# Detecting Objects using Deformation Dictionaries

Bharath Hariharan
UC Berkeley
bharath2@eecs.berkeley.edu

C. Lawrence Zitnick
Microsoft Research
larryz@microsoft.com

Piotr Dollár
Microsoft Research
pdollar@microsoft.com

## Abstract

*Several popular and effective object detectors separately model intra-class variations arising from deformations and appearance changes. This reduces model complexity while enabling the detection of objects across changes in viewpoint, object pose, etc. The Deformable Part Model (DPM) is perhaps the most successful such model to date. A common assumption is that the exponential number of templates enabled by a DPM is critical to its success. In this paper, we show the counter-intuitive result that it is possible to achieve similar accuracy using a small dictionary of deformations. Each component in our model is represented by a single HOG template and a dictionary of flow fields that determine the deformations the template may undergo. While the number of candidate deformations is dramatically fewer than that for a DPM, the deformed templates tend to be plausible and interpretable. In addition, we discover that the set of deformation bases is actually transferable across object categories and that learning shared bases across similar categories can boost accuracy.*

## 1. Introduction

Objects within general categories vary significantly from one instance to the next. We can group the numerous factors contributing to these changes into two broad categories. First, the appearance of an object may change due to lighting variation or albedo differences, arising, for example, from differences in a person's clothing. Second, which is the focus of this work, are the changes that result from deformations due to viewing angle, object pose, or articulated motion such as the movements of a person's arms or legs.

Many detectors are naturally invariant to a limited amount of appearance variation or small deformations but have trouble handling the more extreme changes that real world objects undergo. For instance, Histogram of Oriented Gradients (HOG) features combined with a linear SVM [5] are effective for relatively rigid objects such as pedestrians but have trouble handling more general human poses. Mixture models can address these concerns to some extent [11, 6]. However, as the number of components increases, so does the number of model parameters; simultaneously, the amount of data for training each component decreases. This is a classic recipe for overfitting and limits the gains from further increasing the number of components.

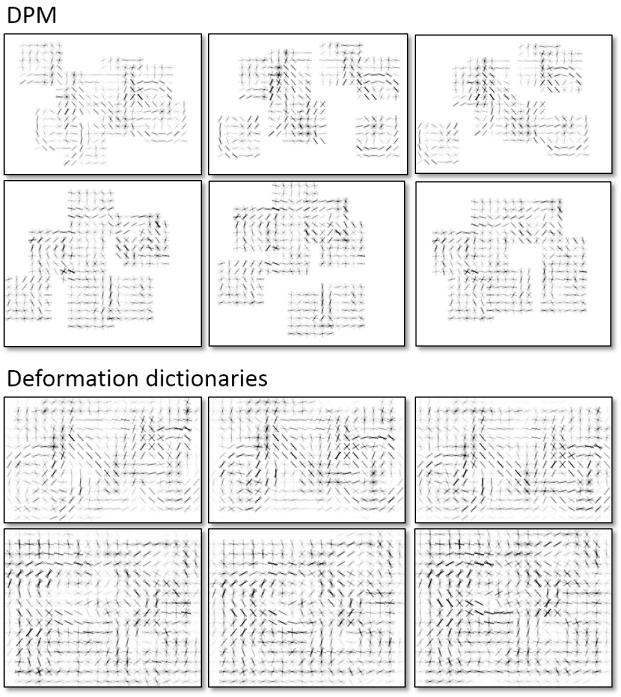


Figure 1: HOG templates generated by sampling the Deformable Parts Model (DPM) [11] and our model. While a DPM is capable of modeling an exponential number of candidate templates, the templates are prone to being implausible and disjoint. In contrast, our deformation dictionary captures only a small but feasible and interpretable set of smooth deformations. The main and surprising result of this work is that a model using only a small dictionary of deformations per component is capable of matching and in some instances outperforming DPMs.

An effective alternative is to train a single canonical appearance template and transform it to account for different views or deformations [4, 11, 15]. This greatly reduces the number of parameters that needs to be learned and leads to

better generalization. A critical aspect of these approaches is the model used for deformations. A common approach to modeling deformations popularized by Felzenszwalb *et al*. [11] are part-based models. Their Deformable Parts Model (DPM) represents an object as a set of parts that translate around an anchor location. The part movements form the set of allowed deformations, while the part appearance templates are shared across the deformations. The model achieves excellent detection performance due in part to its ability to share parameters across deformations.

A common assumption is that the exponential number of deformations enabled by the DPM is also critical to its success. In this paper, we explore whether this assumption is correct and whether alternative deformation models may achieve similar performance. Using a DPM, we may sample the position of the parts to generate HOG templates (Figure 1). Notice that many samples produce implausible configurations of parts. In practice, this may lead to a reduction in precision as unlikely configurations are given high scores. Is it possible to define a deformation model that is more effective at constraining the set of deformations, while not being overly restrictive and reducing recall?

In this paper, we introduce a new deformation model. Our approach consists of a canonical appearance template together with a set or 'dictionary' of likely deformations represented as 2D flow fields. Our goal is to learn a dictionary of flow fields that limits the deformations to those that are highly likely while simultaneously being diverse enough so as not to lead to a reduction in recall. We obtain our deformation dictionary by first computing flow basis vectors extracted from the training data using Principle Component Analysis (PCA). After projecting onto the basis vectors, we obtain a discrete set of deformations by clustering. In order to handle extreme deformations and large changes in appearance or occlusion, we extend our model to include multiple components similar to the DPM. The result is a model with shared appearance parameters that is capable of effectively modeling large deformations.

Interestingly, even though our model only contains a small number of deformations, as opposed to the exponential number of deformations possible in the DPM, we achieve similar and even somewhat improved performance on the PASCAL VOC 2012 dataset [10]. In addition, we demonstrate that the deformation bases are actually transferable across object categories and that learning flow bases across similar categories can boost performance.

## 2. Related work

The notion of objects as a collection of parts predates deformable part models. Constellation models [24] used similar models for image classification. Pictorial structures [12] showed the benefit of such models for pose estimation. The DPM [11] extended this idea to object detection, and introduced two innovations over a single HOG detector: a mixture model to capture different aspects, and a set of high resolution part filters in each mixture component that can move about an anchor location. Conditioned on the root's anchor location, each part in the DPM can move independent of the others, leading to a so-called star model.

Since the introduction of the DPM, several papers have investigated the different design choices that DPM makes. Zhu *et al*. [27] shows that a DPM can be expressed as an exponentially large mixture model where all components share parameters. They find that improvements in performance are due in equal part to the sharing of parameters as well as the ability to construct new unseen mixture components at test time. However, the experiments of [6] suggest that one can come quite close to DPM performance using small mixture models if the components clump together visually similar examples. In this paper, we investigate a middle ground: at test time our model amounts to a small mixture model, but at train time we share parameters among the mixture components. We show that this model can perform as well as or better than the DPM.

Previous papers have looked at different kinds of deformations. The "parts" in the DPM are initialized in a heuristic and unsupervised manner, and the deformations are relative to the root ("star" model). Azizpour and Laptev [2] show improvements in performance by providing part level supervision and replacing the star model with a tree model imputed from the training examples. Vedaldi and Zisserman [23] discard the notion of parts entirely and instead have a small number of predefined global rotation and scaling transformations. Ladicky *et al*. [15] propose a deformation field where each HOG cell in the template moves separately. To prevent the template from "falling apart," they force the deformation field to be locally affine. Our work differs from these models in that we impose a minimal set of *a priori* restrictions on the set of deformations.

Following the tradition of modeling deformations as moving parts, Hejrati and Ramanan [14] introduce part level mixtures and use part level supervision to improve object detection; this follows prior work on pose estimation [26]. Their aim is to improve both detection and pose estimation. Some recent work on pose estimation tries to incorporate 3D constraints directly into the DPM training [20, 19]. Parts are modeled as having 3D locations, resulting in deformations that are 3D, and the mixture components capture different poses or aspects. While these models capture the physical basis of deformations, they require part level supervision. In contrast, we are interested in the case when such supervision is not available.

Other ways of sharing parameters have also been investigated. Bilinear models can be used to factorize part filters into basis filters and weights that can be learned together [22, 21]. Such parameter sharing provides small improve-

ments. Information can also be transferred from one category to another [1, 16]. Aytar and Zisserman [1] use a model trained on one category to regularize training in a different category, while [16] transfers actual transformed training examples from one category to another. Endres *et al.* [9] shares "body plans" between object categories, where a body plan specifies an arrangement of parts.

Outside of object detection, the notion of deformations has surfaced time and again in the vision community. Many of the ideas in this paper echo active appearance models [4], where face landmarks are allowed to deform, and the appearance after deformation is modeled separately. Winn and Jojic [25] propose a generative model for recognition that uses a deformation field to model intraclass variation. However their approach requires rich MRF priors and is limited to simple uncluttered scenes, while we tackle the full-blown object detection problem.

Since it is hard to match pixels across disparate images, [17] propose to use discriminative SIFT features to do the mapping, an idea this work builds upon. A lot of work in image classification and/or matching, for instance [8, 3], warps and aligns images for the purpose of computing similarities. Liu et al. [18] use HOG features to align and cluster images and Drayer and Brox [7] try to match object instances using HOG features with the aim of aligning them. However, such alignment is difficult in the cluttered and occluded scenes that are common in object detection settings.

## 3. Model Overview

Let us begin by considering a typical object detector based on HOG-templates [5]. Given a weight vector $w$, we can compute the score of an image window by computing its HOG features $x$ and doing a dot product with $w$. However, the template and feature vector have a spatial structure: they are divided into a grid of cells. We can make this explicit by writing the features in cell $(i, j)$ as $x_{ij}$ and the corresponding weights as $w_{ij}$. Using this notation, the matching score is given by:

$$f_w(x) = \sum_{i,j} w_{ij}^T x_{ij} \tag{1}$$

In our approach, we want to construct a detector in which the HOG templates can deform. We accomplish this by allowing each HOG cell in the template to move as a unit. A deformation can then be written as a flow field defined over the cells in the template. A cell $(i, j)$ moves to the location $(i + u_{ij}, j + v_{ij})$ where $(u_{ij}, v_{ij})$ is the flow at $(i, j)$. The score of the deformed template can then be written as:

$$f_w(x) = \sum_{i,j} w_{i+u_{ij},j+v_{ij}}^T x_{ij} \tag{2}$$

The above assumes a discrete flow, where each $u_{ij}$ and $v_{ij}$ is integral. However, this is not necessary. We can allow a continuous flow with real values for $u_{ij}$ and $v_{ij}$ using bilinear or bicubic interpolation to deform $w$. In general, given a continuous flow field $(u, v)$, we can represent the score given by a deformed template as:

$$f_w(x) = \sum_{i,j} \Big( \sum_{k,l} \alpha_{kl}^{ij} w_{kl}^T \Big) x_{ij} \tag{3}$$

The coefficients $\alpha$ are the mixing weights over $w$ and are determined by the flow field $(u, v)$. In practice they are quite sparse. Note that the above equation is still linear in both $w$ and $x$; this allows us to write $f_w(x)$ as:

$$f_w(x) = w^T D x \tag{4}$$

The (sparse) matrix $D$ is determined by the flow field $(u, v)$. The description of how we find our candidate flow fields is deferred until Section 4.

Our model considers the deformation as a latent variable and maximizes the score over possible deformations:

$$f_w(x) = \max_{D \in \mathcal{D}} \; w^T D x + w_d^T \psi(D) \tag{5}$$

Here $\mathcal{D}$ is our deformation dictionary containing a set of candidate deformations and the term $w_d^T \psi(D)$ allows us to score each deformation $D \in \mathcal{D}$. This is useful, as some deformations are more likely than others. In practice we use an indicator function for $\psi(D)$; thus $w_d^T \psi(D)$ amounts to assigning each $D$ a bias.

Note that the linear nature of Equation 5 means that we can interpret it either as a fixed template $w$ acting on a deformed feature vector $Dx$, or a deformed template $D^T w$ acting on $x$. The latter interpretation implies that at test time this model can be seen as choosing from a set of templates $D^T w$, one for each $D \in \mathcal{D}$.

### 3.1. Augmented Model

Taking inspiration from the DPM [11], we augment our model with a few additions. We add a "coarse" root template $w_r$ that does *not* deform, in addition to the "fine" template $w_f$ which does:

$$f_w(x) = w_r^T x + \max_{D \in \mathcal{D}} \; w_f^T D x + w_d^T \psi(D) \tag{6}$$

Moreover, while our deformation dictionary allows us to capture a range of deformations, it cannot handle very large or extreme deformations, e.g., changing viewpoint from the front to the side view of a car. To capture such large deformations we utilize a mixture model over aspect ratios similar to [11]. Specifically, we use a separate model $f_w^c(x)$ of the form given in Eqn. 6 for each mixture component $c$. Our final model takes on the form:

$$f_w(x) = \max_c f_w^c(x) \tag{7}$$

To summarize, our model is composed of a mixture of components $c$ where each component is equipped with:

1. $w_r^c$: a single "coarse" root template capturing the general appearance of that aspect.
2. $w_f^c$: a single "fine" template that is at twice the resolution of the coarse template.
3. $D^c \in \mathcal{D}^c$: A set of deformations that can be applied to the fine template $w_f^c$ to produce deformed templates.
4. $w_d^c$: a vector of weights for scoring the deformations.

We emphasize that there is a separate set of weights and deformations for each component (e.g., the front and side views of a car will have their own set of deformations).

### 3.2. Training

Equations 6 and 7 are simply a specific form of the general class of latent variable models [11]:

$$f_w(x) = \max_z w^T \phi(x, z) \tag{8}$$

Therefore, given a deformation dictionary $\mathcal{D}$ for each component, the model can be trained using a latent SVM [11]. For the sake of completeness, we briefly describe the training algorithm here. The training objective takes the form:

$$\min_w \frac{1}{2}\|w\|^2 + C\sum_i (1 - y_i f_w(x_i)) \tag{9}$$

Separating the loss into the loss over positive and negative examples and expanding $f_w(x)$ using (8), we get:

$$\min_w \frac{1}{2}\|w\|^2 + C\sum_{i\in -}(1 + \max_z w^T\phi(x_i, z)) + C\sum_{i\in +}(1 - \max_z w^T\phi(x_i, z)) \tag{10}$$

All terms except the last are convex and the last term becomes linear (and hence convex) once we fix the latent variables for the positives. Hence, following [11], we train the model by iterating between finding the latent variables for the positives, and optimizing over the resulting convex objective. The optimization of the convex objective is itself an iterative process and requires running over negative training images and collecting hard negatives.

### 3.3. Relationship to Other Models

We described our model with the assumption that a deformation dictionary of candidate flows is given. In the next section, we will describe how we compute our dictionary. However, we note that it is possible to express deformable part models and even other proposed models such as [15] as a special case of Eqn. 4. This is because each placement of the parts in a DPM corresponds to a particular flow field, and so the DPM corresponds to allowing $\mathcal{D}$ to be the (exponentially large) set of all possible part placements. We can thus compare our model to these other deformable models:

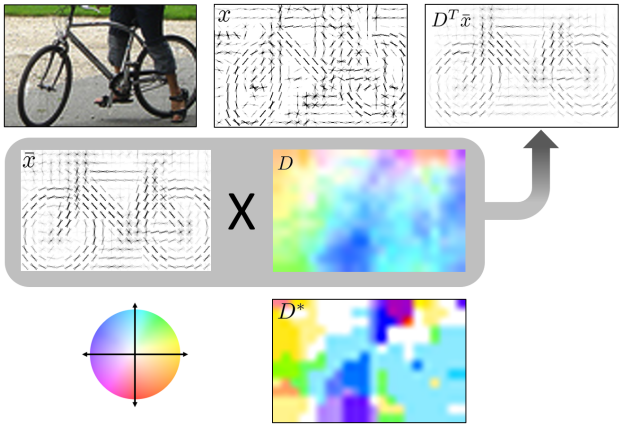


Figure 2: *Top*: from left to right, an exemplar bicycle, its WHO (whitened HOG [13]) features $x$, and the template after applying the projected deformation to the average bicycle template $D^T\bar{x}$. *Middle*: the average bicycle template $\bar{x}$, and the projected flow $D$ on to the top $k = 5$ PCA bases. *Bottom*: color coding for the flow fields shown in this paper, and the raw flow $D^*$ estimated using block matching.

**Size of $\mathcal{D}$:** Our model uses a small deformation dictionary, whereas part based models and models like [15] allow an exponentially large number of deformations. For instance, a DPM allows 81 possible placements for each part, and since the 9 parts are independent, one gets $81^9$ possible deformations. A larger $\mathcal{D}$ allows one to capture a larger number of poses, even ones not seen at train time [27]. However it might also introduce implausible deformations that increase the number of false positives (see Figure 1).

**Form of $\mathcal{D}$:** The inability to synthesize an exponential number of templates may not matter as long as we capture the common deformations in the data. On the other hand, a DPM places parts using simple heuristics, ignoring the structure of the object category. A small set of deformations that are estimated in a data-driven manner might in fact end up covering the space better than an exponentially large set of heuristically determined deformations.

**Form of $\psi(D)$:** An exponentially large $\mathcal{D}$ also forces one to score deformations in a way that allows efficient inference. For instance, parts in a part based models have to be arranged in a tree structure or else inference is intractable, but a tree structure might assign high scores to inconsistent part placements and implausible deformations. In contrast, our model can learn a bias for every deformation and simply penalizes unlikely deformations.

## 4. Generating the Deformation Dictionary

We want to create $\mathcal{D}$ in a data driven manner, capturing the observed deformations while excluding unlikely or

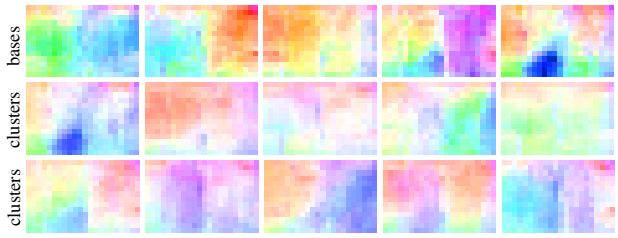


Figure 3: *Top*: The $k = 5$ estimated flow bases for bicycles. *Middle/Bottom*: the $m = 10$ centroids obtained by clustering bicycle templates in the projected space. See text.

implausible deformations. However, the training data does not come with annotated deformations, so our first task is to estimate the deformations present in the training data.

Given two feature vectors $x_1$ and $x_2$ in HOG space, our goal is to compute a coarse flow field $(u, v)$ that best deforms $x_1$ into $x_2$. In other words, for each cell in $x_1$ we want to find a likely "target" cell in $x_2$. We do this using a simple block-matching approach. For each cell $(i, j)$ in $x_1$, we take a 5x5 patch centered on $(i, j)$ and find the best matching location $(i', j')$ in $x_2$ in its local neighborhood ($\pm 2$ cells). The resulting flow is $u_{i,j} = j' - j$ and $v_{i,j} = i' - i$.

In order to measure the quality of a match, we use the dot product between the 5x5 HOG templates in $x_1$ and $x_2$ as the scoring function. Finally, we observe that the flow fields can be better estimated using whitened HOG features (WHO) [13]. Whitening HOG suppresses correlations across cells and has proven useful in a number of scenarios.

### 4.1. Generating Candidate Deformations

Given a set of examples belonging to a single component or aspect, we can compute the flow as above between the mean whitened feature vector $\bar{x}$ and each of the examples. Using these individual deformations, we define a two stage procedure for generating the deformation dictionary $\mathcal{D}$.

**Regularization.** The individual flow fields tend to be noisy and have large discontinuities (each cell may move independently). However, we have a large set of flows, and this allows us to regularize the result. We do PCA on the flows and project each flow to the top $k = 5$ PCA axes. This makes each flow smoother and allows for subpixel flows.

Figure 2 shows the raw and regularized flows, denoted by $D^*$ and $D$ respectively[1], for a single bicycle exemplar. The WHO feature vector $x$ has a non-trivial deformation relative to the mean template $\bar{x}$ resulting from the out-of-plane rotation of the front wheel. The projected flow $D$ removes much of the noise from the raw flow $D^*$ while preserving the overall deformation. Observe the warped template $D^T\bar{x}$ matches the feature vector $x$ better than $\bar{x}$.

[1]In a slight abuse of notation we use $D$ to refer to both the flow field and its corresponding deformation matrix.

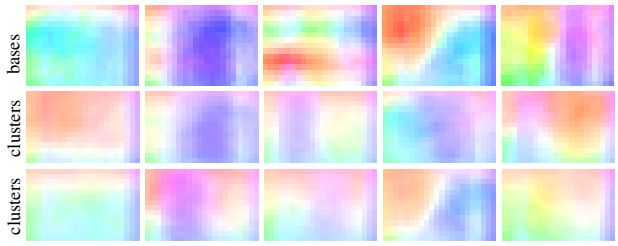


Figure 4: *Top*: The top $k = 5$ estimated flow bases obtained by computing PCA over flows from *all* categories. *Middle/Bottom*: the $m = 10$ centroids obtained by clustering bicycle templates in the *common* projected space. The flows obtained are much smoother than in Figure 3; on the other hand, some category specific deformations may be missed.

**Clustering.** To select a set of representative flows we cluster the projected flows into $m = 10$ clusters using K-Means. We use the mean flow in each cluster as a candidate deformation $D$, thus forming our deformation dictionary $\mathcal{D}$.

Figure 3 shows the top $k = 5$ PCA bases for the bicycle category (top row) along with the $m = 10$ cluster centers (middle/bottom rows). Observe that the flows capture complex motions such as scaling and foreshortening effects that cannot be explained by simple part translations. Moreover, the flows are smooth implying that the deformed templates will retain their holistic appearance and won't 'fall apart', as is often the case with independently moving parts.

### 4.2. Shared Deformation Bases

While we expect each category to deform in unique ways, there may be groups of categories that deform similarly (e.g. bicycles and motorbikes). Moreover, some regularities will be common to all categories. This suggests that deformation bases can be shared among subsets or even across all categories. Additionally, utilizing a common basis may help regularize the estimated deformations, especially when there are few training examples.

To see if this intuition is correct, we perform experiments on PCA bases constructed using deformations pooled across similar categories and also across all categories. Then for each category, as before, we project the exemplar deformations to the top 5 PCA axes and cluster the deformations to produce the deformation dictionary $\mathcal{D}$. This allows each category to share the space in which the clustering is performed while retaining its own unique dictionary.

Figure 4 shows the 5 PCA bases computed across *all* categories and the resulting deformation dictionary for bicycles (contrast with Figure 3). Note how they are smooth and capture non-trivial deformations, but may fail to capture some category specific details. Detailed experiments using shared deformation bases are given in Section 5.3.

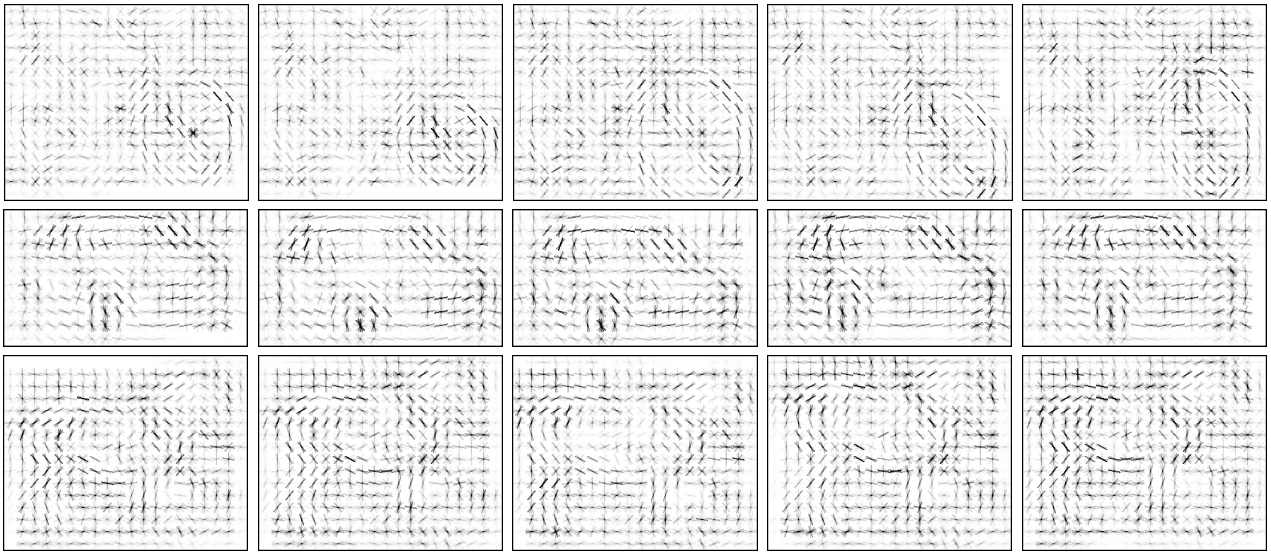

Figure 5: Visualizations of the learned model for bicycle (top), car (middle) and horse (bottom). Each row shows 5 (of 10) templates from *one* of the components in the model. All templates in a row are produced by deforming a canonical template.

## 5. Results

Like part models, our model shares parameters across several templates, but unlike part models, it does not allow for an exponential set of deformations. We do experiments to evaluate the impact of these two aspects of our model. Our first set of experiments compares against different types of mixture models and shows that parameter sharing plays a critical role. In our second set of experiments we compare to part-based models and show that our model can match (and improve upon) part-based models with our small but carefully constructed deformation dictionary.

In our final set of experiments, we evaluate the impact of sharing the deformation bases across multiple categories and show that such sharing can be beneficial.

We do all our experiments on the PASCAL VOC 2012 set [10]. We train all models on the train set and report average precision (AP) on the validation set. We use a re-implementation of the original DPM work [11] that we found easier to modify. We do non-max suppression on top of the output of the detector and clip the detected boxes to the image boundary, but do not do bounding box regression. Detection takes $4.4$s using our model (implemented as a vanilla mixture model) compared to $3.6$s for DPM.

### 5.1. Parameter Sharing

How important is parameter sharing for detection? We examine this question using several baselines that are comparable to our approach but do not share parameters. Let $n$ denote the number of components; each component has a low-resolution root filter and $m$ high-resolution fine filters computed using the same HOG template with $m$ discrete deformations. We propose three baselines:

| method | bike (100 samples) | bike (full data) |
|---|---|---|
| $n$-component | 30.4 | 40 |
| $nm$-component | 26 | – |
| $n$-comp$\times m$ fine | 36.2 | 44.8 |
| Ours | **38.9** | **46.5** |

Table 1: Comparison to various baseline models trained with varying amounts of data, see text for details.

1. An $n$-component model, with each component having just a low-resolution root filter. This model lacks the high-resolution filters and deformations.
2. An $nm$-component model, with each component having just a low-resolution root filter. This baseline is a standard mixture model with as many components as we have high-resolution templates.
3. An $n$ component model where each component has a root filter and $m$ high-res templates trained *without* parameter sharing. This baseline is closest to our model but the templates are not related by deformations.

We train the models on both the full training set and a subset of 100 examples to gain insight into performance with low amounts of training data. In both experiments we set $m = 10$ and we use $n = 3$ components for the full training dataset and $n = 1$ for the smaller dataset.

Results on bicycles are shown in Table 1. As expected, the $n$-component model performs poorly, since a small number of components is not enough to capture intra-class variation. The $nm$-component model performs worse, as it severely overfits (on the full data training failed to converge due to overfitting issues). Using $n$ components with $m$ templates per component does significantly better, but is

| | n-comp | n-comp ×m-fine | DPM | ours | ours -common |
|---|---|---|---|---|---|
| plane | 27.3 | 32.0 | **40.5** | 36.6 | 35.8 |
| bike | 40.0 | 44.8 | 45.2 | 46.5 | **46.9** |
| bird | 1.3 | 3.0 | 1.9 | 4.2 | **4.4** |
| boat | 4.5 | 4.3 | 4.0 | **5.4** | 4.9 |
| bottle | 16.4 | 17.6 | 19.2 | 20.1 | **20.7** |
| bus | 47.3 | 55.1 | 53.0 | 54.9 | **55.3** |
| car | 27.3 | **36.4** | 35.3 | 36.3 | 35.5 |
| cat | 9.1 | 17.8 | 18.6 | **19.8** | 19.7 |
| chair | 7.1 | 9.8 | **14.6** | 11.7 | 11.5 |
| cow | 5.9 | 11.3 | 10.3 | **13.4** | 12.3 |
| table | 2.9 | 6.0 | 3.9 | **8.9** | 8.0 |
| dog | 5.8 | 7.1 | **10.7** | 7.3 | 7.7 |
| horse | 23.5 | 33.5 | 32.9 | **35.6** | 34.7 |
| moto | 27.1 | 26.4 | **31.4** | 31.1 | 30.8 |
| person | 28.2 | 36.7 | **38.5** | 36.4 | 36.8 |
| plant | 1.7 | 4.3 | 4.0 | **6.6** | 5.1 |
| sheep | 23.5 | 25.6 | 25.5 | **27.0** | 27.0 |
| sofa | 5.4 | 7.8 | 11.6 | 11.0 | **13.6** |
| train | 22.0 | 33.3 | 32.7 | **35.0** | 32.9 |
| tv | 21.3 | 33.4 | **34.2** | 32.7 | 31.9 |
| mean | 17.4 | 22.3 | 23.4 | **24.0** | 23.8 |

Table 2: Average Precision (AP) on PASCAL VOC Val 2012. The first 2 columns are the mixture model baselines described in Section 5.1. The 3rd and 4th columns show results for DPM and our approach. The last column shows results with our model using a common deformation basis.

outperformed by a large margin by our model.

Results for all categories on the full set are in Table 2. We omitted the $nm$-component model owing to overfitting issues. DPM (column 3) and our approach (column 4) both share parameters and do much better than the two mixture model baselines, clearly demonstrating the benefit of parameter sharing.

## 5.2. Comparison with Part Models

Do we need an exponential number of deformations as in the DPM to get good performance? Figure 1 provides some intuition. The figure shows a small set of templates sampled from a trained DPM model (sampled from among the exponential number of possible templates). For comparison, we also show the HOG templates in our trained model. Since the part locations in the DPM are independent given the root node, many of the possible part locations that the DPM scores highly are actually implausible. The HOG templates in our model appear more realistic. Figure 5 shows additional templates from our models.

Our model produces more plausible HOG templates, but the number of candidate deformations is much smaller. Is it still able to capture an adequately broad set of deformations for accurate detection? To answer this question, we compared our approach ($n = 3, m = 10$) with the DPM on all

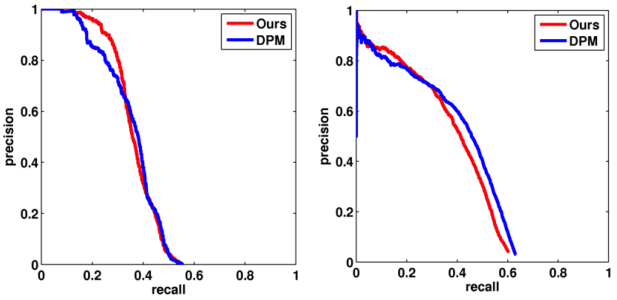


Figure 6: Precision/recall curves for car (left) and person (right) using our model (red) and the DMP (blue). Our model typically has higher precision with lower recall.

categories in the VOC 2012 validation set. The results are shown in Table 2. On average, our approach slightly outperforms the DPM, improving AP in 13 out of 20 categories.

This is a surprising and perhaps counter-intuitive result since part-based deformation models provide greater flexibility in handling deformations. Our conjecture is that while our model might not capture all possible deformations and thus may miss some objects, it is better able to prune out some spurious deformations that can lead to false positives. Examining the precision/recall curves for the various objects supports this hypothesis. Figure 6 shows representative curves for a class (car) where we see improvement and one (person) that we do not. In both cases, we see a boost in precision and a reduction in recall. However, in aggregate across categories the boost in precision outweighs the reduction in recall, leading to improved results in AP.

## 5.3. Deformation Generalization

While one should expect different categories to deform in different ways, we also expect there to be some regularities, such as smoothness. In addition, there may be groups of categories that deform similarly, for example, we may expect bicycles and motorbikes or cows and sheep to occur in similar poses. This suggests that some or all of these categories may share deformation bases. Sharing the bases may also help regularize the estimated deformations, especially when there are fewer training examples.

To see if this intuition is correct, we constructed PCA bases by pooling (a) all deformations seen across all categories and (b) deformations seen in similar categories (motorbike, bicycle and cow, sheep). Figure 4 shows the PCA bases computed across *all* categories and the resulting deformation dictionary for bicycles, see Section 4 for details. Since there is a large imbalance in training examples of people versus other categories, we do not include the person category when computing the shared PCA bases.

The last column in Table 2 shows the results we achieve if we use a single PCA basis across all categories on PASCAL VOC 2012 with the same parameters as before ($n =$

| category | category-specific | common | super-category | DPM |
|---|---|---|---|---|
| Bicycle | 46.5 | 46.9 | **47** | 45.2 |
| Motorbike | 31.1 | 30.8 | **31.6** | 31.4 |
| Cow | 13.4 | 12.3 | **13.7** | 10.3 |
| Sheep | **27** | **27** | 26.9 | 25.5 |

Table 3: Impact of clumping together similar categories for computing deformation basis.

$3, m = 10$). Interestingly, using a common PCA basis only causes a small drop in AP, but we are still on par with DPM, indicating that the deformations in different object categories do have a lot in common.

A common set of PCA bases is also likely to be less noisy in cases of insufficient training examples. When training bicycles with only 100 examples, a common basis achieves an AP of $39.6$ versus an AP of $38.9$ for a category-specific basis (DPM achieves an AP of $39.4$ on the same data).

We also hypothesize that similar categories can benefit from sharing deformations. Table 3 shows the performance if we share the deformation basis between motorbike and bicycle and between cow and sheep. In each case accuracy improves, and in most cases AP is even higher than using a category specific deformation basis.

## 6. Discussion

In this paper, we have proposed using a discrete set of deformations. However, we can also search for the optimal deformation within the space defined by our set of 5 PCA bases. Using a greedy search technique, we were able to obtain similar results to that of our discrete model. While the discrete approach is more computationally efficient, it may prove beneficial to search in a continuous space of deformations for some object categories.

In conclusion, we propose an approach to object detection that models deformations and appearance separately. We do so by constructing a deformation dictionary containing a discrete set of candidate flow fields. Interestingly, our model using a small number of deformations is able to improve upon the performance of part-based models that are capable of modeling a exponential number of deformations. In addition, we show that sharing deformation information across categories can lead to improved performance.

## References


[1] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011.

[2] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*, 2012.

[3] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *PAMI*, 23(6), 2001.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[6] S. K. Divvala, A. A. Efros, and M. Hebert. How important are 'deformable parts' in the deformable parts model? In *Parts and Attributes Workshop, ECCV*, 2012.

[7] B. Drayer and T. Brox. Distances based on non-rigid alignment for comparison of different object instances. In *Pattern Recognition*, 2013.

[8] O. Duchenne, A. Joulin, and J. Ponce. A graph-matching kernel for object categorization. In *ICCV*, 2011.

[9] I. Endres, V. Srikumar, M.-W. Chang, and D. Hoiem. Learning shared body plans. In *CVPR*, 2012.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9), 2010.

[12] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005.

[13] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012.

[14] M. Hejrati and D. Ramanan. Analyzing 3d objects in cluttered images. In *NIPS*, 2012.

[15] L. Ladicky, P. H. S. Torr, and A. Zisserman. Latent svms for human detection with a locally affine deformation field. In *BMVC*, 2012.

[16] J. J. Lim, A. Torralba, and R. Salakhutdinov. Transfer learning by borrowing examples for multiclass object detection. In *NIPS*, 2011.

[17] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: dense correspondence across different scenes. In *ECCV*, 2008.

[18] X. Liu, Y. Tong, and F. W. Wheeler. Simultaneous alignment and clustering for an image ensemble. In *ICCV*, 2009.

[19] B. Pepik, P. Gehler, M. Stark, and B. Schiele. 3D$^2$PM - 3D deformable part models. In *ECCV*, 2012.

[20] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3D geometry to deformable part models. In *CVPR*, 2012.

[21] H. Pirsiavash and D. Ramanan. Steerable part models. In *CVPR*, 2012.

[22] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Bilinear classifiers for visual recognition. In *NIPS*, 2009.

[23] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial occulsion. In *NIPS*, 2009.

[24] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, 2000.

[25] J. Winn and N. Jojic. LOCUS: Learning object classes with unsupervised segmentation. In *ICCV*, 2005.

[26] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[27] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? In *BMVC*, 2012.