# Transferring Visual Category Models to New Domains

*Kate Saenko*
*Brian Kulis*
*Mario Fritz*
*Trevor Darrell*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2010-54
http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-54.html

May 7, 2010

# Transferring Visual Category Models to New Domains

Kate Saenko, Brian Kulis, Mario Fritz and Trevor Darrell
UC Berkeley EECS, ICSI

May 7, 2010

**Abstract**

We propose a method to perform *adaptive transfer* of visual category knowledge from labeled datasets acquired in one image domain to other environments. We learn a representation which minimizes the effect of shifting between source and target domains using a novel metric learning approach. The key idea of our approach to domain adaptation is to learn a metric that compensates for the transformation of the object representation that occurred due to the domain shift. In addition to being one of the first studies of domain adaptation for object recognition, this work develops a general adaptation technique that could be applied to non-image data. Another contribution is a new image database for studying the effects of visual domain shift on object recognition. We demonstrate the ability of our adaptation method to improve performance of classifiers on new domains that have very little labeled data.

## 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. [1], [2], [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 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.


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

Figure 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.5 for dataset descriptions). Classification accuracy is averaged over 31 object categories, and over 5 random 80%-20% splits into train/test data.

There are many scenarios where we have a *source* domain with plenty of labeled examples, but want to recognize objects in a *target* domain for which we have very few labels. As Figure 1 shows, it is often insufficient to just train the object classifier on the source domain, as its 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 change the feature distribution and thus violate the assumptions of the classifier. There are many possible causes of visual domain shift, including changes in the camera used to collect the data, the resolution of the images, the lighting, the background, and even the prevalent pose of the objects. 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 (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 [4]. In this paper, we argue that addressing the problem of domain adaptation for ob-

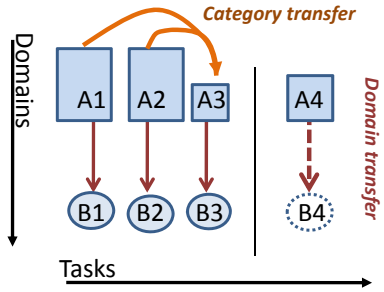



Figure 2: Unlike category transfer methods, our method does not transfer structure between related tasks, but rather transfers the structure of the domain shift to new tasks (which may or may not be related), such as from tasks 1,2 and 3 to task 4, as shown in the figure.

ject recognition is essential for two reasons: 1) while labeled datasets are becoming larger and more available, they still differ significantly from many interesting 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 source datasets to new domains.

In this paper, we propose to compensate for image domain mismatch by using a machine-learning method to automatically adapt existing object classifiers. We introduce a novel adaptation technique based on learning cross-domain metrics. While we evaluate our technique on object recognition, it is a general adaptation method that could be applied to non-image data. Our approach leverages labeled source domain data, together with a small amount of supervised target domain data, to learn a distance metric that essentially reduces differences between the domains. The target domain supervision could consist of either class labels, or similarity/dissimilarity constraints (i.e. this target-domain object is similar to this source-domain object). We focus on two scenarios: one in which significantly more labels are available in the source domain than in the target domain for all categories, and one in which some categories do not have any labels in the target domain. The latter scenario is important as it requires the adaptation method to transfer learned domain knowledge to new categories it encounters in the target domain.

Rather than committing to a specific form of the classifier, we only assume that it operates over (kernelized) distances between examples. This enables our method to benefit a broad range of classification methods, from

k-NN to SVM, as well as clustering methods. The key idea is to learn a distance metric that places examples from different domains that belong to the same category closer together. Metric learning has been successfully applied to a variety of problems in vision and other domains (see [5, 6, 7] for some vision examples) but to our knowledge has not been applied to domain adaptation. In this work, we adapt the information theoretic method of [8], which takes a set of constraints as input and learns a Mahalanobis distance function $d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T A(\boldsymbol{x}_i - \boldsymbol{x}_j)$, parametrized by a positive semi-definite matrix $A$, such that the constraints are satisfied. The algorithm can be kernelized and can be easily applied over large-scale data. Given $m$ tasks with labels in both domains, we generate constraints between pairs of object examples (one from the source and one from the target) that should be considered either similar or dissimilar, and learn a metric that appropriately satisfies such constraints. One of the key advantages of our metric-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.

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. [9]), we do not presume any degree of relatedness between the categories that are used to learn the transferred structure and the categories to which the structure is transferred (see Figure 2). 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.

In the next section, we relate our approach to existing work on domain adaptation. Section 3 provides background knowledge on the information-theoretic metric learning method of [8], and Section 4 presents the domain adaptation algorithm. We evaluate our approach on a new dataset designed to study the problem of visual domain shift, which is described in Section 5, and show empirical results of object classifier adaptation on several simulated and real visual domains in Section 6.

## 2 Related Work

The domain adaptation problem has recently started to gain attention in the natural language community. Daume III [10] proposed a domain adaptation approach 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 [4]. 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. [11] 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. [12] defines a weight for each source training sample based on distance to the target domain, and re-trains the SVM classifier was with re-weighted patterns. The domain transfer SVM (DT-SVM) proposed by Duan et al. [13] 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.

## 3 Information Theoretic Metric Learning

Metric learning has been shown to be successful in a number of learning tasks in vision [5, 6, 7]; in this section we give an overview of information-theoretic metric learning, the method we adapt. For more details, we refer the reader to [8].

The goal of Mahalanobis-based metric learning methods is to learn a distance function parameterized by a positive semi-definite matrix $A$. Given two data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the Mahalanobis distance function is given by

$$d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T A (\boldsymbol{x}_i - \boldsymbol{x}_j).$$

The fact that $A$ is positive semi-definite (i.e., it has non-negative eigenvalues) ensures that the resulting distance is non-negative. Further, by factorizing $A$ as $A = G^T G$, we can equivalently view the Mahalanobis distance as $(G\boldsymbol{x}_i -$

$G\boldsymbol{x}_j)^T(G\boldsymbol{x}_i - G\boldsymbol{x}_j)$; that is, the distance is simply the squared Euclidean distance after applying the linear transformation specified by $G$.

We aim to learn the matrix $A$ given side-information about the desired metric. This information is often either similarity/dissimilarity constraints (pairs of points that should have a small/large distance) or relative distance constraints (two points should have a smaller/larger distance than two other points). In this paper, we utilize similarity and dissimilarity constraints, which we will denote as pairs $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}$ for similarity constraints and $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D}$ for dissimilarity constraints. The optimization problem seeks to find a matrix $A$ such that the learned distances are small for the pairs in $\mathcal{S}$ and large for the pairs in $\mathcal{D}$. Information-theoretic metric learning (ITML) formulates this problem as follows:

$$\begin{aligned} \min_{A \succeq 0} \quad & D_{\ell d}(A, A_0) \\ \text{s. t.} \quad & d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq u \quad (i,j) \in \mathcal{S}, \\ & d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq \ell \quad (i,j) \in \mathcal{D}, \end{aligned} \tag{1}$$

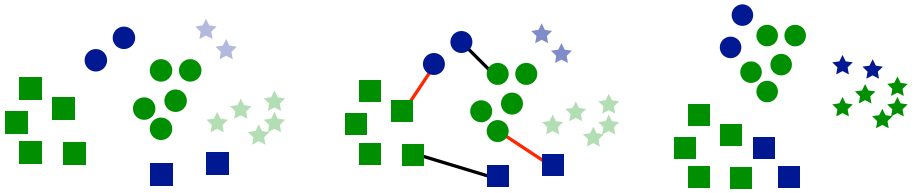
where the regularizer $D_{\ell d}(A, A_0)$ is given by $\operatorname{tr}(AA_0^{-1}) - \log\det(AA_0^{-1}) - d$ ($d$ is the dimensionality of the data) and is defined only between positive semi-definite matrices. This regularizer is called the *LogDet divergence* and has many properties desirable for metric learning such as scale and rotation invariance [8]. $A_0$ is the initial Mahalanobis matrix, and is often chosen to be the identity. Note that one typically adds slack variables, governed by a tradeoff parameter $\gamma$, 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 $A$ for (1). At each step of the algorithm, a single pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ from $\mathcal{S}$ or $\mathcal{D}$ is chosen, and an update of the form

$$A_{t+1} = A_t + \beta_t A_t(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T A_t$$

is applied. In the above, $\beta_t$ 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 $\epsilon$.

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. The natural notion of similarity



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

Figure 3: The key idea of our approach to domain adaptation is to learn a metric that compensates for transformations due to domain shift. By leveraging similarity and dissimilarity constraints (b) we aim to reunite samples coming 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. We are also interested in applying our classifier to new categories (represented by the lightly-shaded stars); our transformations should be effective in learning such new categories. This figure is best viewed in color.

(or kernel) for Mahalanobis metrics is $\boldsymbol{x}_i^T A \boldsymbol{x}_j$, which is just a generalized inner product (the Mahalanobis distance is directly computed using this similarity). Given a matrix of data points $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n]$, the resulting kernel matrix is given by $K = X^T A X$. It is straightforward to show that the updates for ITML may be written in terms of the kernel matrix by multiplying the updates on the left by $X^T$ and on the right by $X$, yielding

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

where $\boldsymbol{e}_i$ is the $i$-th standard basis vector and $K_t = X^T A_t X$. Typically, one selects $A_0 = I$, and so $K_0 = X^T X$, corresponding to some kernel matrix over the input data when we map the input data points in $X$ 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, 7] for details.

# 4 The Metric Learning Approach to Domain Adaptation

In this section we present a domain adaptation algorithm that utilizes the metric learning paradigm described in the previous section to compensate for domain shift.

Our approach is to learn a metric that is invariant to domain-induced alterations of the features. The idea of encapsulating the changes in feature distribution due to domain shift within a Mahalonobis distance seems intuitive, yet, to the best of our knowledge, has not been explored in computer vision or elsewhere. Our main hypothesis 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 existing similarity and dissimilarity constraints between points in the two domains. Since the matrix $A$ corresponding to the Mahalanobis distance 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. Because a linear transformation may not be sufficient, we optionally kernelize the distance matrix to learn non-linear transformations.

We illustrate this idea with a conceptual example. Figure 3(a) shows a three class problem (denoted by the shapes) with corresponding samples from two domains (green and blue). We would like to leverage the more abundant green samples in order to enable or improve classification of the blue samples. Despite the fact that the green and the blue domain form well posed classification problems, they have either been drawn from different parts of the underlying data distribution, or an unknown transformation has caused them to shift. We are to a certain degree agnostic as to the exact transformation process, as it is a very complex composite of the imaging process and environmental conditions. As visualized in Figure 3(b), we form a set of constraints representing similarity constraints (black) as well as dissimilarity constraints (red) *across* the two domains. In (c), the kernel learned by ITML to satisfy these constraints effectively transforms the space to map the green and blue samples of the same class closer together, while keeping the inter-class distances large.

**Generating Cross-Domain Constraints:** Assume that there are $n$ categories, with data from each category denoted as $d_i$, consisting of $(\boldsymbol{x}, y)$ pairs of input data and category labels. 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, $D^s = \{d_1^s, ..., d_n^s\}$, and a few labeled examples for each category in the target domain data, $D^t = \{d_1^t, ..., d_n^t\}$. In the second case, we have the same $D^s$ but only have labels for a subset of the categories in the target domain, $D^t = \{d_1^t, ..., d_m^t\}$, where $m < n$. Here, our goal is to adapt the classifier trained on the tasks $m+1, ..., n$, which only have source domain labels, to obtain a new classifier, which reduces the predictive error on the target domain by accounting for the domain shift. We do this by

applying the transformation learned on the $m$ categories to the features in the source domain training set of the new categories, and re-training the classifier.

To generate similarity constraints $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{S}$ and dissimilarity constraints $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D}$ necessary to learn the domain-invariant transformation, we use the following procedure. We sample a random pair consisting of a labeled source domain sample $(\boldsymbol{x}_{\boldsymbol{i}}^s, y_i^s)$ and a labeled target domain sample $(\boldsymbol{x}_{\boldsymbol{j}}^t, y_j^t)$, and create a constraint

$$\begin{aligned} d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) &\leq u \quad \text{if} \quad y_i = y_j, \\ d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) &\geq \ell \quad \text{if} \quad y_i \neq y_j. \end{aligned} \tag{2}$$

Alternatively, we can generate constraints based not on class labels, but on information of the form: target sample $\boldsymbol{x}_{\boldsymbol{i}}$ is similar to source sample $\boldsymbol{x}_{\boldsymbol{j}}$. 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 a domain-invariant metric.

**A synthetic example:** To demonstrate how the metric-based domain adaptation algorithm works on a simple example, we create a synthetic domain by taking 31 object categories from the webcam dataset described in Sec.5, and adding random noise to a subset of the feature dimensions. We randomly select 20 of the 800 histogram dimensions and add independent Gaussian noise to each selected dimension. Figure 4(a) shows an example of an original feature vector (in this case, a normalized histogram of vector-quantized local SURF [14] features) and its noisy version. This constitutes a moderate amount of noise, and the corresponding "domain shift" significantly degrades the performance of a nearest-neighbor classifier trained on the original data.

Figure 4(b) visualizes the cross-domain metric learned using positive correspondence constraints, generated by enforcing that the distance between each original-domain point and its noisy version should be small (0.001). We plot the diagonal of the $A$ matrix that parametrizes the Mahalonobis distance (the off-diagonal entries were all close to zero). We see that the learned distance correctly gives near-zero weights to the noisy dimensions (marked with red stars) and near-one weights to the rest. In Figure 4(c), we show the weights learned using positive and negative constraints based on class labels. The learned transformation no longer recovers simply the domain

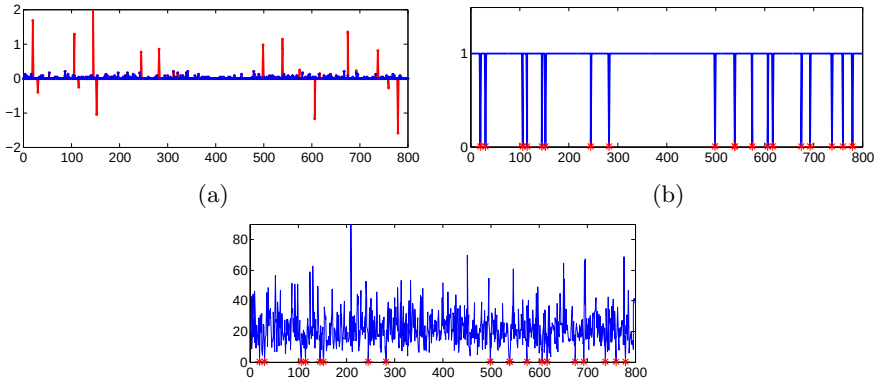



Figure 4: Illustration of how our method learns a metric to compensate for a synthetic domain shift; (a) an original sample (blue) and its noisy-domain version (red); (b) weights learned by the metric using positive correspondence constraints; (c) weights learned using label-based constraints. Noisy dimensions are marked with red stars. This figure is best viewed in color.

shift, but also learns something about the categories, and the off-diagonal entries are no longer close to zero. However, both types of constraints are successful and result in a metric that recovers the original performance when used with the nearest-neighbor classifier.

# 5 A Database for Studying Effects of Domain Shift in Object Recognition

As detailed earlier, effects of domain shift have been largely overlooked in previous recognition studies. Therefore, one of the contributions of this paper is a database that allows researchers to study, evaluate and compare solutions to the domain shift problem by establishing a multiple-domain labeled dataset and benchmark. The database, benchmark code, and code for our method will be made available to the community upon time of publication.

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. Our database provides a total of 4085 images

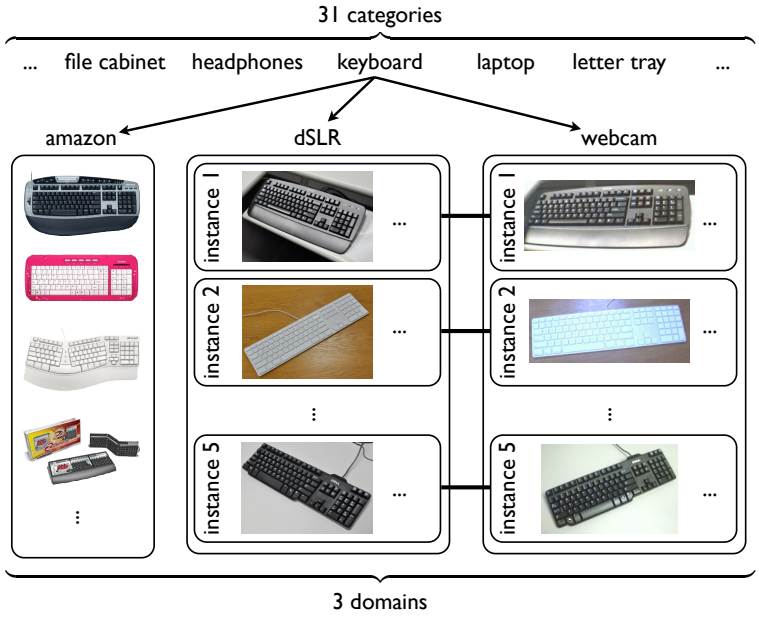



Figure 5: New dataset for investigating domain shifts in visual category recognition tasks. Images of instances form 31 categories are downloaded from the web as well as captured by a high definition and a low definition camera.

of 31 visual categories originating from 3 domains:[1]

**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 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. For each of the 31 categories we collected on average 90 images. The images capture the large intra-class variation of these categories, but typically show the instances only from a canonical viewpoint.

**Images from a digital SLR camera:** The second domain consists of images that are captured with a digital SLR camera in an office environment with natural lighting conditions. The images have high resolution

[1]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.

(4288x2848) and low noise. We have recorded images of 5 instances for each of the 31 categories. For each of the instances there are on average 3 images taken from different viewpoints, for a total of 423 images.

**Images from a webcam:** The third domain consists of images 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. We recorded the same 5 instances as for the dSLR case summing to a total of 795 images.

Each domain poses a challenging categorization task in itself. Beyond that, it facilitates the investigation of how performance degrades when transferring from one domain to another and how effectively different methods can compensate for this domain shift. The database lends itself to two main settings. First, category models learned from the web can be evaluated on the dSLR and webcam images, which can be thought of as in situ observations on a robotic platform in an office environment. In order to properly investigate the effects of domain shift in the absence of the effects of viewpoint variation, we also annotated the database according to canonical views for the dslr and webcam data. Second, domain transfer between the high-quality dSLR image to low resolution webcam images allows for a very controlled investigation of domain shift problems as the same instances were recorded in both domain.

# 6 Experiments

In this section, we evaluate our metric-learning based domain adaptation approach by applying it to k-nearest neighbor classification. k-NN classifies the test sample using the majority vote of the closest training samples. (There is no actual training stage in k-NN, so "training" here refers to the fact that the true label of sample is known.) k-NN is sensitive to domain shift, in particular one that affects distances between samples.

**Implementation:** All images were resized to the same width and converted to grayscale. Local scale-invariant interest points were detected using the SURF [14] detector to describe the image. SURF features have been shown to be highly repeatable and robust to noise, displacement, geometric and photometric transformations. We set the blob response threshold to 1000, and the other parameters to default values. A 64-dimensional non-rotaionally invariant SURF descriptor was used to describe the patch surrounding each detected interest point.

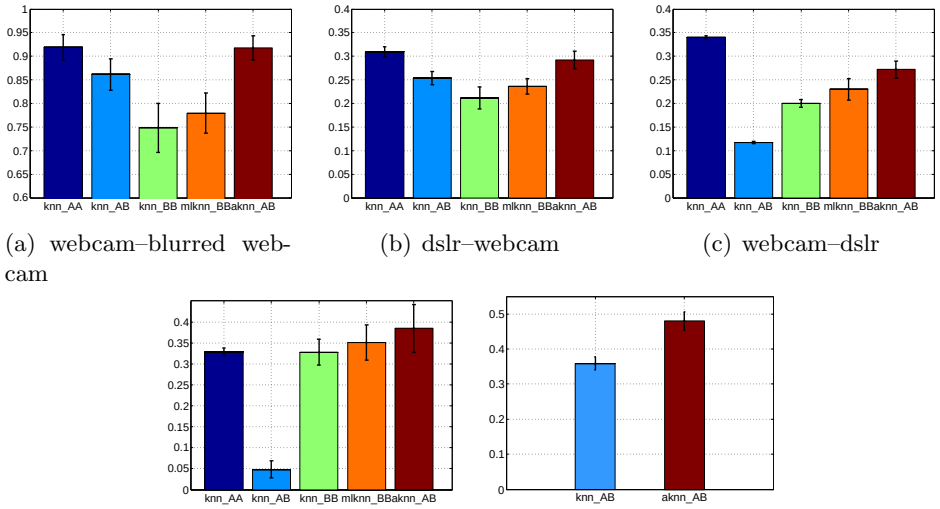
After extracting a set of SURF descriptors for each image, vector quantization into visual words was performed to generate the final feature vector. A codebook of size 800 was constructed by k-means clustering on a randomly chosen subset of the amazon database. All images were converted to histograms over the resulting visual words. No spatial or color information was included in the image representation for these experiments; these would no doubt improve the absolute performance for all methods.

In the following, we compare k-NN classifiers that use the learned cross-domain metric to those that that operate in the original feature space using a Euclidean distance, and show results on several visual domains. We explore two settings for domain adaptation: one in which all categories have (a small number of) labels in the target domain, and one in which the test data belong to categories that only have labels in the source domain. We refer to these as "same-category" and "new-category" settings.

**Same-category setting:** First, we perform a proof-of-concept experiment on a synthetic image domain, and then show results on the Amazon, webcam and dslr domains from our dataset. In all of the following experiments, we generate constraints between all cross-domain image pairs in the training set based on class labels. We kernelize the metric using an RBF kernel with width $\sigma = 1.0$. As a performance measure, we use accuracy (total number of correctly classified test samples divided by the total number of test samples) averaged over 20 randomly selected train/test sets. Here, and in the rest of the section, $A$ refers to the source domain and $B$ refers to the target domain.

To simulate a domain shift where the effective resolution is reduced and some of the information is lost, we create a synthetic domain by applying a Gaussian blur filter to the original webcam images. We then extract SURF features on the blurred images as described above, except that here we sample points on a regularly spaced grid, rather than use an interest point detector, so as to isolate the effects of blurring on the quantized visual words while keeping the point locations the same. We use 20 training images per category in domain $A$ (webcam) and 6 images in domain $B$ (blurred webcam).

Figure 6(a) shows the results. First, as a point of reference, we plot the performance of the k-NN classifier trained on the source domain $A$ and tested on images from the same domain ($knn\_AA$). The next bar shows the accuracy when the training examples come from $A$ and the test examples from $B$ ($knn\_AB$). Here, the reduced effective resolution degrades performance (although not by that much, considering the high level of blurring used) from 91% to 85%. However, after applying the cross-domain metric




(a) webcam–blurred webcam (b) dslr–webcam (c) webcam–dslr

(d) amazon–dslr (e) webcam–dslr

Figure 6: (a)-(d) Domain adaptation results with target labels for all categories; (e) Domain adaptation results for categories not labeled in the target domain

to the data and re-testing the k-NN classifier using the new distances, accuracy goes back up to 91% (*aknn_AB*). This demonstrates that, despite the reduced amount of frequency information in domain $B$ images, they still contain the same amount of object-related information, and that our method compensates for the drop in performance that is caused by the change in the feature distribution.

We also show two baseline methods, both trained using only labeled examples from domain B. The first is a k-NN classifier (*knn_BB*), which we see does not perform as well with the limited amount of labeled examples we have available in B. The second is a k-NN classifier that uses a class-specific metric learned on domain B samples (*mlknn_BB*). Note that our method ultimately relies on a classifier that does not use class-specific learning, and so this baseline is not directly comparable; we show it here as an example of a more powerful classifier, which nevertheless fails to perform as well as our adapted k-NN classifier, given the small amount of labeled target data.

Next, we perform adaptation between real image domains from our dataset. 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. Results of adaptation

Figure 7: Examples of 5 nearest neighbors retrieved for dslr query images (larger images) from the amazon dataset (smaller images), using the Euclidean metric (top row of each set) and the learned cross-domain metric (bottom row of each set)

for the dslr-to-webcam shift, using 8 labels in dslr and 2 labels in webcam, per category, are shown in Figure 6(b). Since webcam actually has more training images, the reverse webcam-to-dslr shift is probably better suited to adaptation. We show results with 20 webcam and 3 dslr labels in Figure 6(c). In both cases, our adapted k-NN classifier ($aknn\_AB$) outperforms the non-adapted Euclidean-distance k-NN classifiers ($knn\_AB$ and $knn\_BB$) and also k-NN with metric learning ($knn\_BB$).

The shift between the amazon and the dslr/webcam domains is the most drastic. Next, we adapt from amazon to the canonical-pose dslr data, training a cross-domain metric by choosing 20 images per category from amazon and 3 images per category from dslr. Results are shown in Figure 6(d), where we see that, even for this challenging problem, the adapted k-NN classifier outperforms the non-adapted baselines.

**New-category setting:** Here we show results for the case when we don't have target labels for all of the categories, and so we must transfer the domain-invariant metric to new categories. We use the first 15 categories in

the webcam and dslr domains to learn the metric, forming correspondence constraints between images of the same object instance in roughly the same pose. We test the metric on the remaining 16 categories, averaging performance over 20 random splits into train/test data. The results are shown in Figure 6(e). We compare the baseline, non-adapted k-NN classifier, which uses the last 16 categories in domain A as training data and domain B as test data ($knn\_AB$) and the adapted k-NN classifier using the metric learned on the first 15 categories ($aknn\_AB$). Our approach clearly learns something about the domain shift, significantly improving the performance. Note that the overall accuracies are higher as this is a 16-way classification task.

# 7 Conclusion

Many successful object recognition methods expect the test image to be drawn from the same distribution as the training dataset, making them brittle in the face of shifting imaging conditions. In this paper, 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. The output of our algorithm is a learned kernel function, which can be computed over arbtirary new points and can scale for very large data sets. Our approach can be applied to adapt a wide range of visual models which operate over 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, from a simulated loss of image information, to the large shifts caused by switching between datasets collected using different paradigms.

# References

[1] Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR. (2007)

[2] Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: ICCV. (2007)

[3] Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2008)

[4] Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. ACL (2007)

[5] Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proc. CVPR. (2005)

[6] Hertz, T., Bar-Hillel, A., Weinshall, D.: Learning distance functions for image retrieval. In: CVPR. (2004)

[7] Kulis, B., Jain, P., Grauman, K.: Fast similarity search for learned metrics. IEEE PAMI **39** (2009) 2143–2157

[8] Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. ICML (2007)

[9] Stark, M., Goesele, M., Schiele, B.: A shape-based object class model for knowledge transfer. In: ICCV. (2009)

[10] Daume III, H.: Frustratingly easy domain adaptation. In: ACL. (2007)

[11] Yang, J., Yan, R., Hauptmann, A.G.: Cross-domain video concept detection using adaptive svms. ACM Multimedia (2007)

[12] Jiang, W., Zavesky, E., Chang, S., Loui, A.: Cross-domain learning methods for high-level visual concept classification. In: ICIP. (2008)

[13] Duan, L., Tsang, I.W., Xu, D., Maybank, S.J.: Domain transfer svm for video concept detection. In: CVPR. (2009)

[14] Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: ECCV. (2006)