Tabula Rasa: Model Transfer for Object Category Detection

Yusuf Aytar Andrew Zisserman
Department of Engineering Science
University of Oxford
{yusuf,az}@robots.ox.ac.uk

Abstract

Our objective is transfer training of a discriminatively trained object category detector, in order to reduce the number of training images required. To this end we propose three transfer learning formulations where a template learnt previously for other categories is used to regularize the training of a new category. All the formulations result in convex optimization problems.

Experiments (on PASCAL VOC) demonstrate significant performance gains by transfer learning from one class to another (e.g. motorbike to bicycle), including one-shot learning, specialization from class to a subordinate class (e.g. from quadruped to horse) and transfer using multiple components. In the case of multiple training samples it is shown that a detection performance approaching that of the state of the art can be achieved with substantially fewer training samples.

1. Introduction

There has been considerable progress recently in object category *detection*: the task of determining if one or more instances of a category are present in an image and, if they are, localizing them [6]. Indeed, for certain types of object categories and images (e.g. compact objects, typical viewpoints), discriminatively trained template part-based detectors perform very well, and source code is freely available for training and development [9]. The negative though is that current methods require training the detector from scratch for each new category – a costly procedure which requires an adequate supply of positive and negative annotated data. For challenges like ImageNet [2] and beyond, where the goal is thousands of categories, this procedure is not scalable for object category detection.

One solution to this problem is to represent object categories indirectly by their attributes, and to learn to detect attributes that can be applied across multiple categories [10, 12, 13, 7, 19, 29]. The benefits are that each attribute can be learnt from multiple classes, so training data is plentiful, and that the attribute representation can be applied to classes that were not used in the training, so it is scalable. However, currently almost all attribute recognition is for image classification (not detection) and even for classification (a simpler task than detection) the performance is inferior to direct discriminative training.

(a) (b) (c)

Figure 1. **The benefit of transfer learning**. The learnt HOG detector template for a motorbike (a) is used as the source for learning a bicycle template together with the samples shown in (b). The resulting learnt bicycle HOG detector template (c) clearly has the shape of a bicycle. Note, here and in the rest of the paper we only visualize the positive components of the HOG vector.

An alternative solution, that we investigate in this paper, is to benefit from category detectors that have previously been learnt for *similar* categories by *transferring* information to a new target category. In particular, our objective is to apply transfer learning to the SVM discriminative training framework for HOG template models of Dalal & Triggs [1] and Felzenszwalb *et al.* [9]. The key intuition is that the learnt template records the spatial layout of positive and negative orientations. Classes that are geometrically similar (e.g. a horse and a donkey) – those that can be 'morphed' into one another by local deformations – will have correspondingly similar HOG templates. To achieve this transfer in the target detector training, the previously learnt template is introduced as a regularizer in the cost function. As illustrated in figure 1, this enables a detector to be learnt for the target category using substantially fewer samples than tabula rasa.

To this end, we introduce and compare three models, including a novel cost function for rigid template transfer and a *geometric* transfer model where the template is deformed using local flow. All three models are defined by convex optimization problems. We also show that the selection of training samples is critical but can be determined

by the source category for best results in the case of one-shot learning.

Related work. Model based transfer learning, originally developed in the machine learning literature as *adaptation* [14, 27], has been applied to computer vision primarily for image classification [17, 22, 23, 27], rather than detection. The work closest to ours is that of Tomassi *et al.* [22, 23] who also use a discriminative SVM framework (though with a quadratic loss), and include the previously learnt model as a regularizer. We extend their work by developing new models and also by considering the more challenging detection problem. We also extend their model by applying local flow algorithms on the classifier template. Others have investigated transfer learning from multiple classes [18, 26] though again for classification. The more general task of reducing training requirements by incorporating prior models has also been considered by Fei-Fei *et al.* [8] for one shot learning, metric-learning by Fink [11], and hierarchical classification by Zweig and Weinshall [30]. Stark *et al.* [21] consider a more geometric based transfer between models, though this is manual at the moment.

There is another school of transfer learning where classifiers are transferred between *domains* [3, 4, 20, 28], for example by learning feature distributions for the source and target domains, but we are not concerned with this type of domain transfer problem here.

In the next section we define the problem, and then introduce two new model transfer methods building on an existing model transfer method of [14, 27]. In the experiments (section 4) we compare transfer learning models for both one-shot learning and multiple samples in the case of transferring from one category to another; and also investigate the case of specialization from a class to a subordinate class.

2. Model Transfer Support Vector Machines

Suppose that we wish to detect an object category of interest, which will be referred as the *target category*, and assume that we have a well trained detector for a visually similar *source* category. Then, the goal is to learn an object detector for the target category by transferring knowledge from the source category detector with the guidance of a few available samples of the target category.

We are concerned here with detection using a sliding window classifier in the manner of [1]. The classifier is linear and is specified by a template vector w, with a scoring function $w^\mathsf{T}x$, where x is the feature vector. Then the task is to learn w for the target category using a few positive training instances x_i, and the source category detector w^s.

In the remainder of the section we introduce and motivate model based transfer learning for support vector machines (SVM). Three variants will be defined: Adaptive Support Vector Machines (A-SVM), a direct application of the classifier transfer of [14, 27]; Projective Model Transfer SVM (PMT-SVM) which relaxes the transfer of the A-SVM model; and Deformable Adaptive SVM (DA-SVM), where the source template w^s is geometrically transformed during the learning.

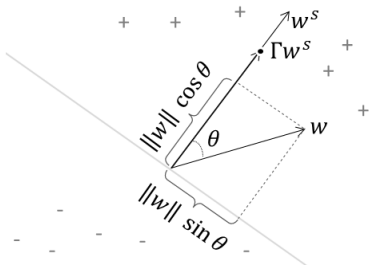


Figure 2. Visualization of the projection of the vector w onto w^s, and onto the separating hyperplane orthogonal to w^s.

2.1. Adaptive SVM (A-SVM)

The idea, originally introduced by [14, 27], is to learn from the source model w^s by regularizing the distance between the learned model w and w^s. As usual, x_i are the training samples, $y_i \in \{-1, 1\}$ the corresponding labels, and $l(\mathbf{x}_i, y_i; w, b) = \max\left(0, 1 - y_i(w^\mathsf{T}x_i + b)\right)$ the hinge loss. The objective function is:

$$L_A \quad = \quad \min_{w,b} \|w - \Gamma w^s\|^2 + C \sum_i^N l(\mathbf{x}_i, y_i; w, b) \quad (1)$$

where Γ controls the amount of transfer regularization, C controls the weight of the loss function, and N the number of samples.

Discussion. We present a brief analysis of A-SVMs to motivate our second model. Intuitively transfer regularization for an A-SVM is like a spring between Γw^s and w, and is equivalent to providing training samples from the source class. The transfer can also be understood by expanding the regularization term. Assume that w^s is l_2 normalized to 1 then

$$\|w - \Gamma w^s\|^2 = \|w\|^2 - 2\Gamma\|w\| cos\theta + \Gamma^2 \quad (2)$$

where $\|w\|^2$ provides the 'normal' SVM margin maximization and $-2\Gamma\|w\| cos\theta$ induces the transfer by maximizing $cos\theta$, i.e. by minimizing the angle θ between the w^s and w as shown in figure 2. Note, that the transfer term $\|w\| cos\theta$ is maximized (and thus the cost minimized) when $\theta = 0$.

However, the term $-2\Gamma\|w\| cos\theta$ also encourages $\|w\|$ to be larger (as this reduces the cost) and this prevents margin maximization. Thus Γ, which should define the amount of transfer regularization, becomes a tradeoff parameter between margin maximization and knowledge transfer.

2.2. Projective Model Transfer SVM (PMT-SVM)

Rather than transfer by maximizing the transfer term $\|w\| cos\theta$, we can instead minimize the projection of w onto the seperating hyperplane orthogonal to w^s (and thereby reduce θ, see again figure 2). In this approach, we can increase the amount of transfer (Γ) without penalizing margin

maximization. The objective function for Projective Model Transfer SVM (PMT-SVM) is:

$$L_{PMT} = \min_{w,b} \|w\|^2 + \Gamma\|Pw\|^2 + C\sum_i^N l(\mathbf{x}_i, y_i; w, b)$$
$$st : w^\mathsf{T} w^s \geq 0 \quad (3)$$

where P is the projection matrix $P = I - \frac{w^s w^{s\mathsf{T}}}{w^{s\mathsf{T}} w^s}$, Γ controls the amount of transfer regularization, and C controls the weight of the hinge loss. $\|Pw\|^2 = \|w\|^2 sin^2\theta$ is the squared norm of the projection of the w onto the source hyperplane. $w^\mathsf{T} w^s \geq 0$ constrains w to the positive halfspace defined by w^s. As $\Gamma \to 0$, (3) becomes a 'classical' SVM objective. Note that the formulation is convex and can be minimized using quadratic optimization.

2.3. Transferring from a deformable source

Transfer regularization can also be performed by using a deformable source template, where small local deformations are allowed for a better fit of the source template to the target. For instance, the wheel part of a motorbike template can be increased in radius and reduced in thickness for a better fit to a bicycle wheel (see figure 1). These small deformations provide more flexible regularization. Local deformations are implemented by the flow of weight vectors from one cell to another, as described in more detail in section 3. The deformation is governed by the following flow definition:

$$\tau(w^s)_i = \sum_j^M f_{ij} w_j^s,$$

where τ denotes the flow transformation, w_j^s is the j^{th} cell in the source template, the flow parameters f_{ij} define the amount of transfer from the j^{th} cell in the source template to the i^{th} cell in the transformed template. Note that one source template cell can contribute to multiple transformed template cells.

To generalize from the rigid A-SVM to deformable transfer formulation, w^s in (1) is replaced with $\tau(w^s)$ which gives the following Deformable Adaptive SVM (DA-SVM) objective:

$$L_{DA} = \min_{f,w,b} \|w - \Gamma\tau(w^s)\|^2 + C\sum_i^N l(\mathbf{x}_i, y_i; w, b) + \lambda\left(\sum_{i\neq j}^{M,M} f_{ij}^2 d_{ij} + \sum_i^M (1 - f_{ii})^2 d\right) \quad (4)$$

where d_{ij} is the spatial distance between the i^{th} cell and j^{th} cell, d is the penalization for the additional flow from the i^{th} source cell to the i^{th} target cell, and λ the weight of the deformation.

Again, the hyper-parameter Γ controls the amount of transfer. The additional parameter λ controls the deformability: high values of λ make the model more rigid so that the solution of (4) approaches that of (1), conversely small values allow a very flexible source template with less regularization ability.

Discussion. The DA-SVM objective (4) defines a convex optimization problem. Even though the term $\|w - \Gamma\tau(w^s)\|^2$ may appear to be non-convex (due to the product of the terms in f_{ij} and w_k), a short calculation shows that the Hessian matrix is positive definite.

2.4. Introducing latent variables

In a similar manner to [9], latent variables can be introduced that specify the position and scale of the detection ROIs relative to the annotation ROIs. As demonstrated in [9] the introduction of latent variables boosts the performance of the detector, because the training is no longer adversely affected by variations in the annotation and can use each training sample more fully. The disadvantage is that the complete optimization problem is no longer convex.

In more detail, the optimization problem becomes $L_{latent} = \min_{z\in Z(X)} L$, where L can be any of the model transfer objectives, $Z(X)$ defines all possible bounding box positions and scales of positive and negative samples, and z selects from these sets. In brief, for positive samples z defines a better alignment and for negative samples it defines the boxes that are more confused with positives and harder to classify as negative. Even though L_{latent} is not convex, it becomes a convex objective function when z is fixed, as all the transfer model objectives are convex.

3. Implementation details

HOG features. The features used are HOG [1] with the extensions proposed by [9] and using their source code. Each HOG cell is represented by a 32 dimensional vector. The feature vector x thus has dimension 80×32. As in [9] we use a 8×10 arrangement of HOG cells. The transfer experiments are based only on the root filter of [9] without parts. Most of the experiments use a single component, but multiple components are used in section 4.1.3.

Flow. The weight vector w corresponds to the grid of $M = 80$ HOG cells tiled in a rectangle, with each cell represented by a $D = 32$ dimensional vector (so w has dimension $M \times D$). Write $\mathbf{h}_j$ for the D dimensional part of w corresponding to the HOG cell j (so that w is a concatenation of M vectors $\mathbf{h}_j$, $j = 1 \ldots D$). The flow is restricted to act on the entire vectors $\mathbf{h}_j$, as $\sum_j f_{ij}\mathbf{h}_j$ i.e. it does not mix components of the cell vectors. This is a simplification that could be removed if necessary.

Training. We are provided with a training set of annotated images with tight bounding boxes around each positive instance (see section 4). During training, we switch between optimizing latent variables (bounding box positions and scales of positive and negative samples) and SVM objectives. The SVMs are trained either via quadratic programmig using the MOSEK [15] optimization toolkit, or via stochastic gradient descent using the VLFeat library [24]. In

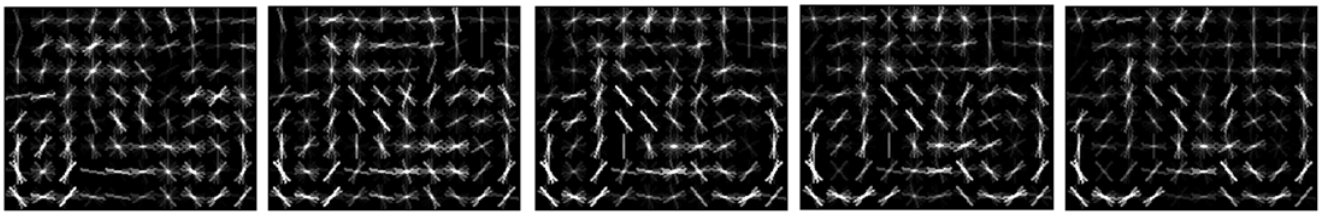


Figure 3. Bicycle classifiers learned using a motorbike classifier as the source and with an increasing number of samples (1,3,6,9,20) from left to right. Note the transition from a template that looks like a motorbike (left) to one that looks like a bicycle (right). A-SVM method is used for learning.

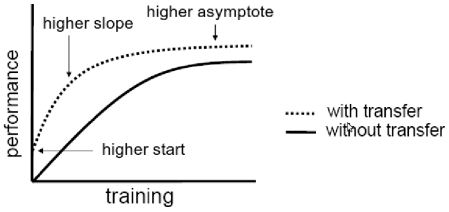


Figure 4. **The benefits of transfer learning.** The three types of performance improvement aspirations from transfer learning. The x-axis is the number of training samples for the target class. (figure from [16, 23]).

other respects (alternating for latent variables etc) the training follows that of [9], and in section 4 we show that we obtain a similar performance to [9].

4. Experiments

Transfer learning can provide three types of performance improvements over learning from the target class alone (see Figure 4) [16, 23]: *(1) higher start*: the initial performance is higher, *(2) higher slope*: performance grows more quickly, *(3) higher asymptote*: the final performance is better. The experiments are designed to see which of these are achieved by the model transfer methods.

There are two types of experiments: (i) *inter-class transfer* where the transfer is from one category to another; and (ii) *specialization* where the transfer is from superior class to subordinate (e.g. from a generic quadruped category to a specific category).

The evaluations are performed on the PASCAL VOC 2007 dataset [5]. The training and validation sets are used to learn the detector, and the performance is reported as average precision (AP) on the test set using the standard PASCAL procedure and evaluation software. For efficiency purposes, we also select a smaller test subset referred as *PASCAL-500* which consists of all the positive samples of the target class and a random selection of other images up to 500. The complete test set is referred as *PASCAL-COMPLETE*.

For most of the experiments we restrict the training to *side views* of the categories horse, sheep, cow, motorbike, and bicycle. These are obtained directly from the VOC data using the pose attributes provided in the annotations. In the training only side facing objects are used and the obtained filter is always facing left (i.e right facing samples are mirrored before training). Table 1 gives the number of side facing positives in the training and test sets for each class. In section 4.1.3 we lift the restriction of side view samples and use all the views for training multiple component models.

Categories	#pos. samples		Felz. [9]	Base. SVM
	train	test		
horse	45	53	40.1%	**44.6%**
sheep	45	43	30.7%	**37.1%**
cow	38	41	**24.9%**	21.5%
bicycle	62	76	50.1%	**59.0%**
motorbike	36	57	**38.1%**	33.8%

Table 1. The comparison of the average precision (AP) results obtained from baseline SVM and Felzenszwalb *et. al.* [9] without parts for the *pascal-side-only* object detection task on *PASCAL-COMPLETE* test set.

Evaluations are performed using two different procedures: (i) *pascal-default*, and (ii) *pascal-side-only*. The *pascal-default* case is the PASCAL VOC [5] evaluation procedure including all views. In *pascal-default* evaluations, while obtaining detections we also use the mirrored version of the side facing detector. In the *pascal-side-only* case, only the left side view ground truth of test samples is used for evaluation and true detections of other poses belonging to the target class are not counted towards AP computation.

The hyperparameters C, Γ and λ are learned on the validation set. C is fixed to 0.002 for all the experiments.

Baseline detectors. The baseline detectors are SVM classifiers trained directly on positive samples without any transfer learning. These classifiers provide the source models in the transfer experiments, and also establish the performance that can be achieved for the target class if all the positive samples of that class are used for training. Other than the learning procedure, the baseline is essentially the same as the discriminatively trained detector of [9] with only the root filter (no parts), and we compare to this method using the code provided by the authors. As shown in table 1 the baseline has very similar performance to [9] (if not better). This establishes the state of the art performance for this dataset.

4.1. Between category transfer

In these experiments we investigate two cases: learning a bicycle classifier by transferring from the motorbike classifier, and learning a horse classifier by transferring from

the cow classifier. We discuss first one shot learning, i.e. learning from a single positive sample of the target class, and then multiple shot learning.

4.1.1 One Shot Learning

The one shot learning scenario investigates the *higher start* aspect of transfer learning benefits (see figure 4). Given a single positive target class sample, we compare four learning methods: learning from the given positive sample only (baseline SVM), rigid transfer using A-SVM and PMT-SVM, deformable transfer using DA-SVM. Evaluations are conducted on the *PASCAL-500* test set using the *pascal-side-only* evaluation procedure. The experiments are performed for each training sample of the target class separately.

We explore the scenario where many samples of the target class are available, and we wish to pick the best one in order to gain the maximum performance from single shot learning. In order to see how these four methods respond to varying the quality of the samples, the training samples are ranked using the source classifier. A few examples from high ranked, medium ranked and low ranked bicycle images are displayed in figure 5.

The results for one shot learning are presented in tables 2 and 3. The APs are averaged over the samples from each quality group, namely: high ranked, medium ranked and low ranked. For all the samples, model transfer methods significantly improve over the baseline SVM which shows that the transfer models enjoy the *high start* benefit. For the high ranked samples, model transfer methods improve approximately 15% over the baseline SVM on both the bicycle and horse detection tasks. The improvement over baseline SVM is around 20% for medium ranked samples in both tasks. On the low ranked samples the baseline SVM is increased from 7% to 21% in horse detection, and 14% to 42% in bicycle detection.

For high ranked samples both PMT-SVM and A-SVM have similar performances. For medium ranked samples PMT-SVM outperforms A-SVM. In general we expect PMT-SVM to outperform A-SVM, as argued in section 2. Experimentally, classical SVMs tend to have a very small $\|w\|$ for one positive sample models. But we observe that A-SVM models trained with one positive sample have relatively larger (almost ten times larger) w vectors. This is due to the transfer term which doesn't let w reduce further after a certain value. This problem is rectified in the PMT-SVM model. Note, that PMT-SVM performs much better using some of the medium ranked samples than using high ranked samples. The explanation is that probably these medium ranked samples are closer to the target hyperplane, and this contributes more to the learning procedure than the high ranked ones. For the low ranked samples, which are either highly occluded, blurred or distorted, A-SVM clearly outperforms PMT-SVM. We conclude that PMT-SVM is highly sensitive to bad (low ranked) samples. DA-SVM performs very similarly to A-SVM for all the samples.

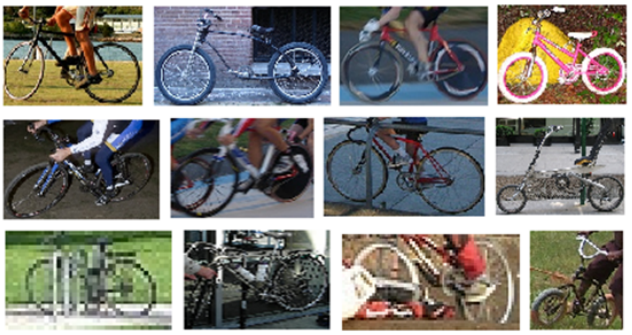


Figure 5. **Examples of ranked *bicycle* samples using the *motorbike* detector.** *Top row:* examples from the high ranked (top 15) samples which are generally not occluded, clean and clearly side facing; *middle row:* the middle ranked (other than top or last 15) samples which are mainly clear, might have small occlusion and viewpoint distortions; and *bottom row:* the low ranked (last 15) samples which are either highly occluded, blurred or not good representatives of side facing class.

Ranks	Base. SVM	A-SVM	DA-SVM	PMT-SVM
01-15	40.5 ± 07.2	**53.9 ± 04.2**	53.7 ± 04.3	53.5 ± 05.7
16-30	33.0 ± 13.5	52.5 ± 08.3	51.9 ± 08.8	**54.7 ± 05.7**
31-45	26.4 ± 13.3	47.1 ± 07.3	47.1 ± 07.6	**48.5 ± 08.7**
46-60	14.0 ± 09.3	42.4 ± 03.7	**42.5 ± 04.2**	27.8 ± 11.3

Source: motorbike(**44.7%**), **Target:** bicycle(**70.1%**), **Test-set:** PASCAL-500, **Test-procedure:** pascal-side-only

In all the tables, test configuration information is given similar to the line above, the values for the source and target are the AP scores of the source(i.e. motorbike) and full target(i.e. bicycle trained with all available samples) detectors on the target task.

Table 2. **Average Precision (AP) comparison of baseline and transfer SVMs on the one shot learning task.** Models are learned using one sample of *bicycle* class and the *motorbike* classifier as the source. The top row displays the average AP results using one of the top 15 (high ranked) samples ranked by the source classifier. Next rows display the next 15 in the ranking. Note the tremendous boost obtained by the transfer method compared to the base SVM (without transfer).

To our knowledge there is no previous work on how to select or weight samples of the target class while performing transfer learning. Under the assumption that the source class is visually similar to the target class, ranking samples with the source detector provides an idea about the quality of available samples. Note, the ranking of samples using the source and full target (i.e. trained with all available samples) classifier has 50% overlap in the top 15 samples, and 66% overlap for the last 15 samples. This source ranking can help during the sample selection or weighting of the samples for transfer learning. Since bad samples clearly deteriorate the performance (see table 2 and 3), low scored samples either should be removed or at least should be assigned small weights.

Ranks	Base. SVM	A-SVM	DA-SVM	PMT-SVM
01-15	15.0 ± 08.0	30.2 ± 04.0	**30.3 ± 03.5**	30.1 ± 05.4
16-30	07.2 ± 04.1	27.0 ± 04.2	27.0 ± 04.3	**27.7 ± 07.1**
31-45	07.2 ± 08.2	**24.1 ± 06.0**	23.8 ± 05.9	11.9 ± 10.3

Source: cow(**26.1%**), **Target:** horse(**60.2%**), **Test-set:** PASCAL-500, **Test-procedure:** pascal-side-only

Table 3. **AP comparison of baseline and transfer SVMs on the one shot learning task.** Models are learned using one sample of *horse* class and the *cow* classifier as the source.

#	Base. SVM	A-SVM	DA-SVM	PMT-SVM
1	05.2 ± 05.6	23.6 ± 02.9	**24.1 ± 03.2**	22.7 ± 12.1
2	09.0 ± 07.7	32.3 ± 03.9	32.4 ± 04.9	**34.3 ± 07.3**
3	18.9 ± 10.9	34.7 ± 05.5	35.0 ± 04.7	**36.0 ± 10.5**
4	24.7 ± 12.2	37.1 ± 04.5	**37.7 ± 04.0**	35.0 ± 05.6
5	28.7 ± 09.5	37.7 ± 06.6	**37.9 ± 06.4**	34.8 ± 06.9

Source: cow(**26.1%**), **Target:** horse(**60.2%**), **Test-set:** PASCAL-500, **Test-procedure:** pascal-side-only

(a)

#	Base. SVM	A-SVM	DA-SVM	PMT-SVM
1	26.9 ± 11.2	51.3 ± 04.5	49.9 ± 05.1	**54.9 ± 04.0**
2	48.4 ± 05.0	**55.5 ± 05.8**	55.2 ± 05.1	55.4 ± 06.0
3	46.9 ± 11.0	54.2 ± 07.1	54.1 ± 06.7	**56.4 ± 06.9**
4	48.2 ± 09.5	**56.0 ± 08.5**	55.4 ± 07.3	54.2 ± 06.0
5	52.5 ± 09.1	58.1 ± 06.5	**58.7 ± 05.6**	56.8 ± 06.4

Source: motorbike(**44.7%**), **Target:** bicycle(**70.1%**), **Test-set:** PASCAL-500, **Test-procedure:** pascal-side-only

(b)

#	Base. SVM	A-SVM	DA-SVM	PMT-SVM
1	26.9 ± 11.2	06.0 ± 09.4	06.2 ± 09.3	**27.8 ± 08.1**
2	48.4 ± 05.0	26.4 ± 05.0	27.8 ± 05.4	**50.0 ± 06.0**
3	46.9 ± 11.0	33.3 ± 12.7	33.5 ± 12.5	**51.4 ± 12.9**
4	48.2 ± 09.5	36.3 ± 14.7	36.0 ± 14.3	**51.1 ± 12.7**
5	52.5 ± 09.1	45.4 ± 13.0	45.5 ± 13.4	**56.0 ± 09.3**

Source: horse(**00.9%**), **Target:** bicycle(**70.1%**), **Test-set:** PASCAL-500, **Test-procedure:** pascal-side-only

(c)

Table 4. **AP results of baseline SVM and model transfer methods.** Transfers are performed **(a)** from cow to horse, **(b)** from motorbike to bicycle, and **(c)** from horse to bicycle (negative transfer). Leftmost column displays the number of positive samples used for learning. In this, and all the following tables and figures, the experiments are performed five times with different randomized orderings of the positive samples.

4.1.2 Multiple shot learning

For the experiments we use a fixed (but random) ordering for four learning methods: target samples only, transfer from the rigid source template using A-SVM and PMT-SVM, and transfer from the deformable source template using DA-SVM. Each experiment is repeated 5 times with a different random order. The APs are averaged for each number of samples. The average removes any idiosyncrasies due to particularly good or bad samples turning up early in the training. These fluctuations in AP can be seen in the standard deviations of the results given in table 4. The transition of the learned template is illustrated in figure 3.

As is clear from table 4, transfer from the source class using A-SVM, DA-SVM and PMT-SVM performs *significantly* better than the baseline SVM, especially for a small number of positive samples. Note that the standard deviations of all three methods are smaller than the baseline SVM, showing that the idiosyncrasies due to good and bad training samples are better tolerated. As the number of positive samples increases, the improvements from transfer learning methods over the baseline decreases.

As can be seen from table 4, PMT-SVM works better for a small number of samples (1-2-3). In table 4(a), one sample PMT-SVM appears worse than A-SVM. In fact, this is caused by one of the orderings where a bad sample is introduced. When we remove that ordering and average the other 4 orderings, for the one sample case, PMT-SVM clearly outperforms A-SVM, with performance 28.12% to 24.64%, respectively. This incident also shows that PMT-SVM is good for one shot learning but it is highly sensitive to sample quality. PMT-SVM also doesn't perform well with large number of samples, probably caused by the introduction of bad samples.

In addition to positive transfer experiments, we also compared the transfer methods in a negative transfer case where we learn a bicycle classifier using a horse classifier as the source. As can be seen from table 4(c) A-SVM and DA-SVM perform worse than the baseline. Since horse is not a suitable class for transfer learning of a bicycle class, the deterioration of A-SVM and DA-SVM is expected. However PMT-SVM still manages to perform some boost over the baseline SVM, which shows that this weaker model can transfer at the coarse (objectness) level as well.

As another type of baseline for the transfer experiments, we include the performance of the source classifier (without transfer) for detecting the target category. This measures the confusion between the two categories. As shown in table 4, if more than 1 positive sample is used then the transfer methods outperform the source classifier for the target category detection.

We also evaluated A-SVM and DA-SVM using a larger scale of positive samples on the PASCAL-COMPLETE test set (see table 5 and figure 6). In all the cases A-SVM and DA-SVM perform better that the baseline SVM. DA-SVM is superior to A-SVM except for the 50 samples case.

In summary, the results show a significant improvement through transfer learning in terms of higher start and higher slope (refer to figure 4).

4.1.3 Multiple shot learning with multiple components (aspects)

These experiments are conducted on classifiers trained with multiple components (aspects) (similar to [9]). We compare two methods: baseline SVM, a classifier with multiple com-

Number of Samples		1	3	5	7	10	15	20	30	50
Test-procedure: pascal-side-only	**Base. SVM**	09.3 ± 08.8	34.2 ± 11.5	41.9 ± 05.9	44.0 ± 09.9	49.9 ± 05.4	55.9 ± 06.8	55.2 ± 03.5	57.9 ± 02.0	58.9 ± 01.3
	A-SVM	28.4 ± 08.1	40.9 ± 06.1	47.3 ± 04.4	48.8 ± 08.4	**52.0 ± 05.9**	56.0 ± 03.8	57.0 ± 03.3	59.0 ± 01.6	**60.2 ± 01.5**
	DA-SVM	**28.7 ± 08.2**	**42.1 ± 05.7**	**48.3 ± 03.6**	**49.1 ± 07.6**	**52.0 ± 05.2**	**57.0 ± 04.7**	**58.0 ± 01.9**	**60.3 ± 02.0**	59.5 ± 00.9
Test-procedure: pascal-default	**Base. SVM**	07.0 ± 04.4	18.6 ± 05.2	22.7 ± 02.1	24.7 ± 04.5	27.1 ± 02.3	29.6 ± 01.9	30.1 ± 01.2	30.7 ± 01.4	31.6 ± 00.9
	A-SVM	14.9 ± 02.5	20.1 ± 02.7	24.0 ± 01.7	25.2 ± 03.1	27.0 ± 02.0	29.9 ± 01.2	31.0 ± 00.9	31.5 ± 01.3	**32.3 ± 00.5**
	DA-SVM	**15.3 ± 02.5**	**20.6 ± 02.4**	**24.5 ± 01.6**	**25.5 ± 03.4**	**27.3 ± 01.7**	**30.2 ± 01.0**	**31.1 ± 00.8**	**31.5 ± 01.3**	32.2 ± 00.7

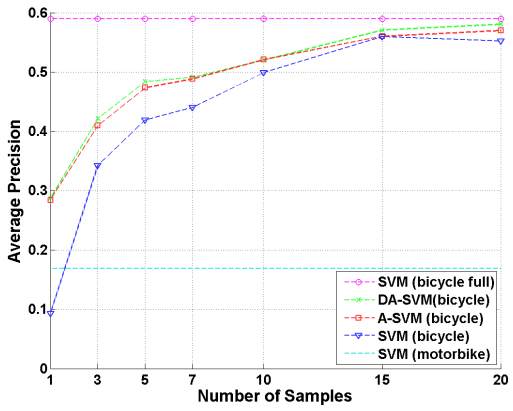
Source: motorbike(pascal-side-only:**16.9%**, pascal-default:**13.2%**), **Target:** bicycle(pascal-side-only:**59.0%**, pascal-default:**32.5%**), **Test-set:** PASCAL-COMPLETE

Table 5. AP results of baseline SVM and model transfer methods for the *bicycle* detection task.

# of Samples	3	5	8	10	13	15	18	20	23	25	50
Base. SVM	10.0 ± 0.8	13.7 ± 4.9	16.6 ± 6.0	16.9 ± 5.7	20.5 ± 4.4	23.4 ± 6.5	**27.8 ± 4.7**	**28.0 ± 4.3**	28.4 ± 3.1	**29.8 ± 5.0**	**36.4 ± 3.4**
A-SVM	**11.9 ± 1.8**	**14.4 ± 4.2**	**17.7 ± 4.0**	**21.9 ± 3.0**	**22.2 ± 3.2**	**25.5 ± 5.0**	26.0 ± 3.3	**28.0 ± 3.7**	**29.7 ± 2.0**	29.4 ± 2.2	33.9 ± 1.9

Source: motorbike(**11.9%**), **Target:** bicycle(**46.6%**), **Test-set:** PASCAL-COMPLETE, **Test-procedure:** pascal-side-only

Table 6. AP results of multiple component baseline SVM and A-SVM for *bicycle* detection task. For A-SVM, the transfer is performed from a multiple component *motorbike* classifier.



Source: motorbike(**16.9%**), **Target:** bicycle(**59.0%**), **Test-set:** PASCAL-COMPLETE, **Test-procedure:** pascal-side-only

Figure 6. AP comparison between baseline SVM and model transfer methods on *bicycle* detection task.

ponents for the root filter learned from the positive samples only; and A-SVM, a multiple component classifier learned by transferring from a multiple component source classifier. The experimental settings are as above using the PASCAL-COMPLETE test set, except now, in training, positive training samples are selected from all poses (not just those of the side views). A model with two distinct components (corresponding to four components once mirrored) is used for the source and the target classifiers. Transfer is performed from the motorbike class to bicycle class.

In the transfer training, each positive training sample is assigned to one of the components of the source classifier, depending on the score of the sample obtained from the source components. Then training is performed in a similar fashion to [9] except we use transfer SVM training instead of classical SVM training. Similarly to the other transfer experiments, transfer methods achieve a good performance, improving over the classical SVM training, particularly for a small number of samples (see table 6). The boost gradually decreases when we increase the number of samples.

4.2. Specialization: superior to subordinate category transfer

These transfer experiments are conducted on the horse, cow and sheep categories of PASCAL VOC 2007. A 'quadruped' category detector is trained from 100 randomly selected examples from the horse, cow and sheep categories. It is then specialized to one of those categories by transfer learning using the model transfer methods. We omitted PMT-SVM since it doesn't perform well for a large number of samples. The experimental settings and procedures are the same as multiple shot learning experiments on PASCAL-COMPLETE test set.

Table 7 shows that specializing from the quadruped class to a subordinate class using model transfer again gives a significant performance improvement over the baseline SVM, especially for a small number of positive samples. Occasionally, using a large number of samples the baseline SVM can perform better than transfer methods. In our experiments the transfer parameter Γ is fixed, decreasing Γ when we have large number of samples would solve the problem, since our models converges to the classical SVM formulation when $\Gamma \to 0$.

Discussion. The benefits of training a superior class (quadruped in this case) are that the training samples can come from multiple subordinate classes. This is similar to the case of attribute training. Indeed the superior class need not involve any training from the target class. However, we are restricted here in using PASCAL VOC – since there are so few categories it is not possible to train a superior class without also including all subordinate classes. Nevertheless, the benefit of specializing by transfer learning is well demonstrated.

5. Conclusions and Future Work

Almost all object category detection methods to date learn the classifier from scratch – tabula rasa. We have proposed a straightforward modification of the learning objective function which retains the benefits of (i) convexity,

Number of Samples		1	3	5	7	10	15	20	30	50
Test-procedure:	**Base. SVM**	03.6 ± 03.8	14.3 ± 07.6	20.0 ± 09.0	25.0 ± 07.3	29.9 ± 04.3	35.9 ± 05.7	40.1 ± 02.8	**45.8 ± 02.6**	**47.1 ± 02.3**
pascal-side-only	**A-SVM**	**21.2 ± 05.5**	**29.7 ± 06.0**	30.9 ± 04.3	**32.6 ± 04.7**	35.3 ± 03.0	**37.8 ± 05.6**	**40.4 ± 03.3**	43.6 ± 03.5	45.4 ± 01.3
	DA-SVM	20.9 ± 05.6	29.2 ± 06.0	**31.5 ± 03.9**	32.1 ± 04.4	**36.6 ± 02.8**	37.2 ± 04.7	40.3 ± 02.9	42.9 ± 03.1	44.0 ± 01.0
Test-procedure:	**Base. SVM**	03.6 ± 03.6	10.3 ± 02.6	10.6 ± 01.8	12.7 ± 02.0	13.8 ± 03.3	14.6 ± 02.4	16.6 ± 01.1	**19.9 ± 00.9**	**21.1 ± 01.5**
pascal-default	**A-SVM**	**11.5 ± 04.0**	**14.5 ± 03.2**	**13.8 ± 03.3**	15.2 ± 03.4	16.0 ± 01.8	16.0 ± 02.8	**17.6 ± 01.0**	**19.9 ± 01.4**	20.6 ± 00.8
	DA-SVM	11.3 ± 04.5	14.2 ± 03.4	13.6 ± 03.0	**15.3 ± 03.0**	**16.2 ± 01.7**	**16.1 ± 02.7**	**17.6 ± 01.0**	19.8 ± 01.9	20.8 ± 00.4

Source: quadruped(pascal-side-only:**15.4%**, pascal-default:**07.9%**), **Target:** horse(pascal-side-only:**44.6%**, pascal-default:**22.3%**), **Test-set:** PASCAL-COMPLETE

Number of Samples		1	3	5	7	10	15	20	30	50
Test-procedure:	**Base. SVM**	03.9 ± 05.0	03.5 ± 03.9	07.1 ± 05.8	08.4 ± 06.1	10.8 ± 07.0	14.0 ± 03.0	13.6 ± 04.7	17.0 ± 03.4	**21.7 ± 00.0**
pascal-side-only	**A-SVM**	12.1 ± 03.1	**12.3 ± 05.9**	**13.2 ± 04.2**	12.7 ± 04.9	**14.7 ± 04.6**	**15.0 ± 04.5**	16.1 ± 03.2	**17.8 ± 02.9**	18.7 ± 00.0
	DA-SVM	**12.6 ± 03.3**	12.1 ± 05.8	13.1 ± 04.0	**12.8 ± 05.2**	14.5 ± 04.5	14.7 ± 03.6	**16.2 ± 03.5**	**17.8 ± 02.1**	18.5 ± 00.0
Test-procedure:	**Base. SVM**	04.9 ± 04.1	07.2 ± 03.1	07.0 ± 03.2	06.8 ± 03.4	08.4 ± 02.7	**10.9 ± 01.2**	**11.6 ± 02.5**	**12.4 ± 01.7**	12.6 ± 00.0
pascal-default	**A-SVM**	**10.6 ± 02.3**	**10.4 ± 03.5**	**09.1 ± 02.5**	09.5 ± 03.1	**10.0 ± 02.2**	10.8 ± 02.1	11.3 ± 00.9	11.7 ± 00.6	11.8 ± 00.0
	DA-SVM	**10.6 ± 02.5**	10.3 ± 03.7	08.7 ± 02.7	**09.6 ± 03.3**	09.7 ± 02.5	10.6 ± 01.8	11.1 ± 01.6	11.5 ± 00.6	**13.3 ± 00.0**

Source: quadruped(pascal-side-only:**08.1%**, pascal-default:**07.1%**), **Target:** cow(pascal-side-only:**21.5%**, pascal-default:**14.9%**), **Test-set:** PASCAL-COMPLETE

Table 7. AP results of baseline SVM and specialization using model transfer methods for the *horse* and *cow* detection tasks.

(ii) optimization methods honed to the special structure of an SVM, and also brings the benefit of learning with fewer training samples. The model transfer methods can act as a 'power-boost' plug-in to any SVM training scheme.

There are a number clear extensions to the model: (i) So far the transfer learning has been applied to the root filter and multiple component scenario of [9], the next step is to extend it to multiple parts. (ii) So far a single feature type has been used, but the model can also be extended to multiple features, such as are used in [25].

Acknowledgements. We are very grateful to Andrea Vedaldi for insightful discussions and the VLFeat library [24]. Financial support was provided by the Royal Academy of Engineering, Microsoft, and ERC grant VisRec no. 228180.

References

[1] N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In *Proc. CVPR*, 2005.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. CVPR*, 2009.

[3] L. Duan, I. Tsang, D. Xu, and S. Maybank. Domain transfer svm for video concept detection. In *Proc. CVPR*, 2009.

[4] L. Duan, D. Xu, I. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. In *Proc. CVPR*, 2010.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[7] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proc. CVPR*, 2009.

[8] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proc. ICCV*, 2003.

[9] P. F. Felzenszwalb, R. B. Grishick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 2010.

[10] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007.

[11] M. Fink. Object classification from a single example utilizing class relevance metrics. In *NIPS*, 2004.

[12] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and simile classifiers for face verification. In *Proc. ICCV*, 2009.

[13] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. CVPR*, 2009.

[14] X. Li. *Regularized Adaptation: Theory, Algorithms and Applications*. PhD thesis, University of Washington, USA, 2007.

[15] Mosek: Optimization Toolkit. http://www.mosek.com, 2010.

[16] E. Olivas, J. Guerrero, M. Sober, J. Benedito, and A. Lopez. *Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques*. 2009.

[17] F. Orabona, C. Castellini, B. Caputo, A. Fiorilla, and G. Sandini. Model adaptation with least-squares svm for adaptive hand prosthetics. In *Proc. ICRA*, 2009.

[18] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *Proc. CVPR*, 2008.

[19] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where – and why? semantic relatedness for knowledge transfer. In *Proc. CVPR*, 2010.

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

[21] M. Stark, M. Goesele, and B. Schiele. A shape-based object class model for knowledge transfer. In *Proc. ICCV*, 2009.

[22] T. Tommasi and B. Caputo. The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories. In *Proc. BMVC.*, 2009.

[23] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Proc. CVPR*, 2010.

[24] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[25] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proc. ICCV*, 2009.

[26] G. Wang, D. Forsyth, and D. Hoiem. Comparative object similarity for improved recognition with few or no examples. In *Proc. CVPR*, 2010.

[27] J. Yang, R. Yan, and A. Hauptmann. Adapting svm classifiers to data with shifted distributions. In *ICDM Workshops 2007*, 2007.

[28] J. Yang, R. Yan, and A. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM Multimedia*, 2007.

[29] X. Yu and Y. Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *Proc. ECCV*, 2010.

[30] A. Zweig and D. Weinshall. Exploiting object hierarchy: Combining models from different category levels. In *Proc. ICCV*, 2007.