Domain Adaptive Object Detection

Fatemeh Mirrashed[1], Vlad I. Morariu[1], Behjat Siddiquie[2], Rogerio S. Feris[3], Larry S. Davis[1]
[1]University of Maryland, College Park [2]SRI International [3]IBM Research
{fatemeh,morariu,lsd}@umiacs.umd.edu behjat.siddiquie@sri.com rsferis@us.ibm.com

Abstract

We study the use of domain adaptation and transfer learning techniques as part of a framework for adaptive object detection. Unlike recent applications of domain adaptation work in computer vision, which generally focus on image classification, we explore the problem of extreme class imbalance present when performing domain adaptation for object detection. The main difficulty caused by this imbalance is that test images contain millions or billions of negative image subwindows but just a few positive ones, which makes it difficult to adapt to the changes in the positive class distributions by simple techniques such as random sampling. We propose an initial approach to address this problem and apply our technique to vehicle detection in a challenging urban surveillance dataset, demonstrating the performance of our approach with various amounts of supervision, including the fully unsupervised case.

1. Introduction

Building visual models of objects robust to extrinsic[1] variations such as camera view angle (or object pose), resolution, lighting, and blur has long been one of the challenges in computer vision. Generally, a discriminative or generative statistical model is trained by acquiring a large set of examples, extracting low-level features which encode shape, color, or texture from the segmented or cropped objects, and finally, training the model (usually a classifier) using the extracted features vectors. Applied to a test image, the trained model generally works if the training set was representative of the test set and of the particular test image, e.g., if the training set contained a sample of the object with the same pose under similar lighting with similar resolution. Unfortunately, there are often cases when this assumption is violated, resulting in a sharp performance drop.

Recently, the machine learning community has focused on cases when these assumptions are violated as part of the problem of domain adaptation, seeking to develop effective mechanisms to transfer or adapt knowledge from one domain to another related domain. While these advances have also been applied by the computer vision community with promising results [22, 17, 15, 1], object models are still being trained and tested on images consisting of only one object zoomed and cropped at the center of a relatively uniform background. As a result, in such experimental settings the general problem of object detection is reduced to that of image classification. While domain adaptation is a challenging problem for image classification, it becomes even more challenging for object detection when target domain labels are unavailable and the majority of the image is occupied by the background class (a random sampling will not be sufficient for effective domain adaptation).

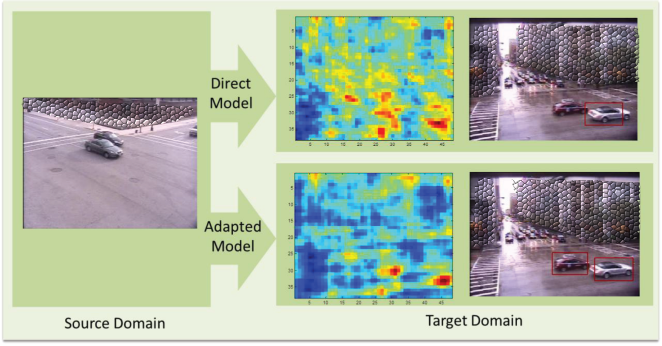


Figure 1. An example of the effects of domain change for the task of vehicle detection and our improved results after domain adaptation. Here, the vehicle detector is trained on training data, the *source domain*, and is applied to testing data (a new domain) that differs from the training data in various ways, e.g., viewing angles, illumination. If we directly apply the trained model to a new domain, the confidence map has multiple peaks, many of which do not correspond to vehicles. After domain adaptation, the highest peaks correspond to the two vehicles in the foreground. (Note: Background regions have been obfuscated for legal/privacy reasons)

We focus on domain adaptation applied to vehicle detection in urban surveillance videos, where the backgrounds, numbers, and poses of the objects of interest are uncontrolled and vary highly. The detection and localization of

[1] as opposed to intrinsic or intra-class variation of an object category with respect to different shapes, sizes, textures, colors, etc.

vehicles in surveillance video, which is typically low resolution, is difficult as it requires dealing with varying viewpoint, illumination conditions (e.g. sunlight, shadows, reflections, rain, snow) and traffic, where vehicles tend to partially occlude each other. These conditions are localized in space and time, allowing us to model realistic domain changes by considering two cameras at different locations and points in time as the source and target domain. As we will demonstrate, the changes between some domains are sufficiently large that the classifier trained on the source domain performs quite poorly. By applying recent domain adaptation techniques, we obtain significant improvements in these cases (Fig 1).

We use Transfer Component Analysis (TCA) [20], an unsupervised domain adaptation and dimensionality reduction technique, to learn a set of common transfer components underlying both domains such that, when projected onto this subspace, the difference in data distributions of two domains can be dramatically reduced while preserving data properties. Standard machine learning algorithms can then be used in this subspace to train classification or regression models across domains. While TCA obtains the transfer components by aligning distribution properties that are not class-aware, i.e., it does not guarantee that the same class in separate domains will project to the same coordinates in the shared subspace, we find that for our problem this alignment yields impressive results. Many other domain adaptation or transfer learning techniques can be applied instead of TCA, but given the surprisingly good performance of TCA on our dataset, we leave a comparison between other potential techniques for future work, focusing instead on studying why TCA works as well as it does on our data.

Our contributions are the following:

- we evaluate a domain adaptation technique, TCA, applied to vehicle detection on a challenging dataset and find that it is surprisingly effective despite its simplicity
- we provide insights into what makes TCA perform so well on our dataset by comparing to basic machine learning techniques (e.g., PCA)
- we propose an initial approach to selecting target samples for domain adaptation in a more general object detection setting (multiple object instances, objects are generally not centered, and the image consists of mostly background)

The remainder of the paper is organized as follows. We review related literature in Section 2, followed by a detailed description of our proposed approach in Section 3. We describe the experiments and results in Section 4 and finally conclude in Section 5.

2. Related Work

Object category recognition and detection approaches that are invariant to view and other extrinsic changes have long been sought by researchers in computer vision [16]. Several methods address changes in view by learning separate appearance models for a small number of canonical poses corresponding to each object category [13, 14]. Other approaches employ parts-based models, which model variations in part appearance and inter-relationships over multiple views [23, 25, 26, 24]. Recently, Gu and Ren [14] proposed a discriminative approach based on a mixture of templates, achieving the best performance on two different 3D object recognition datasets. Unfortunately, this performance gain is achieved at up to an order of magnitude higher cost–depending on the number of templates used–than a comparable view-specific method that employs a similar feature representation.

The problem of learning object models that can generalize to new views or domains is closely related to the problems of transfer learning [21] and domain adaptation [10], the two main groups of work that address the effects of domain change in machine learning. In general, a domain consists of the input data feature space and an associated probability distribution over it. If two domains are different, they may have different feature spaces or different marginal probability distributions. The problem of *domain adaptation* addresses domain changes, when the marginal distribution of the data in the training set (source domain) and the test set (target domain) are different but the tasks or conditional distributions of some additional variables, or labels, given the data are assumed to approximately the same. The problem of *transfer learning* addresses situations in which marginal distributions of data between the domains are the same but either the feature spaces or conditional distribution of the labels given data are different.

The natural language processing community has lately paid considerable attention to understanding and adapting to the effects of domain change. Daume et al [10] model the data distribution corresponding to source and target domains as a common shared component and a component that is specific to the individual domains. Under certain assumptions characterizing the domain shift, there have also been theoretical studies on the nature of classification error across new domains [4, 2]. Blitzer et al [8, 7] proposed a structural correspondence learning approach that selects some pivot features that would occur frequently in both domains. Ben-David et al [3] generalized the results of [8] by presenting a theoretical analysis on the feature representation functions that should be used to minimize domain divergence, as well as classification error, under certain domain shift assumptions. Insights related to this line of work were also provided by [6, 19]. Wang and Mahadevan [9] pose this problem as unsupervised manifold alignment, where source and target manifolds are aligned by preserving a notion of the neighborhood structure of the data points.

In visual object recognition, there is less consensus on the basic representation of the data, so it is unclear how reasonable it is to make subsequent assumptions on the relevance of extracted features [8] and the transformations induced on them [9]. However, there have been recent efforts focusing on domain shift issues for 2D object recognition applications. For instance, Saenko et al [22] proposed a metric learning approach that can use labeled data for a few categories from the target domain to adapt unlabeled categories to domain change. Bergamo and Torresani [5] performed an empirical analysis of several variants of SVM for this problem. Lai and Fox [18] performed object recognition from 3D point clouds by generalizing the small amount of labeled training data onto the pool of weakly labeled data obtained from the internet. Gopalan et al [1] take an incremental learning approach, following a geodesic path between the two domains modeled as points on a Grassmann manifold.

We extend recent work by applying a domain adaptation technique, TCA [20], to the problem of object detection. We study the effects of varying amounts of balanced target domain training samples, similar to the classification setting of [22, 17, 15, 1], and we also explore the automatic acquisition of training data from the target domain, which is more applicable to the detection problem. In the detection setting, the class labels are unavailable, and the classes are highly imbalanced since the majority of windows in the image contain background and only a few are good examples of the object class.

3. Proposed Method

3.1. Formulation

Following the notation of Pan et al. [20], we define a domain to consist of a feature space and a distribution $P(X)$, defined over a set of examples $X = \{x_1, \ldots, x_n\}$ from the feature space. The examples in X have a corresponding set of labels $Y = \{y_1, \ldots, y_n\}$. While domains can differ both in the feature space and in the marginal distribution, we consider only the case where the feature space remains constant across domains. Given training features X_S and labels Y_S from the source domain and training features X_T from the target domain, our task is to learn a model that can predict the labels on new samples from the target domain.

While most domain adaptation methods assume that $P(X_S) \neq P(X_T)$ and that $P(Y_S|X_S) = P(Y_T|X_T)$, TCA [20] replaces the second assumption with a more realistic one, that probability density $P(Y|X)$ may also change across domains, but that there exists a transformation ϕ such that $P(\phi(X_S)) \approx P(\phi(X_T))$ and $P(Y_S|\phi(X_S)) \approx P(Y_T|\phi(X_T))$. Based on these assumptions and given X_S and X_T, TCA obtains the transformation ϕ. A classifier can then be trained on transformed features $\phi(X_S)$ and labels Y_S and applied to transformed out-of-sample target features $\phi(X_T^o)$ to predict labels Y_T^o.

3.2. Transfer Component Analysis

Given training samples from two domains, X_S and X_T, TCA [20] obtains a transformation ϕ to a latent space that minimizes the distance between the transformed distributions while preserving properties of both input feature spaces. This optimization is performed in a reproducing kernel Hilbert space (RKHS), under the assumption that ϕ is a feature map which defines a universal kernel. The distance between the transformed distributions is measured by the empirical estimate of Maximum Mean Discrepancy (MMD):

$$MMD(X_S, X_T) = ||\frac{1}{n_1}\sum_{i=1}^{n_1}\phi(x_{S_i}) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(x_{T_i})||^2,$$

where n_1 and n_2 are the number of samples in X_S and X_T, respectively, and the norm is the RKHS norm. Properties of the input feature spaces are preserved by maximizing the variance of the transformed data.

Instead of directly optimizing for the feature map ϕ, TCA first applies a parametric kernel (e.g., linear or RBF) to obtain the kernel matrix $K = [k(x_i, x_j)] \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ of the source and target training samples, and then searches for a matrix $\widetilde{W} \in \mathbb{R}^{(n_1+n_2)\times m}$ that projects the empirical kernel map $K^{-1/2}K$ to an m-dimensional space $\widetilde{W}^T K^{-1/2} K$. Letting $W = K^{-1/2}\widetilde{W}$, the feature map ϕ induced by the kernel KWW^TK is thus optimized implicitly by the following constrained minimization:

$$\min_W \texttt{tr}(W^T KLKW) + \mu\texttt{tr}(W^T W)$$
$$\texttt{s.t.} \quad W^T KHKW = I.$$

Here, the MMD criterion is rewritten as $\texttt{tr}(W^T KLKW)$, where $L_{ij} = 1/n_1^2$ if $x_i, x_j \in X_S$, $L_{ij} = 1/n_2^2$ if $x_i, x_j \in X_T$, and $L_{ij} = -1/(n_1 n_2)$ otherwise. The term $\texttt{tr}(W^T W)$ is a regularizer that penalizes aribrarily complex solution, and in conjunction with the constraint that the projected data has unit covariance, $W^T KHKW = I$, where H is the centering matrix $H = I - 1/(n_1+n_2)\mathbf{1}\mathbf{1}^T$, it also results in projection directions that maximize data variance. The parameter μ controls the trade-off between minimizing the distance between distributions and maximizing data variance. As Pan et al. [20] demonstrate, this optimization problem can be reformulated without constraints as

$$\max_W \texttt{tr}((W^T(KLK + \mu I)W)^{-1} W^T KHKW).$$

This optimization problem is solved by obtaining the m leading eigenvectors of $(KLK + \mu I)^{-1}KHK$. A new sample x_o is mapped into the latent space by computing $W^T[k(x_1, x_o), \ldots, k(x_{n_1+n_2}, x_o)]^T$, where x_i are the training samples.

3.3. Unsupervised adaptation

For an object detector to adapt to a new domain using our proposed approach, a set of features from the target domain, X_T, is needed during the training stage. A straightforward unsupervised approach to obtaining such a set for a multi-scale sliding window detector would be to randomly select a number of windows that would be encountered during the detection process. However, this would yield a majority of windows from the negative class (which consists of millions or billions of windows when using conventional sliding windows techniques) and only a few if any (most likely poorly localized) positive samples, since there are usually only a few instances of the object of interest in an image. This would cause domain adaptation to adapt only to the background class, and not to the class of interest (while penalty parameters can be modified to deal with imbalance, e.g., C in SVM, that does not help when it is unclear which sample is a positive sample from a randomly sampled dataset). A potential solution would be to introduce a small amount of supervision into the process. Since our proposed approach does not use class labels during the domain adaptation step, it is only important that the classes are balanced by the user somehow to prevent the joint latent space from being dominated solely by the target background class. While this may be an acceptable solution, especially if it is sufficient for the user to annotate a very low number of examples (in our experiments we show that very little supervision is necessary for significant improvements), we are also interested in studying the fully unsupervised case.

In the absence of any supervision, we propose a scheme that relies on a detector trained on X_S and Y_S alone to extract positive and negative examples from the target distribution. Before performing domain adaptation, our scheme involves extracting the top and bottom scoring windows subject to some threshold (after non-maximal suppression), as the positive and negative samples to include in X_T. While a detector trained on the source domain alone would not be very accurate, we expect that regions of very high confidence are more likely to contain the object of interest than the regions of low confidence. While the detection rate may not be high, labels are not needed for the target training set, so labeling mistakes will not be very detrimental. Most importantly, we expect the resulting set of windows to contain more positive samples than if it were selected randomly.

4. Experiments and Results

4.1. Data Set Collection

We collected more than 400 hours of video from 50 different traffic surveillance cameras, located in a large North American city, over a period of several months. We adopted a simple method to extract images of cars from these videos, for training our object detection models. We performed background subtraction and obtained the bounding boxes of foreground blobs in each video frame. We also computed the motion direction of each foreground blob using optical flow. Vehicles were then extracted using a simple rule-based classifier which takes into account the size and motion direction of the foreground blobs. The range of acceptable values of the size and motion-direction were manually specified for each camera view. We manually removed the accumulated false positives. This simple procedure enabled us to collect a large number of images of vehicles(about 220000) in a variety of poses and illumination conditions, while requiring minimal supervision. We utilized the motion direction of each foreground blob for categorizing the images of vehicles of each camera viewpoint into a set of clusters. The clustering of images leads to categorization of the training images into a two level hierarchy, where the first level of categorization is according to the camera viewpoint and the second level is based on the motion-direction within each camera viewpoint. Since all the camera viewpoints are distinct, each leaf node of our hierarchy consists of training images of vehicles in a distinct pose. On an average, each camera viewpoint has about two clusters, resulting in a total of about 99 clusters (leaf nodes of the hierarchy). These clusters, which we call domains, cover an extremely diverse collection of vehicles in different poses, lightings and surroundings. Fig. 2 shows a few examples of the average images of the 99 training domains.

In order to evaluate our approach with respect to object detection, we annotated a set of 1616 frames collected from 21 out of the same 50 cameras that were used for collecting the training data. From each camera viewpoint, frames were collected at different times of the day and contain large variations in illumination due to the changes in the direction of sunlight and the resulting reflections and shadows from buildings. Apart from the viewpoint which changes significantly across the cameras, the amount of traffic also varies. On an average each test image contains between one to three vehicles.

4.2. Image Classification

Since our sliding-window detection approach applies a binary classifier at each window location, we first evaluate the performance of TCA on visual domains by conducting binary classification on our training set of car and background images from 99 domains. For these experiments,

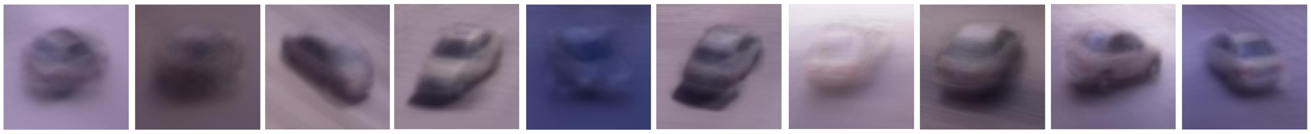


Figure 2. A few examples of the training domains presented here by their average images. Note the variations in pose and illumination across domains.

the classification performance is measured by average precision. For all experiments throughout this paper, we used HOG features (as implemented by [13]) with a dimension of 55,648 to represent images and an SVM with linear kernel (as implemented by *LibLinear* [12]) as the classifier. For the baseline method, we trained the classifier on images from only one of the 99 domains (source domain), and tested it on all the images of the other domains (target domains) without any adaptation. For the cases where the source and target domains were the same, the images were split into half for training and testing. We perform the same procedure for our proposed method but instead trained and tested the classifier on feature vectors projected onto the latent subspace learned by TCA, using a linear kernel and $\mu = 1$ for all experiments. The dimension of the subspace (m) was set to 15 for all the experiments. This selection was done based on the results of a set of pilot experiments in which we varied the values of m from 5 to 500 and observed that classification performance starts to degrade when m is below 10. As shown in Fig. 3 even with the decreasing number of unlabeled samples of the target domain from 50% down to only 10 random samples, the adapted classifier can still outperform the baseline in majority of cases. Once the number of target samples is decreased to 2, we are no longer able to improve performance.[2] Of particular note is that our proposed domain adaptation approach is able to drastically improve results even when the baseline performance is close to chance, often improving performance close to an average precision of 1.

4.2.1 Comparison with Principal Component Analysis (PCA)

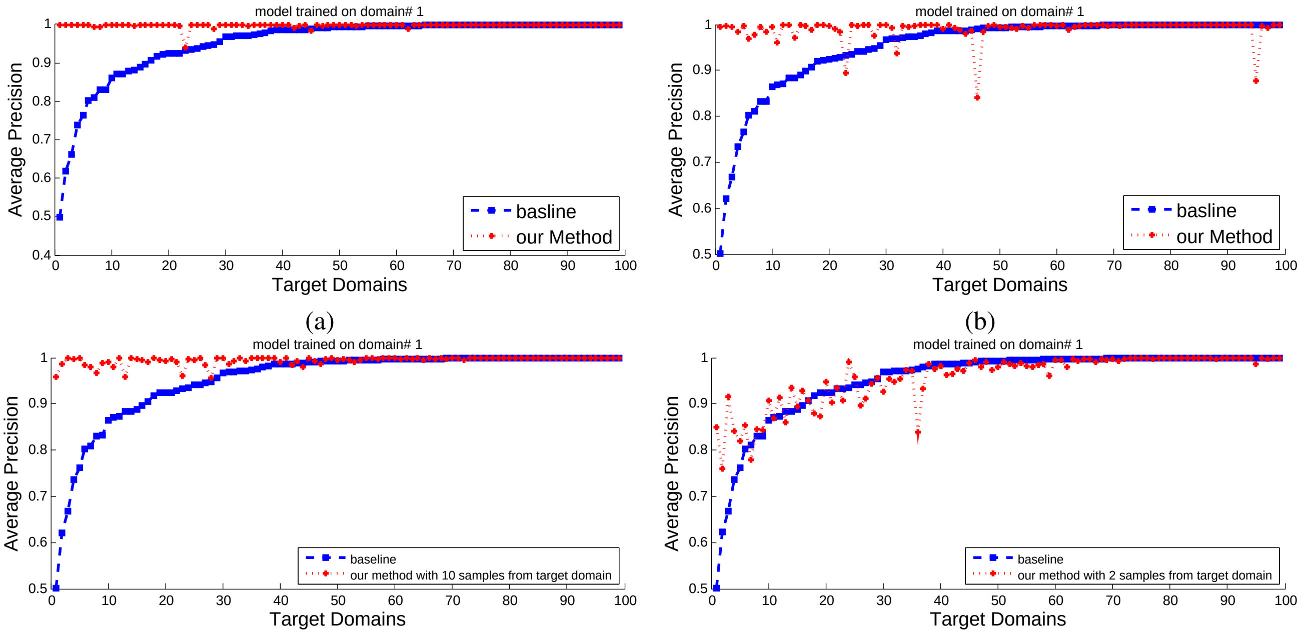
While any domain adaptation approach could be applied to the object detection task, TCA performs surprisingly well aligning means and maximizing data variance with a linear kernel. In an attempt to understand which TCA components lead to these results (dimensionality reduction, mean alignment, variance maximization), we compare to a number of alternatives based on Principal Component Analysis (PCA). We are especially interested in the cases where domains are so different from each other that directly applying the classifier trained on the source domain alone yields classification rates close to chance. As described in section 3, the effect of TCA is threefold: 1) the means in the RKHS are close to each other, 2) data variance is maximized, and 3) the dimensionality of the input data is reduced prior to classifier training. Since PCA obtains a subspace in which the variance of the projected data is maximized, it produces two of the three effects of TCA. In addition, we note that if the MMD criterion is removed from the TCA optimization, the result is that only variance is maximized (as for PCA), but that the formulation ensures that the *projected* data has unit covariance, whereas the standard implementation of PCA yields orthonormal projection vectors instead. To eliminate this difference, we *whiten* [11] the PCA projected data as a post-processing step to ensure that its covariance is also a unit matrix. Figure 4 shows the performance of our baseline approach, TCA, PCA, and PCA with whitening as a post-processing step. Interestingly, performing PCA provides an improvement in performance, but at the cost of some negative transfer in some easy cases. The whitening post-processing step mimicks the results of TCA very closely (although TCA still outperforms), removing most spurious negative transfer cases. While we also performed experiments (not shown) where the means of the source and target distributions were removed to align them exactly, we did not observe an improvement in performance as we did for PCA and PCA with whitening. These preliminary results lead us to believe that it is the combination of dimensionality reduction and whitening which contribute most to the improved adaptation to domain change.

4.3. Object Detection

Here we present the results of running the classifiers trained as described in section 4.2, at multiple scales and in a sliding window detection fashion on our test data set. For our semi-supervised approach, we use 100 positive and negative samples from the target domain for domain adaptation. We perform experiments by applying each of the 99 training domains to each of the 21 testing domains, yielding 99 × 21 possible testing scenarios. Figure 5 shows the performance graphs for two examples of these experiments.

For our proposed method of unsupervised adaptation, where a balanced set of target samples are obtained automatically, we select a subset of testing scenarios where

[2] Note that when only 2 or 10 samples are chosen from the target domain, we repeat the experiment 20 and 10 times, respectively, to remove the effects of selecting a few bad target samples.



(a) (b)

(c) (d)

Figure 3. A comparison of performance of the baseline classifiers with the adapted ones. To simplify visualization, the results have been sorted by the average precisions of the baseline classifiers. Adaptation by (a) 50%, (b) 10%, (c) 10, and (d) 2 samples from the target domains.

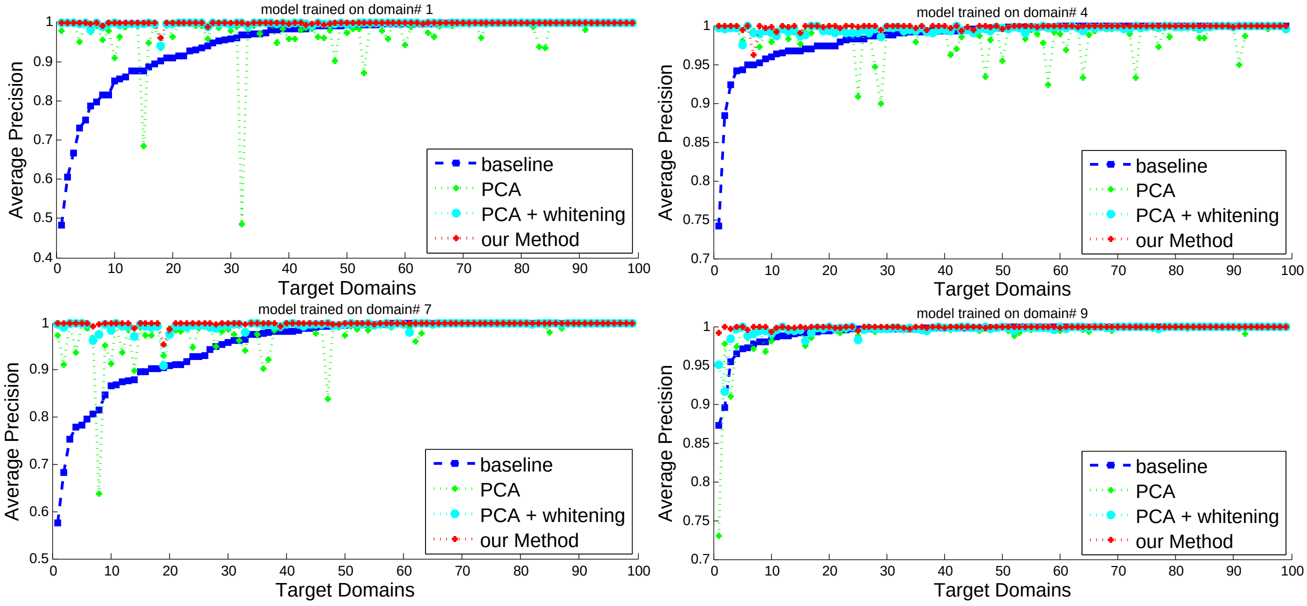
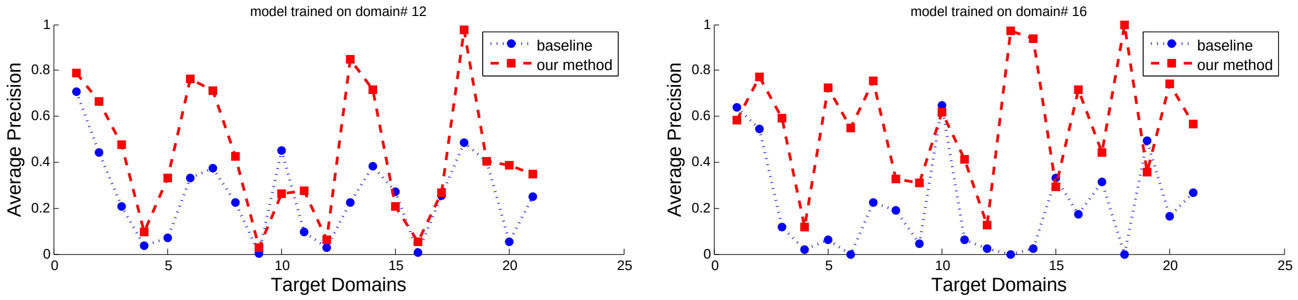


Figure 4. Performance comparison of TCA with PCA and PCA plus whitening on four test domains. By performing PCA and then whitening the projected data, we are able to match much (though not all) of the performance improvement of TCA.

the performance increase by our semi-supervised adaptation approach, in which the target samples are obtained manually, is most pronounced. We focus on these examples because we expect them to be the most difficult ones for our unsupervised approach, since it relies on first applying the baseline algorithm (which is not adapted to the new domain), and it is in these examples that the baseline algorithm is performing the worst.



(a)

(b)

Figure 5. Comparison of performance between baseline detectors and the semi-supervised adapted ones. It showcases two examples of the 99 graphs resulted from the 99 x 21 testing scenarios.

Table 1. Averaging of the detection results with semi-supervised adaptation over all the target domains

Average Precision-baseline detector	0.26
Average Precision-detector with semi-supervised adaptation	0.41
Average performance improvement	61.28%

Table 2. Averaging of the results presented in Figure 6 over all the target domains

Ave. Prec.-baseline detector	0.09
Ave. Prec.-detector with unsupervised adaptation (1 to 12 target samples)	0.17
Ave. Prec.-detector with semi-supervised adaptation (1 to 12 target samples)	0.38

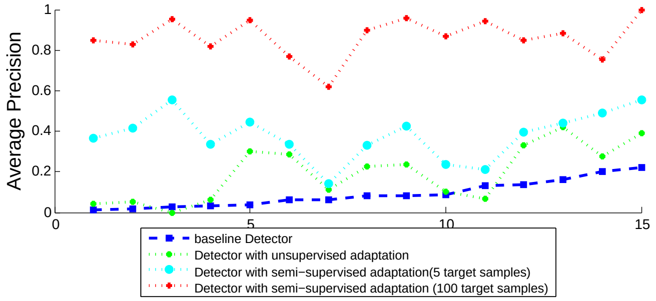


Figure 6. Comparison of different approaches of domain adaptation for detection

A limited number of positive and negative samples from the target domain (1-12 depending on results of the baseline detections) were automatically acquired by running the baseline detector on a few frames of the target domain. The most and least confident predictions by the baseline detectors were used correspondingly as positive and negative samples of the target domain. As Figure 6 shows, while the performance improvement obtained by unsupervised adaption (green curve) is lower than that of semi-supervised method (red curve), it still outperforms the baseline detector (blue curve) in majority of cases.

The difference in performance between the unsupervised and semi-supervised approaches can be a result of two factors: 1) poor quality positive and negative samples, and 2) fewer positive and negative samples from the target domain. To further investigate whether the degradation is a result of the reduced numbers or the poor quality of the samples from the target domain, we repeat the detection experiments with semi-supervised adaptation but restricted the semi-supervised approach to use same exact numbers of target samples as the ones obtained in the unsupervised approach. Comparing the restricted sample semi-supervised approach (cyan curve) to the unsupervised approach (green curve) in Figure 6, we observe that when the baseline classifier (blue curve) performs very poorly on the target domain (left side of the graph), the automatically obtained samples are too noisy for our adaptation method to work. However, it is very promising that our unsupervised approach begins to match the performance of the semi-supervised approach at a relatively low baseline average precision.

5. Conclusion and Future Work

We presented and evaluated an approach for domain-invariant vehicle detection in traffic surveillance videos. Although we demonstrated the effectiveness of our approach on the task of vehicle detection, it can be potentially applied to other object detection problems. Future work includes extending this model to multiple source domains, multiple object categories, and using class labels from the source or target domains when they are available.

Acknowledgements

This research was partially supported by ONR MURI grant N000141010934.

References

[1] R. G. ad R. Li and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011. 1, 3

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan. A theory of learning from different domains. *Machine learning*, 2010. 2

[3] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2007. 2

[4] S. Ben-David, T. Lu, T. Luu, and D. Pál. Impossibility theorems for domain adaptation. *AISTATS*, 2010. 2

[5] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: A domain adaptation approach. In *NIPS*, 2010. 3

[6] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *NIPS*, 2008. 2

[7] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007. 2

[8] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, 2006. 2, 3

[9] C.Wang and S. Mahadevan. Manifold alignment without correspondence. In *IJCAI*, 2009. 2, 3

[10] H. Daumé, III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 2006. 2

[11] R. Duda, P. Hart, , and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001. 5

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 5

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010. 2, 5

[14] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010. 2

[15] V. Jain and E. Learned-Miller. Online domain-adaptation of a pre-trained cascade of classifiers. In *CVPR*, 2011. 1, 3

[16] J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 1979. 2

[17] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011. 1, 3

[18] K. Lai and D. Fox. Object recognition in 3D point clouds using web data and domain adaptation. *International Journal of Robotics Research*, 2010. 3

[19] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. Technical report, 2009. 2

[20] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2), 2011. 2, 3

[21] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010. 2

[22] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 1, 3

[23] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007. 2

[24] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009. 2

[25] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *CVPR*, 2006. 2

[26] P. Yan, S. M. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *ICCV*, 2007. 2