Undoing the Damage of Dataset Bias

Aditya Khosla[1], Tinghui Zhou[2], Tomasz Malisiewicz[1],
Alexei A. Efros[2], and Antonio Torralba[1]

[1] Massachusetts Institute of Technology
{khosla,tomasz,torralba}@csail.mit.edu
[2] Carnegie Mellon University
{tinghuiz,efros}@cs.cmu.edu

Abstract. The presence of bias in existing object recognition datasets is now well-known in the computer vision community. While it remains in question whether creating an unbiased dataset is possible given limited resources, in this work we propose a discriminative framework that directly exploits dataset bias during training. In particular, our model learns two sets of weights: (1) bias vectors associated with each individual dataset, and (2) *visual world* weights that are common to all datasets, which are learned by *undoing* the associated bias from each dataset. The visual world weights are expected to be our best possible approximation to the object model trained on an unbiased dataset, and thus tend to have good generalization ability. We demonstrate the effectiveness of our model by applying the learned weights to a novel, unseen dataset, and report superior results for both classification and detection tasks compared to a classical SVM that does not account for the presence of bias. Overall, we find that it is beneficial to explicitly account for bias when combining multiple datasets.

1 Introduction

Recent progress in object recognition has been largely built upon efforts to create large-scale, real-world image datasets [1–3]. Such datasets have been widely adopted by the computer vision community for both training and evaluating recognition systems. An important question recently explored by Torralba and Efros [4], and earlier by Ponce et al [5], is whether these datasets are representative of the visual world, or in other words, unbiased. Unfortunately, experiments in [4, 5] strongly suggest the existence of various types of bias (e.g. selection bias, capture bias, and negative set bias) in popular image datasets.

In the ideal world, more data should lead to better generalization ability but as shown in [4], it is not necessarily the case; performance on the test set of a particular dataset often decreases when the training data is augmented with data from other datasets. This is surprising as in most machine learning problems, a model trained with more examples is expected to better characterize the input space of the given task, and thus yield better performance. The fact that this common belief does not hold in object recognition suggests that the input space of each image dataset is dramatically different, i.e. the datasets are biased.

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part I, LNCS 7572, pp. 158–171, 2012.

Our key observation for undoing the dataset bias is that despite the presence of different biases in different datasets, images in each dataset are sampled from a common *visual world* (shown in Figure 1). In other words, different image datasets are biased samples of a more general dataset—the visual world. We would expect that an object model trained on the visual world would have the best generalization ability, but it is conceivably very difficult, if not impossible, to create such a dataset.

In this paper, we propose a discriminative framework that explicitly defines a bias associated with each dataset and attempts to approximate the weights for the visual world by undoing the bias from each dataset (shown in Figure 1). Specifically, our model is a max-margin framework that takes the originating dataset of each example into account during training. We assume that the bias of all examples from a given dataset can be modeled using the same bias vector, and jointly learn a visual world weight vector together with the bias vector for each dataset by max-margin learning. In order to model both contexual bias and object-specific bias, we apply our algorithm to the tasks of classification and detection, showing promising results in both domains.

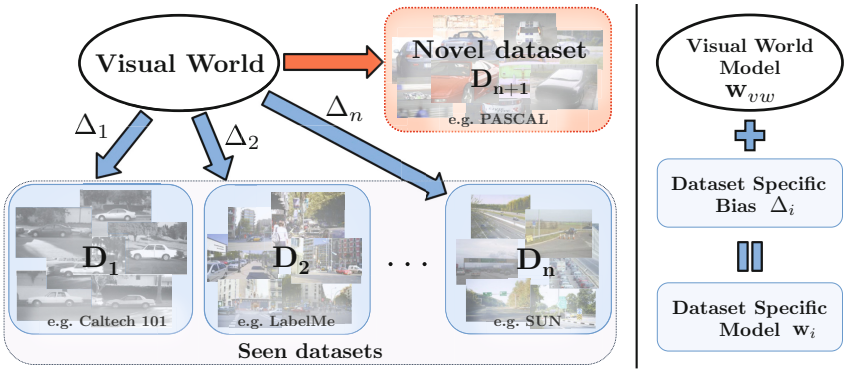


Fig. 1. Left: Sampling from the Visual World. Image datasets are sampled from the same visual world. Each dataset we have collected, such as Caltech101, SUN or LabelMe, has a certain bias that is represented as $\Delta_i, i = 1, \ldots, n$. D_{n+1} represents an independent test set that has not been seen by the model but is sampled from the same visual world. **Right: Overview of our algorithm.** We model the biases as additive linear vectors Δ_i for each dataset. Our goal is to learn a model for the visual world $\mathbf{w}_{vw}$ which has good generalization ability. The dataset specific model $\mathbf{w}_i$ tends to perform well on the corresponding dataset but not generalize well to novel datasets.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 presents the details of our model, including the problem formulation and the optimization algorithm. Experimental results that demonstrate the effectiveness of our model in both classification and detection settings are presented in Section 4 . Section 5 concludes the paper with a summary of our contributions.

2 Related Work

Recently, domain adaptation and transfer learning techniques have been successfully applied to object recognition problems. This line of research addresses the problem of domain shift [6], i.e. mismatch of the joint distribution of inputs between source and target domains. In particular, Saenko *et al.* [7] provide one of the first studies of domain adaptation for object recognition. The key idea of their work is to learn a regularized transformation using information-theoretic metric learning that maps data in the source domain to the target domain. Kulis *et al.* [8] generalize the method in [7] to handle asymmetric transformations in which the feature dimensionality in source and target domain can be different.

However, both of the above methods require labeled data from the target domain as the input consists of paired similar and dissimilar points between the source and the target domain. In contrast, Gopalan *et al.* [9] propose a domain adaptation technique for an unsupervised setting, where data from the target domain is unlabeled. The domain shift in this case is obtained by generating intermediate subspaces between the source and target domain, and projecting both the source and target domain data onto the subspaces for recognition. While most domain adaptation models need to be re-trained for every new domain, Jain and Learned-Miller [10] proposed an online method based on Gaussian process regression that rapidly adapts to the new domain without re-training.

A mathematical framework similar to ours is proposed in [11, 12] for multi-task learning, where solutions to multiple tasks are tied through a common weight vector. The common weight vector is used to share information among tasks but is not contrained to perform well on any task on its own. This is the crucial difference between [11] and our setting: our goal is to learn a common weight vector that can be used independently and is expected to perform well on a new dataset.

We note that our model is different from conventional transfer learning approaches. In terms of problem setting, transfer learning techniques generally fall into three categories [13]: (1) Inductive transfer learning, (2) Transductive transfer learning, and (3) Unsupervised transfer learning. The fundamental difference between our approach and transfer learning approaches is that there is no data available from the target domain during training, and that the target task is the same as the source task.

We evaluate our algorithms on cross-dataset generalization [14–16] in a hold-one dataset out fashion. However, unlike previous works our algorithm explicitly models the dataset bias to mitigate its negative effects. To the best of our knowledge, the problem we address is novel. We hope that our work will provide new insights for the object recognition community on building systems that work in real-world scenarios, and encourage the evaluation of algorithms with respect to better cross-dataset generalization capability.

3 Discriminative Framework for Undoing Dataset Bias

Our aim is to design an algorithm to learn a visual world model, and the bias for each dataset with the following properties: (1) We would expect the visual world model to perform well on average, but not necessarily the best on any particular dataset, since it is not biased towards any one dataset. (2) On the other hand, we would expect the biased model, obtained by combining the visual world model and the learned bias, to perform the best on the dataset that it is biased towards but not necessarily generalize well to other datasets. To this end, we propose a discriminative framework to jointly learn a weight vector corresponding to the visual world object model, $\mathbf{w}_{vw}$, and a set of bias vectors, Δ_i, for each dataset, D_i, that, when combined with the visual world weights result in an object model specific to the dataset. Specifically, we formulate the problem in a max-margin learning (SVM) framework that resembles [11].

3.1 Terminology and Assumptions

In this section, we define the terminology used in the algorithm and some of the assumptions of our model.

Terminology. Assume that we are given n datasets, $D_1, \ldots, D_n$ with a common object class. Each dataset $D_i = \{(\mathbf{x}_1^i, y_1^i), \ldots, (\mathbf{x}_{s_i}^i, y_{s_i}^i)\}$, consists of s_i training examples, $(\mathbf{x}_j^i, y_j^i)$, where $\mathbf{x}_j^i \in \mathbb{R}^m$ represents the m-dimensional feature vector and $y_j^i \in \{-1, 1\}$ represents the label for example j from dataset D_i. In our algorithm, we learn one set of weights, $\Delta_i \in \mathbb{R}^m$, corresponding to the bias of each dataset D_i, and another set of weights, $\mathbf{w}_{vw} \in \mathbb{R}^m$, corresponding to the visual world. The weights are related by the equation, $\mathbf{w}_i = \Omega(\mathbf{w}_{vw}, \Delta_i) = \mathbf{w}_{vw} + \Delta_i$, where $\mathbf{w}_i \in \mathbb{R}^m$ corresponds to the weight vector for dataset D_i.

Assumptions. Our method is general and can be applied to any number of datasets containing a common object class. We assume that the features used are common for all images from all datasets. Further, we assume that the bias between datasets can be identified in feature space (i.e. the features are rich enough to capture the bias in the images). This assumption allows us to model the weights learned for a specific dataset as a function, Ω, of bias and weights for the visual world. This relationship (linear additive) is kept fixed in our paper, but there are other possible ways to model this relationship (e.g. multiplicative, non-linear) that would likely affect the optimization algorithm.

3.2 Algorithm

Our algorithm is largely based on max-margin learning (SVM), and explicitly models the bias vector in feature space for each dataset.

Learning $\mathbf{w}_{vw}$ and Δ_i amounts to solving the following optimization problem:

$$\min_{\mathbf{w}_{vw},\Delta_i,\xi,\rho} \frac{1}{2}||\mathbf{w}_{vw}||^2 + \frac{\lambda}{2}\sum_{i=1}^{n}||\Delta_i||^2 + C_1\sum_{i=1}^{n}\sum_{j=1}^{s_i}\xi_j^i + C_2\sum_{i=1}^{n}\sum_{j=1}^{s_i}\rho_j^i \quad (1)$$

$$\text{subject to} \quad \mathbf{w}_i = \mathbf{w}_{vw} + \Delta_i \quad (2)$$

$$y_j^i \mathbf{w}_{vw} \cdot \mathbf{x}_j^i \geq 1 - \xi_j^i, \quad i = 1 \ldots n, j = 1 \ldots s_i \quad (3)$$

$$y_j^i \mathbf{w}_i \cdot \mathbf{x}_j^i \geq 1 - \rho_j^i, \quad i = 1 \ldots n, j = 1 \ldots s_i \quad (4)$$

$$\xi_j^i \geq 0, \rho_j^i \geq 0, \quad i = 1 \ldots n, j = 1 \ldots s_i \quad (5)$$

where C_1, C_2 and λ are hyperparameters, and ξ and ρ are the slack variables.

We note the changes from the regular SVM setting: (1) the bias vectors, Δ_i regularized to encourage the dataset specific weights to be similar to the visual world weights, (2) additional constraints (described below), and (3) the hyperparameters C_1, C_2 and λ (described below).

Constraints. *Equation 2*: This defines the relationship between $\mathbf{w}_{vw}$, $\mathbf{w}_i$ and Δ_i. We choose a simple relationship to ensure that the objective function is convex. *Equation 3*: The slack variable ξ corresponds to the loss incurred across all datasets when using the visual world weights $\mathbf{w}_{vw}$. The visual world weights are expected to generalize across all datasets, so this loss is minimized across all training images from all datasets. *Equation 4*: The slack variable ρ corresponds to the loss incurred when an example is incorrectly classified by the biased weights. For each dataset, only the corresponding biased weights are required to classify the example correctly, i.e. when training with Caltech101 and SUN, the biased weights for Caltech101 are not penalized if they incorrectly classify an image from the SUN dataset.

Hyperparameters. The hyperparameters C_1 and C_2 are similar to the standard SVM parameter used to balance terms in the learning objective function. C_1 and C_2 allow us to control the relative importance between the two contraints of optimizing loss on the visual world and the individual datasets. λ defines the weight between learning independent weights and a common set of weights for all datasets, i.e. when $\lambda \to \infty$, the biases Δ_i tend towards zero, leading to a common set of weights for all datasets, while $\lambda = 0$ results in the weights for each dataset being independent as there is no restriction on the biases.

3.3 Optimization

In this section, we describe how to optimize Equation 1 described in Section 3.2. We observe that the objective function is convex, thus can be optimized using stochastic subgradient descent. We use the same optimization algorithm for both classification and detection experiments.

We rewrite the objective in an unconstrained form, in terms of $\mathbf{w}_{vw}$ and Δ_i's:

$$\min_{\mathbf{w}_{vw},\Delta_i} \frac{1}{2}||\mathbf{w}_{vw}||^2 + \sum_{i=1}^{n}\left[\frac{\lambda}{2}||\Delta_i||^2 - \mathcal{L}(\mathbf{w}_{vw}, \Delta_i)\right] \quad (6)$$

where $\mathcal{L}(\mathbf{w}_{vw}, \Delta_i) = \sum_{j=1}^{s_i} \Big(C_1 \min(1, y_j^i \mathbf{w}_{vw} \cdot \mathbf{x}_j^i) + C_2 \min(1, y_j^i (\mathbf{w}_{vw} + \Delta_i) \cdot \mathbf{x}_j^i) \Big)$.

Then, we find the subgradients with respect to both $\mathbf{w}$ and Δ_i's:

$$\mathbf{w}'_{vw} = \mathbf{w}_{vw} - \sum_{i=1}^{n} \left[C_1 \sum_{J^i} y_j^i \mathbf{x}_j^i + C_2 \sum_{K^i} y_j^i \mathbf{x}_j^i \right] \tag{7}$$

$$\Delta'_i = \lambda \Delta_i - C_2 \sum_{K^i} y_j^i \mathbf{x}_j^i \tag{8}$$

where $J^i = \{j | y_j^i \mathbf{w}_{vw} \cdot \mathbf{x}_j^i < 1\}$

$K^i = \{j | y_j^i (\mathbf{w}_{vw} + \Delta_i) \cdot \mathbf{x}_j^i < 1\}$

Implementation Details. In our experiments, we set the learning rate, $\alpha = 0.2/i$, where i is the number of iterations. We use a batch size of one example for stochastic subgradient descent with an adaptive cache, similar to [17]. Our classification algorithm takes ∼8 minutes to compute when combining 4 datasets (containing more than 30,000 examples) on a single core. In our experiments, we set the value of C_2 to be some fraction of C_1 to better model a trade-off between loss on visual world and individual datasets.

4 Experiments

To evaluate our framework, we apply our algorithm to two tasks: object classification (identifying whether an object is present in the image) and object detection (localizing an object in the image). We apply our framework to both classification and detection in order to capture different types of biases. In classification, we capture contexual bias as we use global image features that include both the object and its surroundings (i.e. context), while in detection we capture object-specific bias as we only use the information in the provided bounding box annotation. We use four datasets in our experiments, namely PASCAL2007 [3], LabelMe [18], Caltech101 [19], and SUN09 [20]. The experiments are performed on five common object categories: "bird", "car", "chair", "dog" and "person". Our experiments demonstrate that our framework is effective at reducing the effects of bias in both classification and detection tasks.

In our experiments, we use a regular SVM as baseline because it outperforms the common weight vector from [11] (verified experimentally). This is expected as the common weight vector is not constrained to perform any task in [11] as their goal is to improve performance on individual tasks, and the common weight vector is only used to share information across tasks.

4.1 Object Classification

Setup. Our method is flexible to allow the use of many different visual descriptors. In our experiments, we use a bag-of-words representation. First, we densely extract

Table 1. Average precision (AP) of "car" classification on seen datasets. In this case, the train set of the dataset used for testing is available during training (Section 4.1). Pas, Lab, Cal and SUN refer to the four datasets, PASCAL2007, LabelMe, Caltech101 and SUN09 respectively. SVM_{one} refers to a linear SVM that is trained only on the train set of the corresponding test set, while SVM_{all} refers to a linear SVM trained on a combination of all the data from the train set of all datasets. Our visual world model outperforms SVM_{all} indicating improved generalization, and the biased models are comparable to SVM_{one} (0.742 vs 0.743).

(a) Train on all, test on one at a time

Train	Test	$\mathbf{w}_{Pas}$	$\mathbf{w}_{Lab}$	$\mathbf{w}_{Cal}$	$\mathbf{w}_{SUN}$	$\mathbf{w}_{vw}$	SVM_{all}
All	Pas	**0.638**	0.511	0.548	0.495	0.558	0.590
All	Lab	0.690	**0.729**	0.719	0.733	0.729	0.722
All	Cal	0.894	0.928	**0.998**	0.918	0.979	0.936
All	SUN	0.427	0.515	0.530	**0.603**	0.568	0.549
Average		0.662	0.671	0.698	0.687	**0.709**	0.699

(b) Train + test on one

Train	Test	SVM_{one}
Pas	Pas	0.650
Lab	Lab	0.731
Cal	Cal	0.995
SUN	SUN	0.597
Average		0.743

grayscale SIFT descriptors [21] on each image at multiple patch sizes of 8, 12, 16, 24 and 30 with a grid spacing of 4. Using k-means clustering on randomly sampled descriptors from the training set of all datasets, we construct a vocabulary of 256 codewords. Then, we use Locality-constrained Linear Coding (LLC) [22] to assign the descriptors to codewords. A 3-level spatial pyramid [23] with a linear kernel is used for all experiments in this section. The baseline SVM is implemented using Liblinear [24] and the results are evaluated using average precision (AP).

Classification on Seen Datasets. Before we demonstrate the generalization performance of our model on novel datasets, we first show how our model performs on the same datasets it is trained on. Specifically, we use all four datasets for training the model and apply the learned weight vectors to one test set at a time. The results for "car" classification are shown in Table 1. We compare our results against two baseline SVM models, one trained on all datasets (SVM_{all}, Table 1(a)) and another trained on individual datasets (SVM_{one}, Table 1(b)).

The main observations are as follows: (1) The pronounced diagonals in Table 1(a) indicate that each biased model better adapts to its source dataset than other weight vectors (including $\mathbf{w}_{vw}$), and is comparable to training on one dataset at a time (SVM_{one}). (2) The performance of SVM_{one} is significantly better than SVM_{all}, which shows that additional training examples are not always beneficial (also shown in [4]). Together with (1) it implies a clear presence of dataset bias that can significantly impact performance when left untreated. (3) The visual world weights $\mathbf{w}_{vw}$ outperform the baseline SVM_{all} in most cases, demonstrating the improved generalization ability of our model as compared to SVM_{all}, which does not explicitly model dataset bias, and naively concatenates data from all datasets.

Classification on Unseen Datasets. In this experiment, we evaluate the generalization performance of our method by testing on an unseen dataset, i.e. a dataset whose examples are not available during training. During each experiment, we hold out one dataset as the unseen test set, and train the model on the

other three datasets (referred to as *seen sets*). For example, if Caltech101 is the current unseen test set, then the model is trained on PASCAL2007, LabelMe, and SUN09. We also train a linear SVM on the seen sets for baseline comparison. The results are summarized in Figure 2. We observe that when testing on an unseen dataset, the visual world weights, $\mathbf{w}_{vw}$, typically outperform the SVM trained directly on the seen sets. This is because our model treats examples from each dataset as biased samples of the visual world, and in this way learns visual world weights with better generalization ability than the naive SVM. In fact, the naive SVM is a special case of our model with Δ_i's equal to zero, i.e. assuming all datasets are bias-free. Overall, our algorithm outperforms the baseline by 2.8% across all object categories.

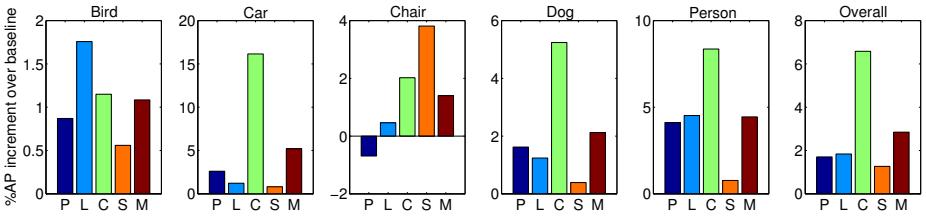


Fig. 2. Classification on unseen datasets. The graphs show *improvement* in percent average precision (%AP) of classification on unseen datasets (Section 4.1) over the baseline (SVM). The labels on the x-axis 'P', 'L', 'C', and 'S' represent the datasets **P**ASCAL2007, **L**abelMe, **C**altech101 and **S**UN09 respectively, while 'M' represents the **M**ean AP increment over all datasets. The five left-most plots correspond to individual object categories while the right-most plot combines the result over all object categories. Overall, our algorithm outperforms the baseline in 24 out of 25 cases, with an overall improvement of 2.8% mAP.

Dataset Classification. In this experiment, we qualitatively and quantitatively evaluate the significance of the learned bias vectors through the task of dataset classification (similar to 'Name That Dataset!' in [4]). We uniformly sample a set of positive images from the test set of different datasets, and predict which dataset each image belongs to using the bias vectors. If the bias vectors are indeed learning the bias as proposed, they would be able to successfully perform this task despite ***not*** being explicitly trained for it.

For "car", the test set consists of $4 \times 90 = 360$ positive images, and similarly, $4 \times 400 = 1600$ for "person" (restricted by the smallest number of positive images in each test set). If the bias vector is learning as proposed, we should expect that images from the i-th dataset would be better classified by Δ_i than by bias vectors of other datasets. To verify this we first train our model on all four datasets, and then apply the learned biases, Δ_i's, to the test set of positive images. The classification performance of Δ_i is measured using average precision. The quantitative results are shown in Table 2, while some qualitative results are shown in Figure 8.

Table 2. Name that dataset! Average precision of dataset classification using the bias vectors (Section 4.1). Each row represents one dataset, while each column represents a particular bias applied to that dataset. We observe that the bias vector corresponding to the particular dataset performs best for this task, suggesting that the bias is being learned as proposed. Note that Caltech101 is the easiest to distinguish from other datasets for both categories (as per our expectation).

	Car				Person			
Datasets	Δ_{Pas}	Δ_{Lab}	Δ_{Cal}	Δ_{SUN}	Δ_{Pas}	Δ_{Lab}	Δ_{Cal}	Δ_{SUN}
PASCAL2007	**0.572**	0.254	0.299	0.314	**0.445**	0.251	0.250	0.382
LabelMe	0.250	**0.373**	0.252	0.315	0.250	**0.536**	0.251	0.314
Caltech101	0.262	0.548	**0.731**	0.250	0.324	0.250	**0.954**	0.250
SUN	0.314	0.251	0.250	**0.593**	0.292	**0.330**	0.251	0.314

The classification results clearly indicate that the bias vectors are indeed learning the specific biases for each dataset. This validates our method of modeling the biases in the chosen way (linear additive in feature space). We emphasize that this result is surprising as the bias vectors were *not* trained to perform this task, and yet, did surprisingly well on it. Furthermore, from Figure 8(a), we can easily identify the contexual bias for cars in each dataset, e.g. SUN09 contains cars on the highways with a prevalent view of the sky, while LabelMe tends to have cars in more urban settings. We can draw similar conclusions from Figure 8(b). It is interesting to note that while many of the top images for person are wrong for LabelMe and SUN09, they share similar visual appearance.

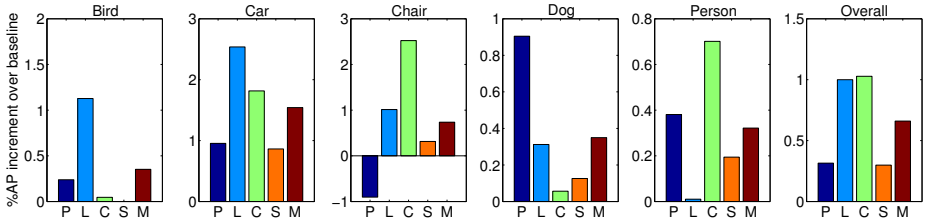
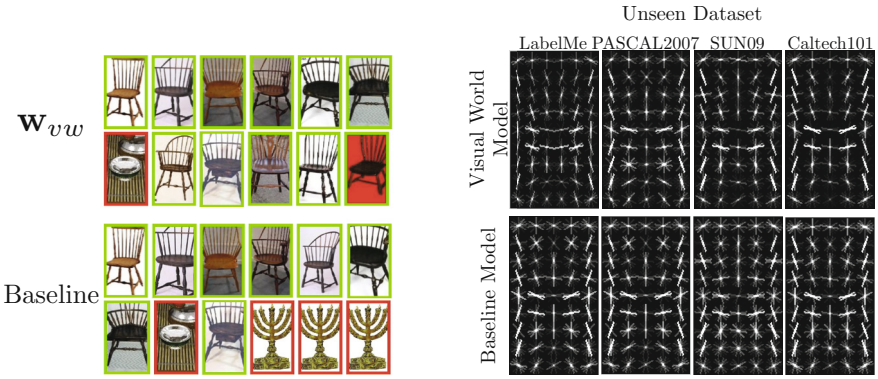


Fig. 3. Detection on unseen datasets. Improvement in percent average precision (%AP) of detection on unseen datasets over the baseline. Refer to the caption of Figure 2 for more details. Note that the graphs indicate that our algorithm outperforms or is comparable to the baseline in most cases, with an overall improvement of 0.7% mAP over the baseline.

4.2 Object Detection

Setup. In this setting, we use our learning algorithm in the deformable parts-based model (DPM) framework by Felzenszwalb *et al* [17]. We learn the DPM without parts and use 2 mixture components to learn the models, for both our algorithm and the baseline (SVM). The mixture models are learned by combining all the images from different datasets and dividing them into 2 components based on aspect ratios. We use the same number of latent positive example mining and hard negative mining updates with the same cache size for all cases.



(a) Detection results (b) HOG templates

Fig. 4. (a) **Comparison of detection results of visual world vs baseline.** Top detection results for "chairs" on Caltech101 comparing visual world model and baseline (SVM). Green/red borders indicate correct/incorrect detections respectively. The scores decrease from left to right and from top to bottom. (b) **Comparison of HOG templates of "chair" for visual world vs baseline.** The visual world weights tend to be more similar to each other, compared to the baseline, despite being trained on different datasets suggesting improved generalization ability of our model. We further observe that there is less 'noise' in the visual world models, likely due to the different biases of the datasets. This figure is best viewed on screen due to the fine differences between the templates.

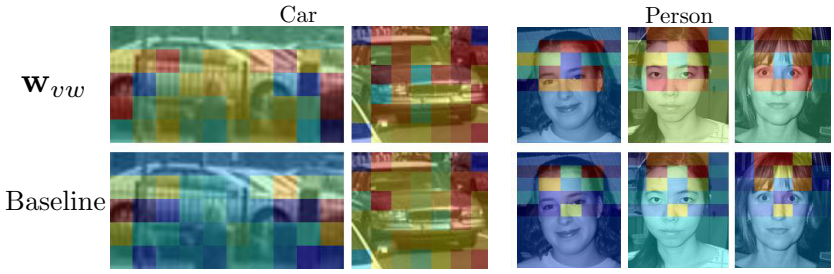


Fig. 5. Spatial distribution of detection weights. Figure showing unique detections and their heatmaps, i.e. detections that are identified by the visual world model but not the baseline. The spatial distribution of weights for "car" and "person" is shown. Red represents the highest score, while blue represents the lowest score. We observe that there are many differences in two sets of heatmaps. The visual world model is better able to generalize by robustly identifying multiple regions corresponding to the object ignored by the baseline method, such as tire/hood for "car", and face/eyes for "person".

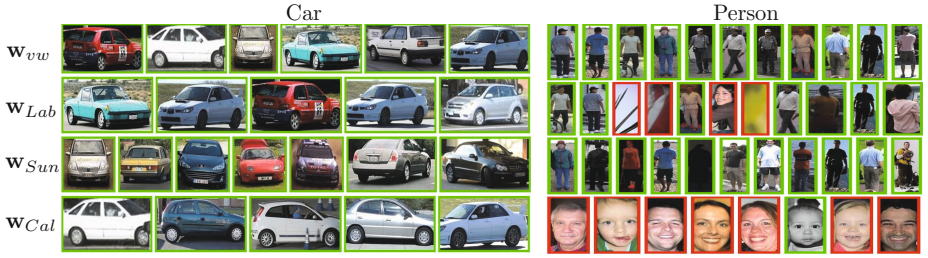
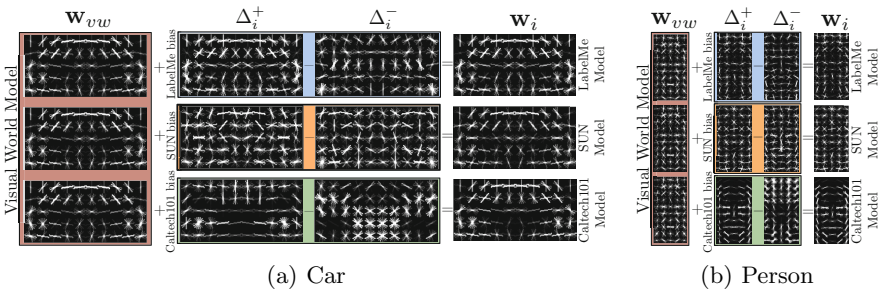


Fig. 6. Top scoring "car" and "person" detections on PASCAL2007. $\mathbf{w}_i$ indicates the dataset specific bias used for the given row. Green/red borders indicate correct/incorrect detections respectively. The scores decrease from left to right. The biases learned for each dataset are quite clear, e.g. for "car", $\mathbf{w}_{Lab}$ tends to prefer cars at an angle of approximately 45°, while $\mathbf{w}_{Sun}$ prefers front/back-facing cars and $\mathbf{w}_{Cal}$ prefers sides of cars. Similarly for "person", we observe that $\mathbf{w}_{Lab}$ prefers full/half body, while $\mathbf{w}_{Sun}$ prefers full body and $\mathbf{w}_{Cal}$ prefers faces. This matches our intuition of the types of examples present in each of these datasets.

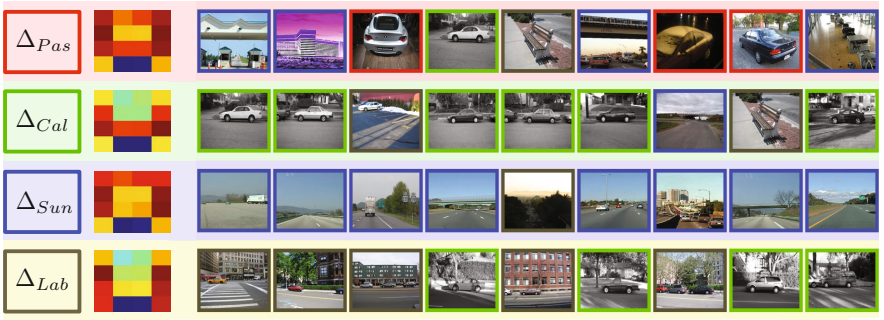


(a) Car (b) Person

Fig. 7. HOG templates for (a)"car" and (b)"person". PASCAL2007 is the unseen dataset for both categories (i.e. the models are trained on the datasets shown in the figure). To visualize both the positive and negative components of the learned bias, we decompose it into two terms: $\Delta_i = \Delta_i^+ - \Delta_i^-$. As shown, the learned bias for each dataset model reflects the bias of the particular dataset, e.g. Caltech101 bias strongly prefers side view of cars with strong positive weights on tires, and frontal faces with focus on facial features, etc.

Detection on Unseen Datasets. We use the same experimental setting as Section 4.1 for detection, where a model is tested on one dataset at a time, while using the other three for training. The results are summarized in Figure 3. Using our framework, which models dataset bias explicitly, we observe performance improvement in the detection task for most cases. Overall, our algorithm outperforms the baseline by 0.7% mAP across all object categories and datasets. We note that this is a difficult task with high variance in object appearance across datasets, and a limited number of training examples available in some of the datasets.

Figure 4(a) shows top "chair" detection of our model and the baseline. We observe that our model is not as easily confused by chair-like objects as the baseline. To explore this further, we visualize the HOG templates for "chair" in



(a) Car

(b) Person

Fig. 8. Dataset classification retrieval results. Top images retrieved by different bias vectors on a pool of positive images sampled from all four datasets. Colored borders indicate the dataset each image is sampled from. For instance, images with a red border are from PASCAL2007, while images with a green border are from Caltech101. Heatmaps in the second column illustrate the importance of each image region for classification (Importance decreases in the order red > green > blue.). The heatmaps are generated using a sum of SVM weights corresponding to different spatial pyramid regions. We observe that the heatmaps confirm our intuition of what is being learned by the model. The heatmaps for cars show that Δ_{Sun} tends to give high importance to the sky region (as seen in retrieved images), while Δ_{Cal} places more importance closer to the center of the image (and we know that cars are centrally located in Caltech101), and similarly for Δ_{Lab}, we observe that the context of street and buildings plays a more important role as cars tend to be small and difficult to localize. We can draw similar intuitions from the person heatmaps.

Figure 4(b). We observe that the models learned by our algorithm tend to be less 'noisy' than the baseline, i.e. as compared to the visual world model, the baseline models depict many gradients that don't correspond to the dominant shape of the object. We speculate that this occurs because the baseline model is forced to fit the biases of all datasets into a single model, while our visual world model is able to identify the common components across the datasets and attribute the remaining edges to various biases. Finally, in Figure 5, we randomly select

some detections found by the visual world weights but not others (including the baseline), and construct a heatmap. The heatmap is based on the detection activation for each HOG cell by the corresponding weights.

Further, we investigate what is learned by the different bias vectors. We visualize the top detection results for "car" and "person" categories in Figure 6 when applying dataset specific models. The biases of the different models are clearly reflected in the detection results. Further, we note that the visual world model has the most 'diverse' detection results compared to dataset specific models. Additionally, we visualize the learned HOG templates in Figure 7. As shown, the HOG templates for the bias terms are quite different for different datasets, implying the effectiveness of our model in capturing the object-specific bias of different datasets. Together with the performance improvement (shown in Figure 3), this implies that our model is effective at modeling and undoing the damage of dataset bias.

5 Conclusion

In this paper, we presented a framework for undoing the damage of dataset bias when combining multiple datasets to train object models, and demonstrated its positive effects in both classification and detection tasks using popular computer vision datasets. Specifically, we introduced a max-margin based model that explicitly defines and exploits the effects of dataset bias. We further demonstrated that the learned bias is indeed indicative of membership to a particular dataset, and hence likely learning both contextual biases and object-specific biases as expected. We would like to emphasize that our framework for learning the visual world model is a first step in building models that explicitly include dataset bias in their mathematical formulation with the goal of mitigating its effect. Further, we hope that this work will encourage the evaluation of algorithms with respect to cross-dataset generalization performance.

Acknowledgements. We thank the anonymous reviewers for their valuable feedback. The paper was co-sponsored by ONR MURIs N000141010933 and N000141010934.

References

1. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for nonparametric object and scene recognition. PAMI 30(11), 1958–1970 (2008)
2. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)
3. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV 88, 303–338 (2010)
4. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR, pp. 1521–1528 (2011)

5. Ponce, J., Berg, T.L., Everingham, M., Forsyth, D., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B.C., Torralba, A., Williams, C.K.I., Zhang, J., Zisserman, A.: Dataset Issues in Object Recognition. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) Toward Category-Level Object Recognition. LNCS, vol. 4170, pp. 29–48. Springer, Heidelberg (2006)
6. Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.: Dataset shift in machine learning. MIT Press (2009)
7. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting Visual Category Models to New Domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010)
8. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR (2011)
9. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: ICCV (2011)
10. Jain, V., Learned-Miller, E.: Online domain adaptation of a pre-trained cascade of classifiers. In: CVPR (2011)
11. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: 10th ACM SIGKDD International Conf. Knowledge Discovery and Data Mining, pp. 109–117 (2004)
12. Ben-David, S., Schuller, R.: Exploiting Task Relatedness for Multiple Task Learning. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 567–580. Springer, Heidelberg (2003)
13. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22 (2010)
14. Bergamo, A., Torresani, L., Fitzgibbon, A.: Picodes: Learning a compact code for novel-category recognition. In: NIPS (2011)
15. Perronnin, F., Sánchez, J., Liu, Y.: Large-scale image categorization with explicit data embedding. In: CVPR, pp. 2297–2304. IEEE (2010)
16. Perronnin, F., Sánchez, J., Mensink, T.: Improving the Fisher Kernel for Large-Scale Image Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
17. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI 32(9), 1627–1645 (2010)
18. Russell, B., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. In: IJCV (2007)
19. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: CVPR Workshop of Generative Model Based Vision (2004)
20. Choi, M.J., Lim, J.J., Torralba, A., Willsky, A.S.: Exploiting hierarchical context on a large database of object categories. In: CVPR, pp. 129–136 (2010)
21. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
22. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR (2010)
23. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
24. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. JMLR 9, 1871–1874 (2008)