Adapting Visual Category Models to New Domains

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell

UC Berkeley EECS and ICSI, Berkeley, CA
{saenko,kulis,mfritz,trevor}@eecs.berkeley.edu

Abstract. Domain adaptation is an important emerging topic in computer vision. In this paper, we present one of the first studies of domain shift in the context of object recognition. We introduce a method that adapts object models acquired in a particular *visual domain* to new imaging conditions by learning a transformation that minimizes the effect of domain-induced changes in the feature distribution. The transformation is learned in a supervised manner and can be applied to categories for which there are no labeled examples in the new domain. While we focus our evaluation on object recognition tasks, the transform-based adaptation technique we develop is general and could be applied to non-image data. Another contribution is a new multi-domain object database, freely available for download. We experimentally demonstrate the ability of our method to improve recognition on categories with few or no target domain labels and moderate to large changes in the imaging conditions.

1 Introduction

Supervised classification methods, such as kernel-based and nearest-neighbor classifiers, have been shown to perform very well on standard object recognition tasks (e.g. [4], [17], [3]). However, many such methods expect the test images to come from the same distribution as the training images, and often fail when presented with a novel *visual domain*. While the problem of *domain adaptation* has received significant recent attention in the natural language processing community, it has been largely overlooked in the object recognition field. In this paper, we explore the issue of domain shift in the context of object recognition, and present a novel method that adapts existing classifiers to new domains where labeled data is scarce.

Often, we wish to perform recognition in a *target* visual domain where we have very few labeled examples and/or only have labels for a subset of categories, but have access to a *source* domain with plenty of labeled examples in many categories. As Figure 1 shows, it is insufficient to directly use object classifiers trained on the source domain, as their performance can degrade significantly on the target domain. Even when the same features are extracted in both domains, and the necessary normalization is performed on the image and the feature vectors, the underlying cause of the domain shift can strongly affect the feature distribution and thus violate the assumptions of the classifier. Typical

(a)

train	test	SVM-bow	NBNN [3]
source	source	54 ± 2	61 ± 1
source	target	20 ± 1	19 ± 1

(b)

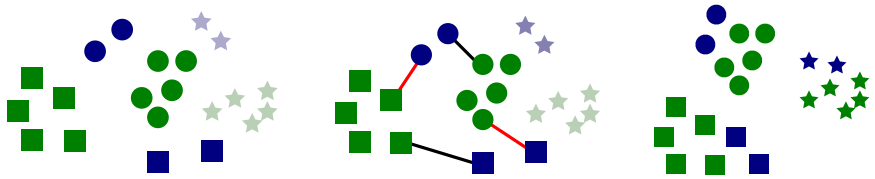
Fig. 1. (a) Example of extreme visual domain shift. (b) Degradation of the performance of two object classification methods (an SVM over a bag-of-words representation (SVM-bow) and the Naive Bayes nearest neighbor (NBNN) classifier of [3]) when trained and tested on these image domains (see Sec.4 for dataset descriptions). Classification accuracy is averaged over 31 object categories, and over 5 random 80%-20% splits into train/test data.

causes of visual domain shift include changes in the camera, image resolution, lighting, background, viewpoint, and post-processing. In the extreme case, all of these changes take place, such as when shifting from typical object category datasets mined from internet search engines to images captured in real-world surroundings, e.g. by a mobile robot (see Figure 1).

Recently, domain adaptation methods that attempt to transfer classifiers learned on a source domain to new domains have been proposed in the language community. For example, Blitzer et al. adapt sentiment classifiers learned on book reviews to electronics and kitchen appliances [2]. In this paper, we argue that addressing the problem of domain adaptation for object recognition is essential for two reasons: 1) while labeled datasets are becoming larger and more available, they still differ significantly from many interesting application domains, and 2) it is unrealistic to expect the user to collect many labels in each new domain, especially when one considers the large number of possible object categories. Therefore, we need methods that can transfer object category knowledge from large labeled datasets to new domains.

In this paper, we introduce a novel domain adaptation technique based on cross-domain transformations. The key idea, illustrated in Figure 2, is to learn a regularized non-linear transformation that maps points in the source domain (green) closer to those in the target domain (blue), using supervised data from both domains. The input consists of labeled pairs of inter-domain examples that are known to be either similar (black lines) or dissimilar (red lines). The output is the learned transformation, which can be applied to previously unseen test data points. One of the key advantages of our transform-based approach is that it can be applied over novel test samples from categories seen at training time, and can also generalize to new categories which were not present at training time.

We develop a general framework for learning regularized cross-domain transformations, and then present an algorithm based on a specific regularizer which



(a) Domain shift problem (b) Pairwise constraints (c) Invariant space

Fig. 2. The key idea of our approach to domain adaptation is to learn a transformation that compensates for the domain-induced changes. By leveraging (dis)similarity constraints (b) we aim to reunite samples from two different domains (blue and green) in a common invariant space (c) in order to learn and classify new samples more effectively across domains. The transformation can also be applied to new categories (lightly-shaded stars). This figure is best viewed in color.

results in a symmetric transform. This special case of transformations has previously been explored for metric learning, and we base the algorithm presented in this paper on the information theoretic metric learning method of [8]. Metric learning has been successfully applied to a variety of problems in vision and other domains (see [6, 11, 14] for some vision examples) but to our knowledge has not been applied to domain adaptation. In work subsequent to that reported in this paper, we have developed a variant of our method that learns regularized asymmetric transformations, which allows us to model more general types of domain shift[1].

Rather than committing to a specific form of the classifier, we only assume that it operates over (kernelized) distances between examples. Encoding the domain invariance into the feature representation allows our method to benefit a broad range of classification methods, from k-NN to SVM, as well as clustering methods. While we evaluate our technique on object recognition, it is a general adaptation method that could be applied to non-image data.

In the next section, we relate our approach to existing work on domain adaptation and transfer learning. Section 3 describes our general framework for domain adaptation and presents an algorithm based on symmetric transformations, i.e. metric learning. We evaluate our approach on a new dataset designed to study the problem of visual domain shift, which is described in Section 4, and show empirical results of object classifier adaptation on several visual domains in Section 5.

2 Related Work

The domain adaptation problem has recently started to gain attention in the natural language community. Daume III [7] proposed a domain adaptation approach

[1] See the technical report [15] for details of the method; for comparison results using this method are shown in the tables below.

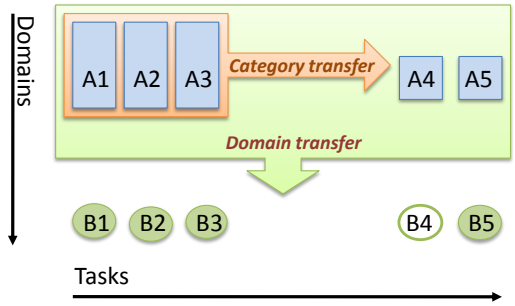


Fig. 3. Unlike category transfer methods, our method does not transfer structure between related tasks, but rather transfers the learned structure of the domain shift from tasks labeled in both domains (e.g. tasks 1,2,3 and 5 in the figure) to tasks unlabeled in the target domain (e.g. task 4), without requiring these tasks to be related.

that works by transforming the features into an augmented space, where the input features from each domain are copied twice, once to a domain-independent portion of the feature vector, and once to the portion specific to that domain. The portion specific to all other domains is set to zeros. While "frustratingly" easy to implement, this approach only works for classifiers that learn a function over the features. With normalized features (as in our experimental results), the nearest neighbor classifier results are unchanged after adaptation. Structural correspondence learning is another method proposed for NLP tasks such as sentiment classification [2]. However, it is targeted towards language domains, and relies heavily on the selection of *pivot* features, which are words that frequently occur in both domains (e.g. "wonderful", "awful") and are correlated with domain-specific words.

Recently, several adaptation methods for the support vector machine (SVM) classifier have been proposed in the video retrieval literature. Yang et al. [18] proposed an Adaptive SVM (A-SVM) which adjusts the existing classifier $f^s(x)$ trained on the source domain to obtain a new SVM classifier $f^t(x)$. Cross-domain SVM (CD-SVM) proposed by Jiang et al. [13] defines a weight for each source training sample based on distance to the target domain, and re-trains the SVM classifier with re-weighted patterns. The domain transfer SVM (DT-SVM) proposed by Duan et al. [9] used multiple-kernel learning to minimize the difference between the means of the source and target feature distributions. These methods are specific to the SVM classifier, and they require target-domain labels for all categories. The advantage of our method is that it can perform transfer of domain-invariant representations to *novel* categories, with no target-domain labels, and can be applied to a variety of classifiers and clustering techniques.

Our approach can be thought of as a form of knowledge transfer from the source to the target domain. However, in contrast to many existing transfer learning paradigms (e.g. [16], [10], [12]), we do not presume any degree of relat-

edness between the categories that are used to learn the transferred structure and the categories to which the structure is transferred (see Figure 3). Individual categories are related across domains, of course; the key point is that we are transferring the structure of the domain shift, not transferring structures common to related categories.

Finally, metric and similarity learning has been successfully applied to a variety of problems in vision and other domains (see [6, 11, 14, 5] for some vision examples) but to our knowledge has not been used for domain adaptation.

3 Domain Adaptation Using Regularized Cross-Domain Transforms

We begin by describing our general domain adaptation model in the linear setting, then, in Section 3.1, show how both the linear and the kernelized version of the particular case of a symmetric transform used in our experiments can be implemented using the metric learning approach of [8].

In the following, we assume that there are two domains $\mathcal{A}$ and $\mathcal{B}$ (e.g., source and target). Given vectors $\boldsymbol{x} \in \mathcal{A}$ and $\boldsymbol{y} \in \mathcal{B}$, we propose to learn a linear transformation W from $\mathcal{B}$ to $\mathcal{A}$ (or equivalently, a transformation W^T to transform from $\mathcal{A}$ to $\mathcal{B}$). If the dimensionality of the vectors $\boldsymbol{x} \in \mathcal{A}$ is d_A and the dimensionality of the vectors $\boldsymbol{y} \in \mathcal{B}$ is d_B, then the size of the matrix W is $d_A \times d_B$. We denote the resulting inner product similarity function between $\boldsymbol{x}$ and the transformed $\boldsymbol{y}$ as

$$\text{sim}_W(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T W \boldsymbol{y}.$$

The goal is to learn the linear transformation given some form of supervision, and then to utilize the learned similarity function in a classification or clustering algorithm. To avoid overfitting, we choose a regularization function for W, which we will denote as $r(W)$ (choices of the regularizer are discussed below). Denote $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_A}]$ as the matrix of n_A training data points (of dimensionality d_A) from $\mathcal{A}$ and $Y = [\boldsymbol{y}_1, ..., \boldsymbol{y}_{n_B}]$ as the matrix of n_B training data points (of dimensionality d_B) from $\mathcal{B}$. We will discuss the exact form of supervision we propose for domain adaptation problems in Section 3.1, but for now assume that it is a function of the learned similarity values $\text{sim}_W(\boldsymbol{x}, \boldsymbol{y})$ (i.e., a function of the matrix $X^T W Y$), so a general optimization problem would seek to minimize the regularizer subject to supervision constraints given by functions c_i:

$$\begin{array}{ll} \min_W & r(W) \\ \text{s.t.} & c_i(X^T W Y) \geq 0, \quad 1 \leq i \leq c. \end{array} \tag{1}$$

Due to the potential of infeasibility, we can introduce slack variables into the above formulation, or write the problem as an unconstrained problem:

$$\min_W \; r(W) + \lambda \sum_i c_i(X^T W Y).$$

In this paper, we focus on a special case of this general transformation learning problem, one that employs a particular regularizer and constraints that are a function of the learned *distances*[2]

$$\mathrm{d}_W(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T W (\boldsymbol{x} - \boldsymbol{y}).$$

The regularizer we consider here is $r(W) = \mathrm{tr}(W) - \log\det(W)$. Note that this regularizer can only be applied when the dimensionalities of the two domains are equal ($d_A = d_B$). This choice of regularizer and constraints has previously been studied as a Mahalanobis metric learning method, and is called *information-theoretic metric learning* (ITML) [14]; we stress, however, that the use of such a regularizer for domain adaptation is novel, as is our method for constructing cross-domain constraints, which we discuss in Section 3.1. We call this approach *symm* for short, since the learned transformation W is always symmetric positive definite.

The fact that W is required to be symmetric may be overly restrictive for some applications. We refer the reader to [15], where we develop an asymmetric version of our domain adaptation model with the regularizer $r(W) = \frac{1}{2}\|W\|_F^2$ and constraints that are functions of learned similarities $\mathrm{sim}_W(\boldsymbol{x}, \boldsymbol{y})$. This method, called *asymm* for short in this paper, can also handle the case when $d_A \neq d_B$.

3.1 Domain Adaptation Using Metric Learning

In this section, we describe our specific algorithm in detail. In using symmetric positive definite matrices, the idea is that the shift can be approximated as an arbitrary linear scaling and rotation of the feature space. We aim to recover this transformation by leveraging labeled data consisting of similarity and dissimilarity constraints between points in the two domains. Since the matrix W corresponding to the metric is symmetric positive semi-definite, we can think of it as mapping samples coming from two different domains into a common invariant space, in order to learn and classify instances more effectively across domains. Note that by factorizing W as $W = G^T G$, we can equivalently view the distance d_W between points $\boldsymbol{x}$ and $\boldsymbol{y}$ as $(G\boldsymbol{x} - G\boldsymbol{y})^T(G\boldsymbol{x} - G\boldsymbol{y})$; that is, the distance is simply the squared Euclidean distance after applying the linear transformation specified by G. The transformation G therefore maps data points from both domains into an invariant space. Because a linear transformation may not be sufficient, we optionally kernelize the distance matrix to learn non-linear transformations.

Generating Cross-Domain Constraints Suppose that we want to recognize a total of n categories (tasks), with training data from category i denoted as

[2] Mathematically, to ensure that such constraints are a function of $X^T W Y$, we let $X = Y$ be the concatenation of data points in both domains. This is possible since the dimensionalities of the domains are identical.

$\mathcal{L}_i$ and consisting of $(\boldsymbol{x}, l)$ pairs of input image observations $\boldsymbol{x}$ and category labels l. There are two cases that we consider. In the first case, we have many labeled examples for each of the n categories in the source domain data, $\mathcal{L}^A = \mathcal{L}_1^A \cup ... \cup \mathcal{L}_n^A$, and a few labeled examples for each category in the target domain data, $\mathcal{L}^B = \mathcal{L}_1^B \cup ... \cup \mathcal{L}_n^B$. In the second case, we have the same training data $\mathcal{L}^A$, but only have labels for a subset of the categories in the target domain, $\mathcal{L}^B = \mathcal{L}_1^B \cup ... \cup \mathcal{L}_m^B$, where $m < n$. Here, our goal is to adapt the classifiers trained on tasks $m+1, ..., n$, which only have source domain labels, to obtain new classifiers that reduce the predictive error on the target domain by accounting for the domain shift. We accomplish this by applying the transformation learned on the first m categories to the features in the source domain training set of categories $m+1, ..., n$, and re-training the classifier.

To generate the similarity and dissimilarity constraints necessary to learn the domain-invariant transformation, we use the following procedure. We sample a random pair consisting of a labeled source domain sample $(\boldsymbol{x}_{\boldsymbol{i}}^A, l_i^A)$ and a labeled target domain sample $(\boldsymbol{x}_{\boldsymbol{j}}^B, l_j^B)$, and create a constraint

$$\begin{aligned} d_W(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B) &\leq u \quad \text{if} \quad l_i = l_j, \\ d_W(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B) &\geq \ell \quad \text{if} \quad l_i \neq l_j. \end{aligned} \tag{2}$$

We call these *class-based* constraints, and we use this procedure to construct a set $\mathcal{S}$ of pairs (i, j) of similarity constraints and $\mathcal{D}$ of dissimilarity constraints. Alternatively, we can generate constraints based not on class labels, but on information of the form "target domain sample $\boldsymbol{x}_i^A$ is similar to source domain sample $\boldsymbol{x}_j^B$". This is particularly useful when the source and target data include images of the same object, as it allows us to best recover the structure of the domain shift, without learning anything about particular categories. We refer to these as *correspondence* constraints.

It is important to generate constraints between samples of different domains, as including same-domain constraints can make it difficult for the algorithm to learn the domain shift. In fact, we show experimentally that creating constraints based on class labels without regard for domain boundaries, in the style of metric learning, does considerably worse than our method.

Learning W using ITML As mentioned above, information-theoretic metric learning (ITML) formulates the problem as follows:

$$\begin{aligned} \min_{W \succeq 0} \quad & \mathrm{tr}(W) - \log \det W \\ \text{s. t.} \quad & d_W(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B) \leq u \quad (i,j) \in \mathcal{S}, \\ & d_W(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B) \geq \ell \quad (i,j) \in \mathcal{D}, \end{aligned} \tag{3}$$

where the regularizer $\mathrm{tr}(W) - \log \det W$ is defined only between positive semi-definite matrices. This regularizer is a special case of the *LogDet divergence*, which has many properties desirable for metric learning such as scale and rotation invariance [8]. Note that one typically adds slack variables, governed by a

tradeoff parameter λ, to the above formulation to ensure that a feasible solution can always be found.

We follow the approach given in [8] to find the optimal W for (3). At each step of the algorithm, a single pair $(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B)$ from $\mathcal{S}$ or $\mathcal{D}$ is chosen, and an update of the form

$$W_{t+1} = W_t + \beta_t W_t (\boldsymbol{x}_i^A - \boldsymbol{x}_j^B)(\boldsymbol{x}_i^A - \boldsymbol{x}_j^B)^T W_t$$

is applied. In the above, β_t is a scalar parameter computed by the algorithm based on the type of constraint and the amount of violation of the constraint. Such updates are repeated until reaching global convergence; typically we choose the most violated constraint at every iteration and stop when all constraints are satisfied up to some tolerance ϵ.

In some cases, the dimensionality of the data is very high, or a linear transformation is not sufficient for the desired metric. In such cases, we can apply *kernelization* to the above algorithm in order to learn high-dimensional metrics and/or non-linear transformations. Let $\bar{X} = [X\ Y]$ be the concatenated matrix of data points from both domains. It is possible to show that the updates for ITML may be written in terms of the kernel matrix by multiplying the updates on the left by $\bar{X}^T$ and on the right by $\bar{X}$, yielding

$$K_{t+1} = K_t + \beta_t K_t (\boldsymbol{e}_i^A - \boldsymbol{e}_j^B)(\boldsymbol{e}_i^A - \boldsymbol{e}_j^B)^T K_t,$$

where $\boldsymbol{e}_i^A$ is the standard basis vector corresponding to the index of $\boldsymbol{x}_i^A$ and $K_t = \bar{X}^T W_t \bar{X}$. $K_0 = \bar{X}^T \bar{X}$ corresponds to some kernel matrix over the concatenated input data when we map data points from both domains to a high-dimensional feature space. Furthermore, the learned kernel function may be computed over arbtirary points, and the method may be scaled for very large data sets; see [8, 14] for details.

4 A Database for Studying Effects of Domain Shift in Object Recognition

As detailed earlier, effects of domain shift have been largely overlooked in previous object recognition studies. Therefore, one of the contributions of this paper is a database[3] that allows researchers to study, evaluate and compare solutions to the domain shift problem by establishing a multiple-domain labeled dataset and benchmark. In addition to the domain shift aspects, this database also proposes a challenging office environment category learning task which reflects the difficulty of real-world indoor robotic object recognition, and may serve as a useful testbed for such tasks. It contains a total of 4652 images originating from the following three domains:

Images from the web: The first domain consists of images from the web downloaded from online merchants (www.amazon.com). This has become a very

[3] Available at `http://www.eecs.berkeley.edu/~mfritz/domainadaptation/`.

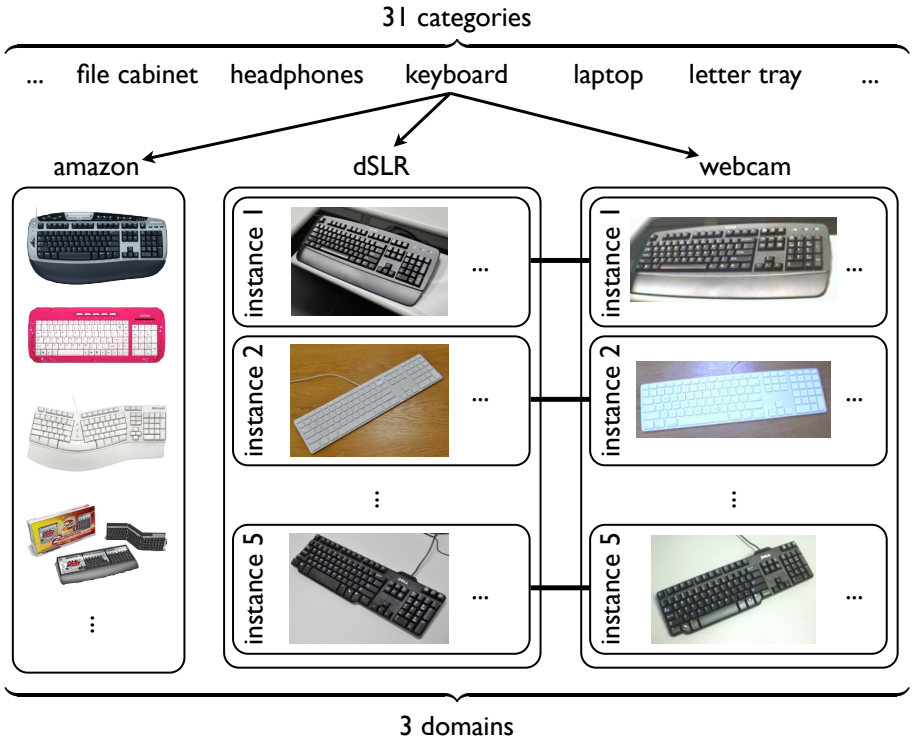


Fig. 4. New dataset for investigating domain shifts in visual category recognition tasks. Images of objects from 31 categories are downloaded from the web as well as captured by a high definition and a low definition camera.

popular way to acquire data, as it allows for easy access to large amounts of data that lends itself to learning category models. These images are of products shot at medium resolution typically taken in an environment with studio lighting conditions. We collected two datasets: *amazon* contains 31 categories[4] with an average of 90 images each. The images capture the large intra-class variation of these categories, but typically show the objects only from a canonical viewpoint. *amazonINS* contains 17 object *instances* (e.g. can of *Taster's Choice* instant coffee) with an average of two images each.

Images from a digital SLR camera: The second domain consists of images that are captured with a digital SLR camera in realistic environments with natural lighting conditions. The images have high resolution (4288x2848) and low noise. We have recorded two datasets: *dslr* has images of the 31 object cat-

[4] The 31 categories in the database are: backpack, bike, bike helmet, bookcase, bottle, calculator, desk chair, desk lamp, computer, file cabinet, headphones, keyboard, laptop, letter tray, mobile phone, monitor, mouse, mug, notebook, pen, phone, printer, projector, puncher, ring binder, ruler, scissors, speaker, stapler, tape, and trash can.

egories, with 5 different objects for each, in an office environment. Each object was captured with on average 3 images taken from different viewpoints, for a total of 423 images. *dslrINS* contains 534 images of the 17 object instances, with an average of 30 images per instance, taken in a home environment.

Images from a webcam: The third domain consists of images of the 31 categories recorded with a simple webcam. The images are of low resolution (640x480) and show significant noise and color as well as white balance artifacts. Many current imagers on robotic platforms share a similarly-sized sensor, and therefore also possess these sensing characteristics. The resulting *webcam* dataset contains the same 5 objects per category as in *dSLR*, for a total of 795 images.

The database represents several interesting visual domain shifts. First of all, it allows us to investigate the adaptation of category models learned on the web to dSLR and webcam images, which can be thought of as in situ observations on a robotic platform in a realistic office or home environment. Second, domain transfer between the high-quality dSLR images to low-resolution webcam images allows for a very controlled investigation of category model adaptation, as the same objects were recorded in both domains. Finally, the amazonINS and dslrINS datasets allow us to evaluate adaptation of product instance models from web data to a user environment, in a setting where images of the same products are available in both domains.

5 Experiments

In this section, we evaluate our domain adaptation approach by applying it to k-nearest neighbor classification of object categories and instances. We use the database described in the previous section to study different types of domain shifts and compare our new approach to several baseline methods. First, we detail our image processing pipeline, and then describe the different experimental settings and elaborate on our empirical findings.

Image Processing: All images were resized to the same width and converted to grayscale. Local scale-invariant interest points were detected using the SURF [1] detector to describe the image. SURF features have been shown to be highly repeatable and robust to noise, displacement, geometric and photometric transformations. The blob response threshold was set to 1000, and the other parameters to default values. A 64-dimensional non-rotationally invariant SURF descriptor was used to describe the patch surrounding each detected interest point. After extracting a set of SURF descriptors for each image, vector quantization into visual words was performed to generate the final feature vector. A codebook of size 800 was constructed by k-means clustering on a randomly chosen subset of the amazon database. All images were converted to histograms over the resulting visual words. No spatial or color information was included in the image representation for these experiments.

In the following, we compare k-NN classifiers that use the proposed cross-domain transformation to the following baselines: 1) k-NN classifiers that operate in the original feature space using a Euclidean distance, and 2) k-NN classifiers

Table 1. Domain adaptation results for categories seen during training in the target domain.

		No shift	Baseline Methods				Our Method	
domain A	domain B	knnAA	knnAB	knnBB	ITML(A+B)	ITML(B)	asymm	symm
webcam	dslr	0.34	0.14	0.20	0.18	0.23	0.25	**0.27**
dslr	webcam	0.31	0.25	0.23	0.23	0.28	0.30	**0.31**
amazon	webcam	0.33	0.03	0.43	0.41	0.43	**0.48**	0.44

Table 2. Domain adaptation results for categories not seen during training in the target domain.

		Baseline Methods		Our Method	
domain A	domain B	knnAB	ITML(A+B)	asymm	symm
webcam	dslr	0.37	0.38	**0.53**	0.49
amazonINS	dslrINS	0.23	0.25	**0.30**	0.25

that use traditional supervised metric learning, implemented using the ITML [8] method, trained using all available labels in both domains. We kernelize the metric using an RBF kernel with width $\sigma = 1.0$, and set $\lambda = 10^2$. As a performance measure, we use accuracy (number of correctly classified test samples divided by the total number of test samples) averaged over 10 randomly selected train/test sets. $k = 1$ was used in all experiments.

Same-category setting: In this setting, each category has (a small number of) labels in the target domain (3 in our experiments) For the source domain, we used 8 labels per category for *webcam*/*dslr* and 20 for *amazon*.

We generate constraints between all cross-domain image pairs in the training set based on their class labels, as described in Section 3.1. Table 1 shows the results. In the first result column, to illustrate the level of performance without the domain shift, we plot the accuracy of the Euclidean k-NN classifier trained on the source domain $\mathcal{A}$ and tested on images from the same domain (knn_AA). The next column shows the same classifier, but trained on $\mathcal{A}$ and tested on $\mathcal{B}$ (knn_AB). Here, the effect of the domain shift is evident, as the performance drops for all domain pairs, dramatically so in the case of the *amazon* to *webcam* shift. We can also train k-NN using the few available $\mathcal{B}$ labels (knn_BB, third column). The fourth and the fifth columns show the metric learning baseline, trained either on all pooled training data from both domains ($ITML(A+B)$), or only on $\mathcal{B}$ labels ($ITML(B)$). Finally, the last two columns show the symmetric variant of our domain adaptation method presented in this paper ($symm$), and its asymmetric variant [15] ($asymm$). knn_BB does not perform as well because of the limited amount of labeled examples we have available in $\mathcal{B}$. Even the more powerful metric-learning based classifier fails to perform as well as the k-NN classifier using our domain-invariant transform.

The shift between dslr and webcam domains represents a moderate amount of change, mostly due to the differences in the cameras, as the same objects were used to collect both datasets. Since webcam actually has more training

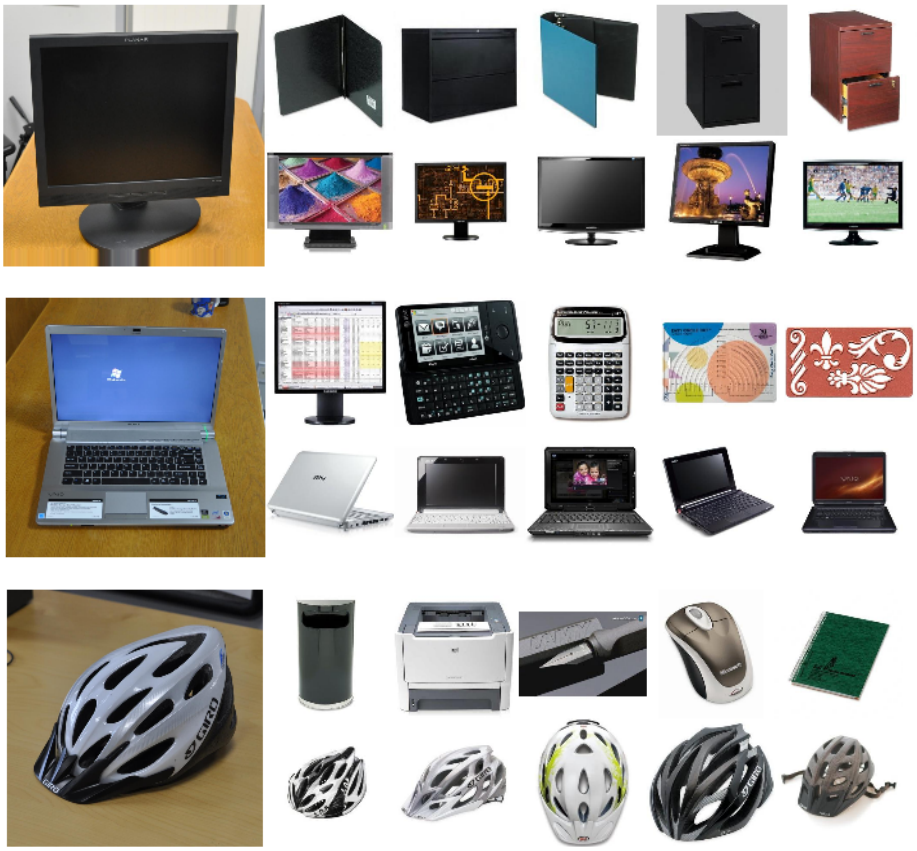


Fig. 5. Examples where our method succeeds in finding images of the correct category despite the domain shift. The large image on the right of each group is a *webcam* query image, while the smaller images are of the 5 nearest neighbors retrieved from the *amazon* dataset, using either the *knn_AB* baseline in Table 1 (top row of smaller images), or the learned cross-domain *symm* kernel (bottom row of smaller images).

images, the reverse webcam-to-dslr shift is probably better suited to adaptation. In both these cases, *symm* outperforms *asym*, possibly due to the more symmetric nature of the shift and/or lack of training data to learn a more general tranformation. The shift between the amazon and the dslr/webcam domains is the most drastic (bottom row of Table 1.) Even for this challenging problem, the adapted k-NN classifier outperforms the non-adapted baselines, with *asymm* doing better than *symm*. Figure 5 show example images retrieved by our method from *amazon* for a query from *webcam*.

New-category setting: In this setting, the test data belong to categories for which we only have labels in the source domain. We use the first half of the categories to learn the transformation, forming correspondence constraints (Section 3.1) between images of the same object instances in roughly the same pose. We test the metric on the remaining categories. The results of adapting *webcam* to *dslr* are shown in the first row of Table 2. Our approach clearly learns something about the domain shift, significantly improving the performance over the baselines, with *asymm* beating *symm*. Note that the overall accuracies are higher as this is a 16-way classification task. The last row shows results on an instance classification task, tackling the shift from Amazon to user environment images. While the symmetric method does not improve on the baseline in this case (possibly due to limited training data, only 2 images per product in *amazon*), the asymmetric method is able to compensate for some of this domain shift.

In both of the above settings, our *symm* method outperforms the standard metric learning baseline $ITML(A+B)$. This clearly demonstrates the advantage of our approach of sampling class-based constraints using inter-domain pairs and, for new-category experiments, of using correspondence constraints.

6 Conclusion

We presented a detailed study of domain shift in the context of object recognition, and introduced a novel adaptation technique that projects the features into a domain-invariant space via a transformation learned from labeled source and target domain examples. Our approach can be applied to adapt a wide range of visual models which operate over similarities or distances between samples, and works both on cases where we need to classify novel test samples from categories seen at training time, and on cases where the test samples come from new categories which were not seen at training time. This is especially useful for object recognition, as large multi-category object databases can be adapted to new domains without requiring labels for all of the possibly huge number of categories. Our results show the effectiveness of our technique for adapting k-NN classifiers to a range of domain shifts.

References

1. H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

2. J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. *ACL*, 2007.
3. O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008.
4. A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.
5. G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *Pattern Recognition and Image Analysis*, 2009.
6. S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. CVPR*, 2005.
7. H. Daume III. Frustratingly easy domain adaptation. In *ACL*, 2007.
8. J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. *ICML*, 2007.
9. L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank. Domain transfer svm for video concept detection. In *CVPR*, 2009.
10. M. Fink. Object classification from a single example utilizing class relevance metrics. In *Proc. NIPS*, 2004.
11. T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *CVPR*, 2004.
12. T. Hertz, A. B. Hillel, and D. Weinshall. Learning a kernel function for classification with small training samples. In *In International Conference on Machine Learning (ICML*, pages 401–408, 2006.
13. W. Jiang, E. Zavesky, S. Chang, and A. Loui. Cross-domain learning methods for high-level visual concept classification. In *ICIP*, 2008.
14. B. Kulis, P. Jain, and K. Grauman. Fast similarity search for learned metrics. *IEEE PAMI*, 39(12):2143–2157, 2009.
15. K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Visual domain adaptation using regularized cross-domain transforms. Technical Report UCB/EECS-2010-106, EECS Department, University of California, Berkeley, July 2010.
16. M. Stark, M. Goesele, and B. Schiele. A shape-based object class model for knowledge transfer. In *ICCV*, 2009.
17. M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.
18. J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. *ACM Multimedia*, 2007.