathcal{S}_M$ to denote the entire set of exemplar models and $\phi(\mathbf{x})$ to denote the feature representation for image patch $\mathbf{x}$. To use all exemplar models in an ensemble, we typically run all the models $\{\mathbf{w}_i | \forall i \in \mathcal{S}_M\}$ on the testing image in a sliding-window manner by evaluating Eq. 1 at every possible location and scale in the image feature pyramid. A threshold is picked to identify whether or not a response

has *fired* and is thus to be considered a detection. Given a collection of detection results from different local models, *Non-Maximum Suppression* is then applied to remove the redundancy and yield the final detection results. Different kinds of exemplar models could be used in this framework, we choose *Exemplar SVM* (ESVM) because of its discriminative power [19].

Specifically, in *Exemplar SVM* (ESVM), the model parameter $\mathbf{w}_i$ is learned using large-margin learning and hard-negative mining technique [7], which gives the model more discriminative power than many other template-based methods [12, 10]. The feature representation used in ESVM is a HOG template which captures the gradient information while providing invariance to small changes in raw pixels. In the large-margin formulation, the positive sample is the HOG vector of a single image patch, the *exemplar*. The negative samples are image patches mined from a large collection of images which do not contain this object. Thus our SVM objective function is:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + C_1 l(\mathbf{w}, b, \phi(\mathbf{x}_E)) + C_2 \sum_{\mathbf{x}\in\mathcal{N}_E} l(\mathbf{w}, b, -\phi(\mathbf{x})), \tag{2}$$

where $\mathbf{x}_E$ is the exemplar vector, $\mathcal{N}_E$ is the set of negative samples while C_1 and C_2 are regularization parameters.

2.2. Model Recommendation

Our approach is to use *Collaborative filtering* (CF), a set of techniques for filtering information from various data sources, originally designed for predicting how a new customer would rate products based on a large collection of ratings from prior customers. This approach has been demonstrated to work well in recommending models for action recognition [20]. In this approach, we treat the response of model *i* on image *j* as a rating R_{ij} and we collect all the ratings in a ratings matrix $\mathbf{R}$. Using matrix factorization such as *Singular Value Decomposition* (SVD), we can discover the latent factors which characterize both the models and the images. The approach based on SVD factorization is termed factor-based *collaborative filtering* in the context of recommendation systems [14]. However, many different approaches can be used to decompose $\mathbf{R}$ in order to predict the rating vectors from a small set of probes (see [15] for a survey). Then given a new testing image, we evaluate the response of a small *probe set* of *P* models on the testing image, yielding a *P*-dimensional vector of probe ratings $\mathbf{R}_p$. By using the matrix decomposition, the ratings on *all* the other models is then estimated. The key feature of this class of approach is that, by exploiting the correlations across the models in the ratings matrix, it is possible to evaluate only a small subset of probe models. In fact, it is possible to get better performance than by evaluating all of the models, again because a large body of prior experience in applying models to images is used.

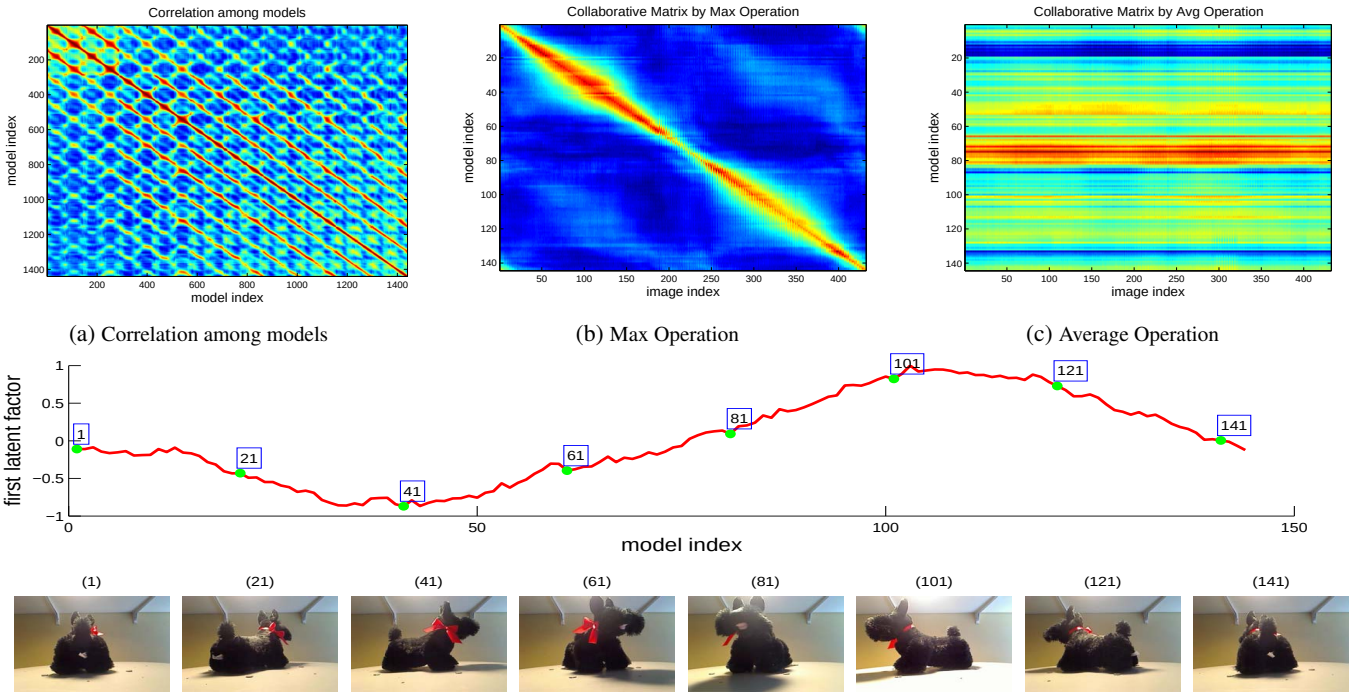
In the original work of [20] the idea was to replace the standard approach of *training* on a new task (the new "customer") by using a large dataset, with the new approach of "guessing" which models would be appropriate for the new task based on prior experience rating many different models on many different tasks (the product ratings of prior customers). Our goal is different from [20] in that we investigate the problem of recommending models to detect objects in a single image at *test time*, whereas they aim at improving the training performance. In order to illustrate it on a simple example, we refer to the same task thoughout this section (Figure 2): The models are ESVMs generated from views of an object sampled in a circle. Similarly, testing images different from the training images are also sampled from the same circle. This setup allows us to produce meaningful displays of the ratings matrix and the latent factors.

Ratings Matrix: In order to use collaborative filtering, we need to pre-compute and store a ratings matrix $\mathbf{R}$ (also called rating store in [20]) which contains responses of the different models evaluated on different images. Let M be the total number of models and N be the total number of image samples, we evaluate all M models on N samples and get a rating matrix $\mathbf{R}$ using a max operation:

$$R_{ij} = \max_{\mathbf{x}\in I_j} f_i(\mathbf{x}) = \max_{\mathbf{x}\in I_j} \mathbf{w}_i^T \boldsymbol{\phi}(\boldsymbol{x}), \tag{3}$$

where $\mathbf{x}$ is an image patch in image I_j. R_{ij} is set to be the maximum response of all patches of I_j evaluated by model $\mathbf{w}_i$. The max operation provides information about the most confident response and thus is robust to noise in model responses. For illustration, the collaborative matrices constructed using max operation and average operation (which is to replace the *max* in equation (3) with *mean*) are shown in Figure 2b and 2c, respectively. Both the models and the images are ordered according to view angles ranging from 0 to 2π. It can be seen that max operation gives us a clear pattern where models have higher responses on images with closer view angles. By contrast, the average seems to be dominated by noise and cannot produce meaningful patterns. This method of using the max does not require labeled bounding boxes in our training data. We merely need to ensure that images used for constructing the rating matrix contain the target object instance. Thus we can exploit a large database of weakly-labeled images (frame level supervision) for this task.

Baseline Estimates: As an analogy to the imbalance in *popularities* of different products, different models have different intrinsic popularities which means that some models tend to have higher responses than others consistently. This imbalance also applies to different images. To remove the effects of such imbalances, we need to estimate a baseline for different models and images from the raw rating matrix $\mathbf{R}$ to make values in the matrix comparable. In [14], a simple additive model is proposed to represent each value



(a) Correlation among models (b) Max Operation (c) Average Operation

(d) Illustration of latent factor

Figure 2: (a) Correlations across the models for the multi-view toy dataset, there are in total 1440 models from 10 sequences. The block diagonal structures refer to the high linear correlations among responses of close-view models; (b) Matrix $\mathbf{R}$ in which the pattern of color-coded elements R_{ij} reveals the strong correlations across models and images when the max operation is used to compute the ratings; (c) The average operation does not reflect the strong correlations (d) Value of the first latent factor obtained after factorizing $\mathbf{R}$ (red plot) for each training image (a few of the training images are shown at the top of the plot), showing that the latent factor does correspond to the viewing angle.

of matrix R as:

$$R_{ij} = \tilde{R}_{ij} + \mu + \alpha_i + \beta_j, \tag{4}$$

where μ is the global baseline of the matrix $\mathbf{R}$, α_i is the baseline response for model i, and β_j is the baseline response for image j. Baselines μ, α_i and β_j can be solved as a least square problem by minimizing the square error $\sum_{ij} ||R_{ij} - \mu - \alpha_i - \beta_j||^2$. $\tilde{R}_{ij}$ is called *residual* and we aim at predicting this residual using collaborative filtering.

Factorization: With rating matrix $\tilde{\mathbf{R}}$, CF discovers the structure in the matrix by transforming responses from both models and images into a latent factor space. The latent factors try to explain the elements of $\tilde{\mathbf{R}}$ by characterizing the tasks and models in a shared space. For example, there might be a dimension in the latent factor space characterizing the angle of viewpoint whereas another dimension characterizes the illumination condition. The latent factors are estimated by factorizing the rating matrix $\tilde{\mathbf{R}}$: $\tilde{\mathbf{R}} = \mathbf{\Theta}^T\mathbf{\Omega}$, where each column of $\mathbf{\Theta}\in\mathbb{R}^{K\times M}$ is the latent feature representation for each model, and K is the number of latent factors. Similarly, each column of $\mathbf{\Omega}\in\mathbb{R}^{K\times N}$ is the latent representation for each image. The most direct way to solve the above factorization problem is to use Singular Value Decomposition: $\tilde{\mathbf{R}} = \mathbf{U}\mathbf{D}\mathbf{V}^T$. We can get the desired factorization by setting $\mathbf{\Theta}^T = \mathbf{U}'\mathbf{D}'$ and $\mathbf{\Omega} = \mathbf{V}'^T$ ($\mathbf{U}'$ is the first K columns of $\mathbf{U}$, $\mathbf{D}'$ is the upper-left square matrix with dimension K from $\mathbf{D}$ and $\mathbf{V}'$ is the first K columns of $\mathbf{V}$). The latent factors obtained through SVD give us strong semantic patterns for both models and images. As shown in Figure 2d, the curve is the magnitude of the first latent factor, which clearly corresponds to the view angle of exemplars.

Prediction: Given an testing image I, we select a set of probes $\mathcal{S}_p$ with size $|\mathcal{S}_p|$ and apply models in $\mathcal{S}_p$ on I to get a probe response vector $\mathbf{p}\in\mathbb{R}^{|\mathcal{S}_p|}$. $\mathbf{p}$ is then normalized as $\tilde{\boldsymbol{p}} = \mathbf{p} - \mu\mathbf{I} - \boldsymbol{\alpha}_p - \boldsymbol{\beta}_p$, where $\boldsymbol{\alpha}_p = \{\alpha_i | i\in\mathcal{S}_p\}$, $\boldsymbol{\beta}_p = \frac{1}{|\mathcal{S}_p|}\sum_j (p_j - \mu)$. We recover the latent feature representation of I from $\tilde{\boldsymbol{p}}$ by:

$$\mathbf{\Theta}_p^T\boldsymbol{\omega}_p = \tilde{\boldsymbol{p}}, \tag{5}$$

where $\mathbf{\Theta}_p$ contains corresponding columns extracted from $\mathbf{\Theta}$ which includes representation of all the models in latent factor space. Multiplying $\boldsymbol{\omega}_p$ with $\mathbf{\Theta}$ we get the residual prediction $\tilde{r}_p$, which we convert to predicted ratings by ap-

plying the baseline transformation: $\mathbf{r}_p = \tilde{\mathbf{r}}_p + \mu\mathbf{I} + \boldsymbol{\alpha} + \boldsymbol{\beta}_p$. Once the model responses on the testing image have been predicted using the above method, ideally we can pick the model that has the highest predicted response to run on the testing image and get detection results. However, it is hard for collaborative filtering to precisely locate the *best* model. Instead, we choose the top K models to form a candidate set and apply models in this set on the testing image[1].

3. Results

Since we are aiming at recommending models for object instance detection. We present experiments on two object instance datasets as Multi-View Toy and RGBD datasets, the first one is a dataset we collect with multiple views and illumination conditions for a particular toy object whereas the second one is a commonly used object instance dataset [16, 17, 1]. We also demonstrate the capability of our approach tackling larger intra-variance by presenting a set of results on PASCAL Car dataset which comprises different car instances.

3.1. Multi-View Toy Dataset

In this dataset, we collected image sequences by fixing the camera at 5 different arbitrarily chosen height and distances from the object instance, and we vary the illumination by turning on/off a lamp above it. The object is set on a turnable table which enables us to collect images of the toy from different views. As a result, we get 10 sequences with lengths ranging from 1006 to 1330 frames. For ESVM training, we sample the training data from each sequence at a rate of 2.5 degree/image, which leads to 144 exemplars per sequence and 1440 exemplars total. The rating matrix is computed by evaluating all 1440 models on another 4320 images sampled from these 10 sequences (432 images per sequence). For testing, we collect 2 sequences (with a different height and distance from the training samples, one with lamp illumination, the other without it) with slight clutter and occlusions. In total, there are 1000 images extracted from these two sequences for testing. Typical training and testing images can be seen in Figure 2d. Note that, in this setting, a naive approach would either apply all 1440 detectors to the testing image or would reduce the number of detectors at training time. The former would lead to high computational cost, whereas the latter would select a one-size-fits-all set of detectors, ignoring the testing image. In contrast, we show in this experiment that we can use a small set of detectors selected in an *input-sensitive* manner and still maintain detection accuracy.

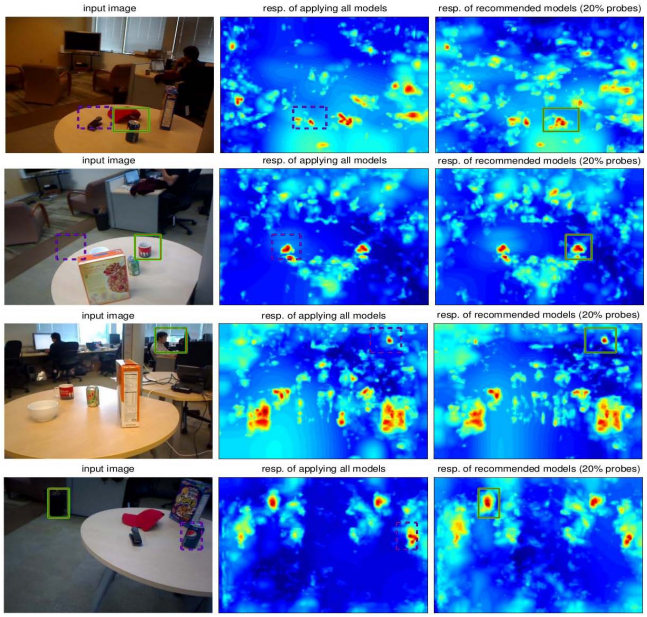
To measure the performance of detections, we follow the standard scoring method used in the PASCAL object detection challenge [6] in which all the detections are assigned their intersection-over-union scores between estimated bounding boxes and ground-truth bounding boxes. All the detections with overlapping scores higher than 0.7 are considered true positive detection, instead of 0.5 which is used for evaluating performance on PASCAL dataset. We report the *average precision* (AP), which is an approximation of the area under the precision-recall curve, with regard to the number of models used as probes. For comparison, the baseline is to randomly sample a model subset with the same number of models as in the probe set and directly apply them on testing images. To show how the detection performance approaches the standard testing procedure, we also plot out the performance obtained by applying all the models on the testing image. The performance of object detection using model recommendation is shown in Figure 6a. As shown in the figure, it is possible to achieve detection performance comparable to the performance of applying all the models (green line) with model recommendation (blue line) using a small fraction (5%) of models. The reason is that the detectors are dynamically selected by combining the prior experience from the ratings matrix with the information from the testing image. The performance curve shown in Figure 6a is the average of 20 rounds.

3.2. RGB-D Object Dataset

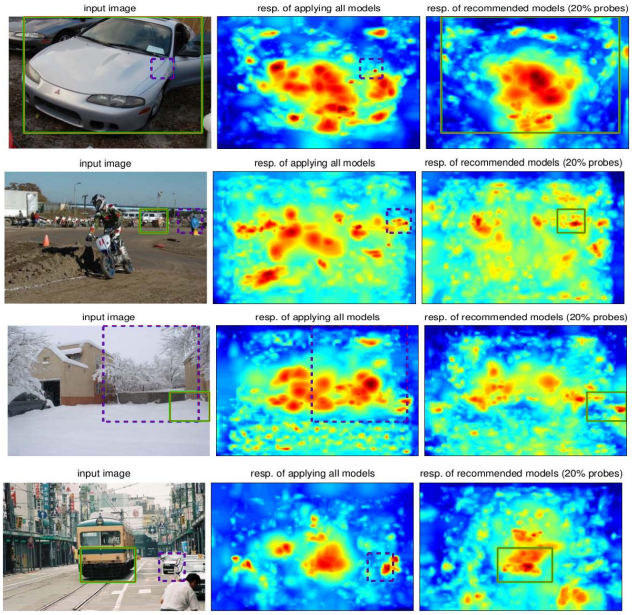
The RGB-D Object Dataset is a large data collection consists of 300 daily objects classified into 51 categories [16]. One notable difference between this dataset and classical object datasets like Caltech 101 and ImageNet is that objects in this dataset are labeled on two different levels: category level and instance level. For example, for the soda can category, the ground-truth labels such as *Pepsi Can* and *Mountain Dew Can* are provided in the annotations. In our experiment, we focus on instance-level detection. Specifically, we pick one instance per category[2] to test our model recommendation method. Since the consecutive frames are very similar, we have sampled images every 5 frames as training data for ESVMs which results in 111, 127, 128 and 118 models of different views for coffee mug, soda can, cap, and bowl, respectively. The collaborative matrices for different instances are then computed by evaluating all the models belonging to the instance on uncropped images for the corresponding category (i.e., we have model i evaluated on all instances from the category to which model i belongs). The precision-recall curves for these four instances are reported in Figure 4. It can be seen that the exemplar models produce robust results using only RGB images without depth information. As shown in Figure 6c,6e,6d,6f, for different object instance, the model recommendation method requires different probe ratios to achieve a performance comparable to the performance of applying

[1] In all our experiments, we set K to 20.

[2] The index of these four instances are coffee mug #1, soda can #1, cap #4 and bowl #4.



(a) RGBD Dataset



(b) PASCAL Car

Figure 3: Figure 3a and 3b show results on the RGBD and PASCAL Car Datasets, respectively. The first two rows show two examples of successful detections. The next two rows correspond to two failed cases. The first failed case corresponds to the case where both exhaustive matching (applying all the models) and model recommendation fails whereas the second one is the case where model recommendation fails but exhaustive matching gives a successful detection.

all the models. This is because different instances have different variations when inspected from different views and thus make the correlation among models for views vary from instance to instance. However, it is always possible

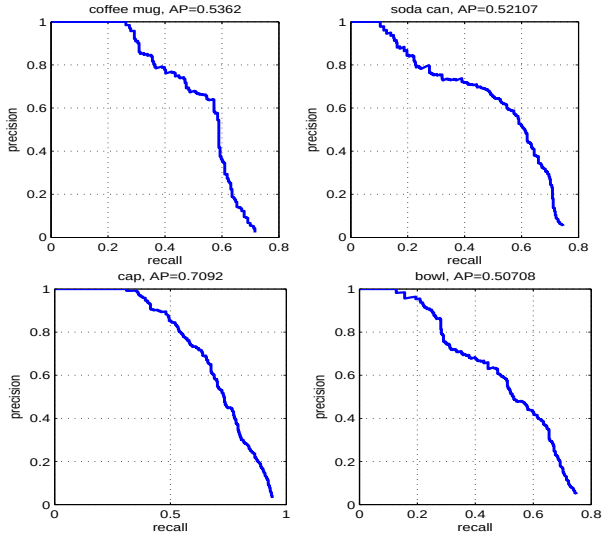


Figure 4: Precision-recall curve for four selected instances in RGB-D Object Dataset.

for the recommendation system to reach good performance using a relatively small number of probes.

In Figure 5, the x-axis is the ratio of probes and the y-axis is the proportion of testing images for which model recommendation successfully recommended the highest response model in the candidate set. For clarity of comparison, we also plot out the baseline, which is a straight line going through origin and $(1, 1)$. This result shows that the model recommendation gives us accurate recommendation with a small number of probes. Note that for the bowl instance, the model recommendation system actually outperforms the result of applying all the models. The reason for this is that model recommendation filters out many false positives when responses among models have strong correlations with each other.

3.3. PASCAL2007 Car Dataset

Experiments on Multiview Toy Dataset and RGB-D Object Dataset demonstrate the capability of model recommendation for object instance detection. We also conduct experiments on PASCAL dataset to see whether our method can generalize to the case of larger variations among models. We use the *car* category in PASCAL 2007 *trainval* set which includes 1250 instances for different cars from different views with clutter and occlusions as our exemplars. As can be seen from Figure 1, the PASCAL Car dataset has much larger variation among models than the datasets we use in previous experiments. The rating matrix is computed by evaluating all the exemplar models on images of the *Motor vehicle, Automotive vehicle* entry from the large-scale dataset *Imagenet*. This set includes 1748 images with each

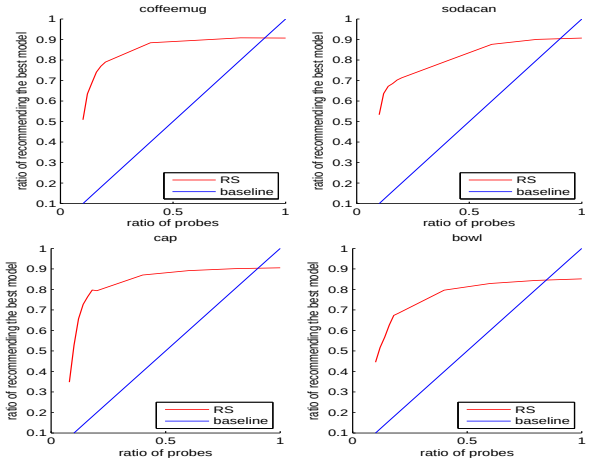


Figure 5: The x-axis is the ratio of probes and the y-axis is the proportion of testing images for which model recommendation successfully recommend the highest-response-model into the candidate set.

containing at least one car instance[4]. Note that we do not use use any object-level annotation like bounding boxes in the process generating the rating matrix. The testing images are taken from the PASCAL 2007 *test* set which contains 4952 images. The result of model recommendation is shown in Figure 6b. For average precision, we report the raw detection results without adding the steps of score calibration and context rescoring [19] for both the baseline and our method. We demonstrate that, even with larger variations among models, model recommendation is able to suppress false positives and thus provide better performance using a comparatively small number of probes.

Experiments on these three datasets consistently demonstrate the capability of model recommendation for object detection. Model recommendation provides a way of reducing the number of models applied to images and thus enables the model set to scale up when detecting objects using exemplar models. When model responses have strong linear correlations, it is even possible for model recommendation to yield better performance than applying all the models because it filters out false positive detections.

4. Discussion

In this paper, we explore the use of model recommendation for exemplar-based object detection. A subset of the exemplar models that we refer to as *probe set* is applied to the testing image, and the responses of the probes are used to predict responses of all the models on the testing image. A recommended set of models with the highest estimated responses is then applied to the testing image to obtain the final detections. The key to this approach is to combine offline information about the responses of a large pool of detectors on a large set of images with input-sensitive response of a few detectors. With such an approach, it is possible to *scale up* the number of exemplar models while keeping the computation at a tractable level. In addition to tractability, we observe that for some tasks it is possible for model recommendation to outperform the results of applying all the models. By leveraging information from responses of different models, model recommendation method is able to contextually predict which models are likely to fire on the testing image and thus helps us to avoid false positive detections. A future direction of research is to integrate the use of meta-information, such as image-tags or scene priors, to condition the recommendation.

References

[1] L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1729–1736. IEEE, 2011. 5

[2] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *Computer Vision–ECCV 2010*, pages 168–181. Springer, 2010. 1

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 1

[4] 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*, pages 248–255. IEEE, 2009. 7

[5] S. K. Divvala, A. Efros, and M. Hebert. Object instance sharing by enhanced bounding box correspondence. 2011. 1

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 5

[7] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1, 3

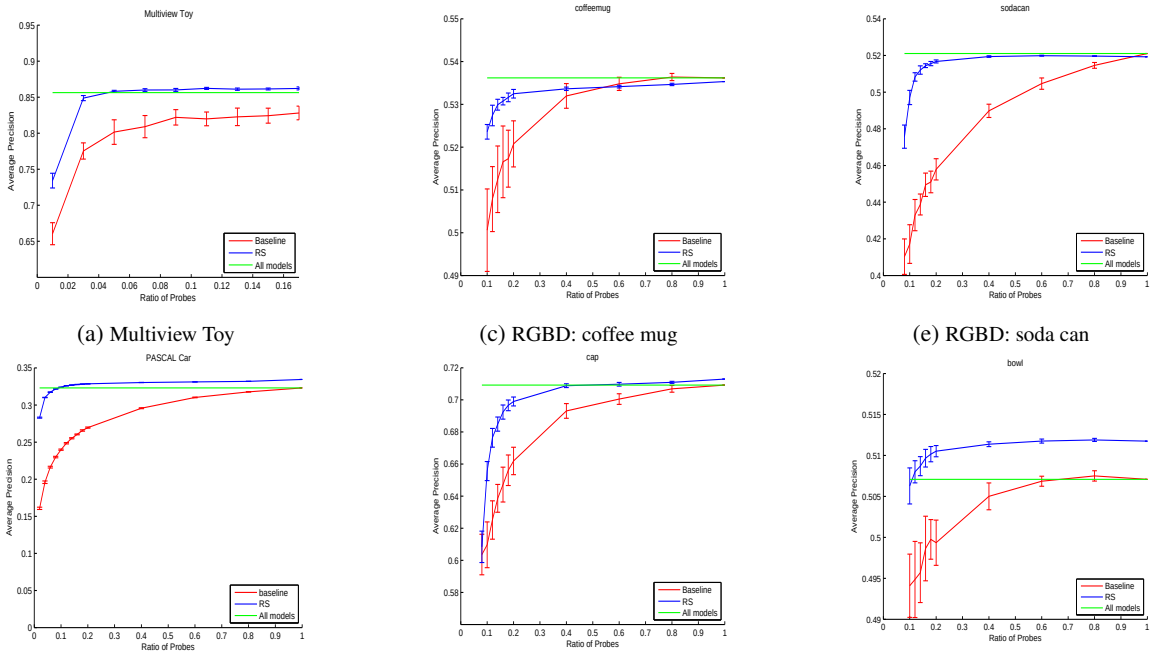
[8] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 2241–2248. IEEE, 2010. 1

[9] T. Gao and D. Koller. Active classification based on value of classifier. In *NIPS*, volume 24, pages 1062–1070, 2011. 1, 2

[10] D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 87–93. IEEE, 1999. 3

[11] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *Computer Vision–ECCV 2010*, pages 408–421. Springer, 2010. 1

[12] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *Computer Vision–ECCV 2012*, pages 459–472. Springer, 2012. 3



(a) Multiview Toy (c) RGBD: coffee mug (e) RGBD: soda can

(b) PASCAL Car (d) RGBD: cap (f) RGBD: bowl

Figure 6: Results for model recommendation, the x-axis is the fraction of models used as probes while the y-axis is the average of the estimation of *average precision* over 20 rounds.

[13] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2257–2264. IEEE, 2010. 1

[14] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):1, 2010. 3

[15] Y. Koren and R. Bell. *Advances in Collaborative Filtering*. Springer, 2011. 2, 3

[16] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011. 5

[17] K. Lai, L. Bo, X. Ren, and D. Fox. Sparse distance learning for object recognition combining rgb and depth information. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4007–4013. IEEE, 2011. 5

[18] T. Malisiewicz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[19] 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*, pages 89–96. IEEE, 2011. 1, 3, 7

[20] P. Matikainen, R. Sukthankar, and M. Hebert. Model recommendation for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2256–2263. IEEE, 2012. 2, 3

[21] B. C. Russell, A. Torralba, C. Liu, R. Fergus, and W. T. Freeman. Object recognition by scene alignment. In *In NIPS*. Citeseer, 2007. 1

[22] H. O. Song, T. Darrell, and R. B. Girshick. Discriminatively activated sparselets. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 196–204, 2013. 1

[23] 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*, pages 802–815. Springer, 2012. 1

[24] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 1, 2