

Shared Parts for Deformable Part-based Models

Patrick Ott and Mark Everingham
School of Computing
University of Leeds
{ott|me}@comp.leeds.ac.uk

Abstract

The deformable part-based model (DPM) proposed by Felzenszwalb et al. has demonstrated state-of-the-art results in object localization. The model offers a high degree of learnt invariance by utilizing viewpoint-dependent mixture components and movable parts in each mixture component. One might hope to increase the accuracy of the DPM by increasing the number of mixture components and parts to give a more faithful model, but limited training data prevents this from being effective. We propose an extension to the DPM which allows for sharing of object part models among multiple mixture components as well as object classes. This results in more compact models and allows training examples to be shared by multiple components, ameliorating the effect of a limited size training set. We (i) reformulate the DPM to incorporate part sharing, and (ii) propose a novel energy function allowing for coupled training of mixture components and object classes. We report state-of-the-art results on the PASCAL VOC dataset.

1. Introduction

Object category detection and localization is a fundamental goal for computer vision. Recent years have seen great advances in accurate recognition through adoption of methods from the machine learning community and developments in image descriptors and object modeling. Results reported by the annual PASCAL Visual Object Class (VOC) challenge [4] have shown steady progress on the large-scale and challenging data distributed as part of that challenge.

Most state-of-the-art approaches to object detection are based on a sliding-window framework [18, 3, 6, 16] or efficient variations thereof [11, 16], where a binary object/non-object classifier is applied to a descriptor extracted from an image window. A recent method using this approach, which stands out for its success on recent VOC challenges [4] is the deformable part-based model (DPM) of Felzenszwalb *et al.* [6]. Most previous sliding-window methods have used a fixed global window descriptor, *e.g.* the Histogram of Oriented Gradients (HOG) descriptor of Dalal & Triggs [3].

The DPM method incorporates a higher degree of *learnt* invariance by partitioning the object model into a set of local parts which are allowed to move around subject to soft spatial constraints. The use of parts allows the image descriptor to adapt to the underlying image structure, and potentially gives improved generalization by the independence assumptions in part appearance made by the model. While the notion of parts has a long history in computer vision [10], the use of *discriminative* part learning in the DPM method, in contrast to the predominance of generative approaches in other part-based models [12, 21, 9] has proven particularly effective. In addition, the DPM method uses the idea of a mixture model to capture the large variation in appearance that an object category may exhibit. Different detectors are learnt for different ‘aspects’ or modes of appearance of an object, *e.g.* frontal/lateral viewpoint. Each of these ‘mixture components’ is effectively a separate detector, consisting of a linear classifier applied to a global window descriptor and a set of parts specific to that detector. At training time, each mixture component is essentially trained independently, from a subset of training examples assigned to that component.

While the mixture model approach used by the DPM has been shown to be effective for a small number of mixture components *e.g.* representing front/side views of an object, attempting to improve accuracy by increasing the number of mixture components is not straightforward for several reasons: (i) since mixture components have separate parts the number of parameters increases linearly with the number of mixture components; (ii) since training examples are assigned to a single mixture component, the amount of training data per component decreases linearly with the number of components. In practice, the increase in number of parameters and coinciding decrease in training data results in poor generalization. In addition, the computational resources needed both at training and test time increase linearly with the number of parts and mixture components.

Contributions. In this work we propose extensions to the DPM which allow more powerful detectors to be built which make efficient use of the available training data.

The key idea is to *share* parts between detectors so that (i) the overall number of parameters is reduced, encouraging generalization from finite training data; (ii) training examples are shared across all relevant parameters, such that the paucity of available training data has less negative impact; (iii) computational expense at training and test time is reduced, giving potential for scaling to a large number of classes. We propose to share parts both (i) *within* object categories, *e.g.* a ‘wheel’ part may be shared across mixture components representing different viewpoints of a car; (ii) *across* object categories, *e.g.* the same wheel part may be shared by detectors for both cars and motorbikes.

We pose learning of the extended DPM with shared parts as minimization of a novel energy function allowing simultaneous learning of parts for multiple object classes and mixture components. This additionally offers a ‘modular’ reformulation of the original DPM [6] allowing for a more compact and intuitive explanation of the model.

Related work. Part-based models have received a great deal of attention in the literature. The DPM method of Felzenszwalb *et al.* [6] simultaneously learns the object detector (a binary SVM classifier) and parts without part-level training annotation, effectively casting the problem of part ‘discovery’ as a multiple instance SVM learning problem [1]. Zhu *et al.* [22] reformulate the DPM as a structural SVM learning problem [20], and incorporate the notion of a hierarchy of parts at different granularities. Vedaldi & Zisserman [17] also propose a structured output model for object detection which implicitly models parts by accounting for alignment of the features representing an object class. However, neither consider sharing parts.

Previous work on sharing parts across object models has focused on boosted classifiers, where ‘parts’ correspond to weak classifiers [15, 13]. Torralba *et al.* [15] propose a method for selecting weak classifiers simultaneously for a set of object class detectors, showing improved efficiency and accuracy with small training sets. Opelt *et al.* [13] use this approach to learn a ‘visual alphabet’ of shape and appearance which is shared among different object classes. A notable improvement made by the DPM over these methods is the ability to model different ‘aspects’ of an object, *e.g.* viewpoints, without training annotation.

In addition to methods adopting a ‘conventional’ classifier-based approach to object detection, Conditional Random Fields (CRF) have been used to represent object models having a dense arrangements of parts. A particularly elegant piece of work in this area is that of Winn *et al.* [19] which proposes the ‘layout consistent random field’. This combines local part detectors with a CRF which places constraints on neighboring parts *e.g.* that the middle of a car should be next to the front. Schnitzspan *et al.* [14] follow a similar approach and model part appearance and location as nodes in a latent CRF, using an EM algorithm to simultane-

ously discover parts and learn models of their appearance.

Earlier methods have incorporated the idea of part-based models in a generative fashion, by modeling the probability distribution over object appearance using a factorizable composition of object parts, which also allows for the possibility of sharing parts. Fergus *et al.* [8] propose a generative model for object detection based on constellations of parts. They define a joint probability density over shape, appearance, scale and part-level occlusion. Mikolajczyk *et al.* [12] propose a hierarchical representation which enables sharing of edge based features among several object classes. The method of Fidler & Leonardis [9] also adopts a hierarchical representation of an object class by using edgelet-type features shared among object classes. Zhu *et al.* [21] consider part sharing for multi-view and multi-object detection by proposing Recursive Compositional Models, which allow for hierarchical modeling of parts. However, the models rely on a complex inference scheme and have not been evaluated on the most challenging datasets [4]. As noted by the authors [21], their framework might require discriminative learning to “produce a system with high performance”. This is generally the case for generative models, which have been outperformed by relatively simpler architectures trained using state-of-the-art discriminative methods [4]. We take this into account and build our proposed method on the discriminative DPM which has proven performance, extending this approach by incorporating new mechanisms for part sharing among different viewpoints and object classes.

Outline. In Section 2 we review and reformulate the DPM method of Felzenszwalb *et al.* [6]. Section 3 describes (i) our proposed model for sharing parts; (ii) a learning scheme that allows for coupled learning of detectors for multiple object classes. Section 4 presents quantitative results and analysis on the PASCAL VOC [4] dataset. We offer conclusions in Section 5.

2. Deformable Part Model

In this section we review the DPM [6] and reformulate it in order to incorporate the notion of shared parts. The proposed reformulation is ‘modular’ in the sense that we define the classifier in terms of individual part responses rather than as operating on a feature vector formed by concatenating a subset of features determined by the part positions, *c.f.* [6]. Fig. 1 provides an overview of the functionality of the DPM and compares it to our proposed framework.

Features. We adopt a standard sliding window detector scheme, extracting features x for each window of the image. As in previous work [6] we use features based on the HOG descriptor [3], in which local histograms of gradient orientation are computed in a set of square ‘cells’ laid out on a regular 2D grid. Adjacent cells are aggregated and normalized to give ‘blocks’ with greater invariance to local lighting and spatial deformation.

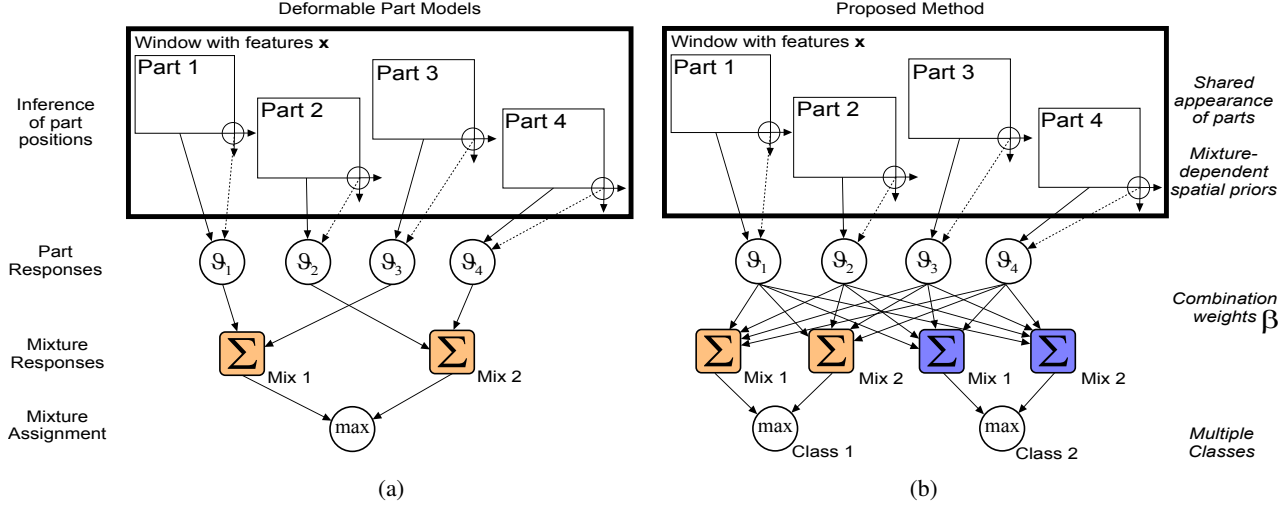


Figure 1. Comparison of the detection framework of Felzenszwalb *et al.* [6] and the framework proposed in this paper. In both frameworks part responses are derived by summing responses to linear part filters (solid lines) and spatial priors (dotted lines). (a) Felzenszwalb *et al.* framework: parts are strictly assigned to only one mixture component and object class models are trained independently. (b) Proposed framework: part responses are linearly combined in each mixture component. Multiple object detectors can be trained simultaneously while sharing part responses among them.

Parts. Because of the 2D grid structure of the HOG descriptor, object parts can naturally be defined by extracting a subset of contiguous features from \mathbf{x} . Let $\mathbf{z} = \langle z_x, z_y \rangle^\top$ represent the position of a part within a window, and let us further assume parts are rectangular and of fixed size. The features describing the image area the part covers are represented by $\phi(\mathbf{x}, \mathbf{z})$, *i.e.* the operator $\phi(\cdot)$ extracts a rectangular patch of features from the 2D structure of \mathbf{x} .

The position \mathbf{z} of a part is not static but inferred at test time, with soft constraints on its position learnt at training time. By allowing parts to move the classifier is essentially re-aligned to the underlying image structure, increasing invariance and improving generalization.

Inference of part positions. Each part is linked to an anchor position and incurs a penalty for moving too far away from its anchor. The displacement of a part with respect to an anchor position $\mathbf{a} = \langle a_x, a_y \rangle^\top$ is represented by

$$\psi_{\mathbf{a}}(\mathbf{z}) = \left[(a_x - z_x)^2, (a_x - z_x), (a_y - z_y)^2, (a_y - z_y) \right]$$

By defining a linear cost in terms of this displacement $\mathbf{v} \cdot \psi_{\mathbf{a}}(\cdot)$, which is a quadratic function of the part's offset from its anchor position, inference of the part positions is made efficient using the generalized distance transform [5, 6].

At test time, given features \mathbf{x} the best placement \mathbf{z}^* of a part is chosen to maximize the following objective function:

$$\vartheta(\mathbf{x}, \mathbf{w}, \mathbf{v}, \mathbf{a}) = \max_{\mathbf{z} \in \mathcal{Q}} \{ \mathbf{w} \cdot \phi(\mathbf{x}, \mathbf{z}) - \mathbf{v} \cdot \psi_{\mathbf{a}}(\mathbf{z}) \} \quad (1)$$

where \mathcal{Q} is the space of all possible part placements, \mathbf{w} is a vector of weights representing the ‘part filters’ (a linear classifier capturing part appearance) and \mathbf{a} is an anchor position. We call the vector \mathbf{v} the *spatial prior* of the part

since $-\mathbf{v} \cdot \psi_{\mathbf{a}}(\mathbf{z})$ defines a (un-normalized) Gaussian log likelihood over the part position.

This objective moves parts into positions where the part filters score highest while the displacement cost soft-bounds the score by penalizing part displacement w.r.t. the anchor position \mathbf{a} . In the remainder of the paper we refer to $\vartheta(\mathbf{x}, \mathbf{w}, \mathbf{v}, \mathbf{a})$ as a *part response*.

Window responses. We now briefly outline how part responses defined by Eqn. 1 are combined into *window responses* which can be ranked and subjected to non-maximum suppression to provide detections. Let us assume that there is a set of p parts with part filters $\mathcal{W} = \{ \mathbf{w}^1, \dots, \mathbf{w}^p \}$, spatial priors $\mathcal{V} = \{ \mathbf{v}^1, \dots, \mathbf{v}^p \}$ and anchors $\mathcal{A} = \{ \mathbf{a}^1, \dots, \mathbf{a}^p \}$.

To obtain a window response corresponding to confidence that the window contains the object class of interest, the individual part responses are accumulated by summing and adding a bias term:

$$g(\mathbf{x}) = b + \sum_{i=1}^p \vartheta(\mathbf{x}, \mathbf{w}^i, \mathbf{v}^i, \mathbf{a}^i) \quad (2)$$

When computing the confidence $g(\mathbf{x})$, part positions $\mathcal{P} = \{ \mathbf{z}^1, \dots, \mathbf{z}^p \}$ are inferred by Eqn. 1. Note that because the part responses are accumulated by summing, the position of each part can be inferred independently [6]. This also leads to a ‘modular’ view of the framework – the final classification confidence is defined as a simple sum of individual part responses. We return later to the question of whether such a simple accumulation scheme is the best choice.

Mixture model. To cope with different modes of appearance of an object class, for example viewpoints, the DPM uses a set of window classifiers in the manner of a mixture model. For each mixture component, or cluster of appearance, a classifier is trained using the corresponding training examples (mutually exclusive to other mixture components). The final classification confidence $h(\mathbf{x})$ for a window with features \mathbf{x} is then computed as the maximum over all the component classifiers $g^i(\mathbf{x})$:

$$h(\mathbf{x}) = \max_{i=1\dots d} g^i(\mathbf{x}) \quad (3)$$

where d is the number of mixture components. In the following we refer to function $h(\cdot)$ as the *detector* and to the functions $g^i(\cdot)$ as the mixture components. The particular mixture component m^* satisfying the above equation, *i.e.* $m^* = \arg \max_m \{g^m(\mathbf{x})\}$, is referred to as the *mixture assignment* of the example with features \mathbf{x} .

3. Sharing Parts

In our extension to the DPM we propose to share parts among mixture components. Since notionally, and empirically, the mixture components correspond to different viewpoints of an object, it is natural to have parts which may appear in multiple mixture components, *e.g.* a part which is visible from a range of views. Representing such a shared part explicitly, rather than requiring that an identical part be learnt in a set of mixture components, enables the learning of stronger models given a finite training set.

Sharing parts also allows us to learn stronger models in the sense that efficient modeling of ‘intermediate’ visual modes, such as three-quarter views of an object, becomes possible. Consider for example a car detector with frontal and lateral- mixture components. Individually both mixture components will respond rather weakly when presented with a car from a three-quarter viewpoint. However, by ‘blending’ their part-responses we can increase the response to such an example.

Fig. 1(b) provides an overview of the proposed framework using part sharing and illustrates the differences compared to the standard DPM implementation, *c.f.* Fig. 1(a).

Shared parts. While by definition we assume that the *appearance* of a part is similar independent of mixture component *e.g.* viewpoint, it is reasonable to assume that the *spatial* configuration of parts should vary. Consider *e.g.* a set of mixture components representing a car seen from viewpoints differing in azimuth from 3/4 rear to 3/4 frontal views – as the viewpoint changes the appearance of a wheel changes modestly, but the relative position of the front and rear wheels changes grossly over the range of viewpoints. We therefore propose a model of part sharing in which appearance is shared across mixture components, but anchor positions and spatial priors are unique to each component.

Additionally we learn ‘combination weights’ β for each part in each mixture component which represent (i) whether we expect to observe the part in a particular mixture component – this is important since some parts might only be visible in particular components *e.g.* in particular viewpoints; (ii) the discriminative ability of a part relative to others in the same component – this is important in ‘calibrating’ the reliability of different parts.

The response of the l th mixture component for a window with features \mathbf{x} is accordingly defined as

$$g^l(\mathbf{x}) = b_l + \sum_{i=1}^p \beta_i^l \vartheta(\mathbf{x}, \mathbf{w}^i, \mathbf{v}^{l,i}, \mathbf{a}^{l,i}) \quad (4)$$

Note that the part filters are shared (*i.e.* a single superscript i for all parts) while spatial priors and anchors are not shared (two-component superscript l, i). As a result, spatial priors and anchors are each linked to a specific mixture component. β can be interpreted as a matrix in which the l th row β^l represents the combination weights used in the l th mixture component.

3.1. Learning

Learning a model consists of estimating parameters $\{\mathcal{W}, \mathcal{V}, \beta\}$ such that they generalize well to unseen data. Recall that $\{\mathcal{W}, \mathcal{V}\}$ define the appearance and spatial configuration of the parts while β defines how those parts are linearly combined in the mixture components. We first explain how to learn these parameters for a single object class, and then provide an extension to model multiple object classes simultaneously while sharing parts among them.

We are given a training dataset $\mathcal{T} = \{(\mathbf{x}^1, y^1), \dots, (\mathbf{x}^n, y^n)\}$ where \mathbf{x}^k is the feature vector of the k th training window and $y^k \in \{-1, +1\}$ is the class of the window (non-object/object). We propose a novel and holistic learning scheme that allows for coupled learning of all parameters, while also inferring latent variables. These latent values are the part placements and mixture assignment of each training example.

Energy function. To learn the mixture components we define an energy function E consisting of two parts: (i) a regularization term R , which ensures good generalization of the learnt detector on unseen data and (ii) a loss term, measuring how well the detector predicts the training data:

$$E(\mathcal{W}, \mathcal{V}, \beta) = \lambda R(\mathcal{W}, \beta) + \sum_{k=1}^n L(y^k, h(\mathbf{x}^k)) \quad (5)$$

where λ sets the relative importance of the regularization term compared to the loss term. This loss term accumulates prediction errors in terms of a loss function $L(\cdot)$ which determines how deviations of $h(\cdot)$ from the target values y^k should be penalized. We use the popular hinge loss:

$$L(y^k, h(\mathbf{x}^k)) = \max\{0, 1 - y^k h(\mathbf{x}^k)\} \quad (6)$$

Note that the proposed energy function allows for simultaneous learning of multiple mixtures as $h(\cdot)$ always picks the mixture component that satisfies the inference scheme of Eqn. 3.

Multi-class extension. At this point the energy $E(\cdot)$ is constrained to a single object class. We can extend our formulation to multi-class learning by reformulating the loss term so that it accumulates the losses of multiple detectors:

$$E(\mathcal{W}, \mathcal{V}, \beta) = \lambda R(\mathcal{W}, \beta) + \sum_{s=1}^r \sum_{k=1}^{n_s} L(y^{s,k}, h^s(\mathbf{x}^{s,k}))$$

where r is the number of object classes we are considering for training. Each object class has its own training dataset $\mathcal{T}^s = \{(\mathbf{x}^{s,1}, y^{s,1}), \dots, (\mathbf{x}^{s,n_s}, y^{s,n_s})\}$ and detector $h^s(\cdot)$, but note that these detectors may share part responses according to the learnt combination weights.

Regularization. The regularization term $R(\cdot)$ encourages good generalization by controlling over-fitting. This is usually achieved by soft-bounding the responses of the mixture components (Eqn. 4), *e.g.* by penalizing the ℓ_2 -norm of the part filters: $\sum_i \|\mathbf{w}^i\|^2$. In our case penalizing the ℓ_2 -norm of the part filters alone is not sufficient since β can still grow towards unreasonably high values.

We constrain all values of β to be positive and given that constraint we can rewrite an individual part response (Eqn. 1) as

$$\begin{aligned} \beta \vartheta(\mathbf{x}, \mathbf{w}, \mathbf{v}, \mathbf{a}) &= \beta \max_{\mathbf{z} \in \mathcal{Q}} \{\mathbf{w} \cdot \phi(\mathbf{x}, \mathbf{z}) - \mathbf{v} \cdot \psi_{\mathbf{a}}(\mathbf{z})\} \\ &= \max_{\mathbf{z} \in \mathcal{Q}} \{(\beta \mathbf{w}) \cdot \phi(\mathbf{x}, \mathbf{z}) - (\beta \mathbf{v}) \cdot \psi_{\mathbf{a}}(\mathbf{z})\} \end{aligned} \quad (7)$$

We recognize the distributivity of the combination weights in the above equation. Regularization hence requires the penalization of the combined entity $\beta \mathbf{w}$. and therefore leads to the following regularization term:

$$R(\mathcal{W}, \beta) = \sum_{l=1}^d \sum_{i=1}^p \left\| \beta_i^l \mathbf{w}^i \right\|^2 \quad (8)$$

For the case of multi-class learning the regularization term is simply extended by summation over all object detectors. We do not wish to regularize the spatial priors \mathcal{V} .

3.2. Optimization Scheme

Optimizing $E(\cdot)$ w.r.t. all parameters while *simultaneously* inferring latent variables is a challenging, non-convex optimization task. Potential approaches include Concave-Convex optimization [20] or by utilizing a scheme that decouples the learning of parameters $\{\mathcal{W}, \mathcal{V}, \beta\}$ from inference of part placements and mixture assignments [6].

We adopt the approach of Felzenszwalb *et al.* [6] and alternate between updating part placements and mixture assignments in an *outer loop* while learning strong mixture

components in an *inner loop*. Once the outer loop has been executed we ‘store’ the inferred part placements of all training examples in the set $\mathcal{Z} = \{\mathcal{P}^1, \dots, \mathcal{P}^n\}$ and the mixture assignments in set $\mathcal{M} = \{m_1, \dots, m_n\}$.

Note that the proposed detection framework can be interpreted as an extended Multiple-Instance (MI) SVM [1] scheme. In such a formulation all combinations of mixture assignments and part placements would make up a ‘bag’ of instances for each training example.

Inner loop. The inner loop assumes part placements \mathcal{Z} and mixture assignments \mathcal{M} are provided by the outer loop. To represent the idea that \mathcal{Z} and \mathcal{M} are fixed we change the notation of the energy function to $E_{\mathcal{Z}, \mathcal{M}}(\cdot)$. Within the inner loop strong mixture components are learnt by minimizing $E_{\mathcal{Z}, \mathcal{M}}(\cdot)$ w.r.t. all parameters $\{\mathcal{W}, \mathcal{V}, \beta\}$:

$$\min_{\mathcal{W}, \mathcal{V}, \beta} E_{\mathcal{Z}, \mathcal{M}}(\mathcal{W}, \mathcal{V}, \beta) \quad (9)$$

This optimization problem is still non-convex and we have found that trying to optimize it directly leads to solutions which generalize poorly. To resolve this problem we propose to alternate between minimizing $E_{\mathcal{Z}, \mathcal{M}}(\cdot)$ w.r.t. to $\{\mathcal{W}, \mathcal{V}\}$ while holding β fixed, and minimizing w.r.t. β while holding $\{\mathcal{W}, \mathcal{V}\}$ fixed. To optimize each of these sub-problems we use L-BFGS [2] for a fixed number of iterations, and repeat until convergence.

Bootstrapping. To learn stronger detectors false positive detections are extracted from training images. A window with features \mathbf{x} qualifies as a false positive if the detector scores inside the margin defined by the hinge loss, *i.e.* $h(\mathbf{x}) \geq -1$. Such false positives are added to the training dataset and $E_{\mathcal{Z}, \mathcal{M}}(\cdot)$ is re-optimized. This process is repeated for a fixed number of iterations or until the number of high-ranking false positives drops below a threshold.

Outer loop. In the outer loop updates are performed on the latent variables \mathcal{Z} and \mathcal{M} . New part positions and mixture assignments are computed for the positive training examples by evaluating the learnt detectors on the corresponding training images. The part positions and mixture assignments which give the greatest score *and* which overlap the ground-truth bounding box by at least 70% are selected. This latent update scheme is similar to the one presented by Felzenszwalb *et al.* [6].

4. Empirical Results

In this section we report the practical effects of sharing parts in a DPM as well as sharing parts over multiple object classes.

4.1. Implementation Details

Features, window size and bounding boxes. Similar to [6] two layers of HOG features [3] are used. The first layer (6×6 pixel cells) models fine appearance while the second

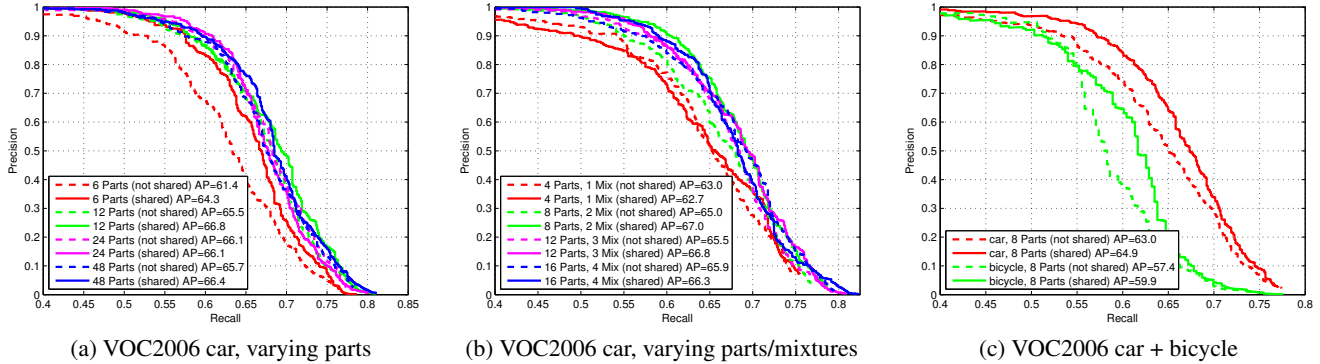


Figure 2. Results of part-sharing for VOC2006. (a) Precision/Recall curves for 6, 12, 24 and 48 parts with and without part sharing using 3 mixture components. (b) Precision/Recall curves for the car class, using 4 parts per mixture and an increasing number of mixture components. (c) Joint learning of car- and bicycle-detectors using 2 mixture components and 8 parts. In the case that part sharing is used we share all parts between all classes and mixtures. Note that in all figures we plot a constrained range of the recall.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	pers	plant	sheep	sofa	train	tv	mean
not shared	26.7	34.9	0.2	0.5	0.1	32.5	32.4	5.1	4.2	6.5	4.7	5.5	15.9	29.7	10.3	0.0	17.3	4.1	9.7	24.4	13.2
shared	24.7	38.2	0.0	1.2	0.2	33.3	37.7	7.3	1.4	4.6	8.1	8.1	21.5	31.8	11.5	6.3	17.0	5.1	9.6	23.9	14.6

Table 1. VOC2010 results (comp3, test). We show Average Precision (AP) for a model with 3 mixtures and 24 parts. Part sharing generally improves AP for the ‘rigid’ classes (bike, boat, bus, car, mbike) where part sharing across viewpoints is natural.

layer (12×12) models coarse appearance of an object class. We organize cells in 2×2 blocks overlapping by one cell and apply ℓ_2 -normalization [3].

To model truncation of an object class by the image boundary we adopt the approach of [7]. We only consider windows with a minimum of 50% visible.

Initial positive training examples are extracted from the training images and normalized to have a height of 64 pixels. The width of a window is set equal to the 75th percentile of all bounding boxes. We add a 6 pixel border around each window. Initial negative examples are obtained by extracting random windows not containing the object(s) of interest.

For each mixture component an ‘average’ bounding box is learnt which maximizes overlap with all ground-truth bounding boxes assigned to that mixture component. The bounding box is updated each time \mathcal{M} is updated. At test time and when executing the outer loop the learnt bounding box is used to predict an object’s extent.

Initialization of parts & mixtures. To compute the initial mixture assignment of each positive training example k -means clustering with d cluster centers is applied to the ratio of width and height of the bounding boxes, where d is the number of mixture components.

An equal number of parts is initially assigned to each mixture component, *i.e.* using p parts overall results in $p/(r \times d)$ per mixture, where r is the number of object classes being considered. We follow [6] by initially training individual sliding window detectors (including bootstrapping). To initialize a part position we search for the position that maximizes the sum of all positive detector weights covered by that part. The search is performed across all

mixture components. If a mixture component has already $p/(r \times d)$ initialized parts assigned to it, we do not consider it. To initialize anchors \mathcal{A} in the other mixture components we extract the positive weights from the area the part covers in the mixture component to which it has been assigned and search for the highest correlating position for this set of weights in the other mixture components. Once initialized the positive detector weights covered by the part in each mixture component are lowered by a factor of 0.5 to ensure that the next part will not be initialized to the same position.

Part filters are of size 8×8 HOG cells and use the first (fine) level of HOG features. In addition to the p movable parts, each mixture component is assigned a fixed root part, which acts on the second (coarse) level of HOG features and spans the entire window.

Training protocol. In the first step of training we learn initial part appearance by fixing part positions to the respective anchor positions and fixing all combination weights β to 0.5. We use $\min\{4rd, 10\}$ rounds of bootstrapping, where r is the number of different object classes for which detectors are being trained, and d is the number of mixture components for each detector

In the next step we ‘activate’ part sharing by allowing for optimization of β , initializing quadratic spatial priors \mathcal{V} to 0.01 and performing 6 rounds of latent updates (outer loops) updating $\{\mathcal{Z}, \mathcal{M}\}$. Each inner loop uses $\min\{4rd, 10\}$ rounds of bootstrapping and optimizes for all free parameters $\{\mathcal{W}, \mathcal{V}, \beta\}$. We found that this number of outer and inner loops causes convergence of the latent values as well as the free parameters. When optimizing $E_{\mathcal{Z}, \mathcal{M}}(\cdot)$ we re-initialize $\{\mathcal{W}, \mathcal{V}, \beta\}$ to the solution of the previous run.

During bootstrapping we scan $800/rd$ randomly selected training images for high-ranking false positives while during the outer loop updates on the latent variables we scan all images containing the object(s) of interest. Once completely scanned the image is resized by a factor of 1.2 until no window fits inside the image. The stride of the sliding window detector is set to 6 pixels. When bootstrapping we select one false positive per mixture per scale and stop the inner loop early if we extract less than 100 false positive windows for all mixture components.

Testing protocol. At test time we scan all images starting at a scale factor of 0.5, *i.e.* twice the original image size. Starting at a lower scale factor helps to detect objects smaller than the size of the window. The stride of the sliding window detector at test time is set to 6 pixels, and the resizing factor between scale-levels is 1.2. To create a final set of windows for a test image we adopt a greedy non-maximum-suppression scheme of [6] with an overlap threshold of 0.4. At test time we do not infer the mixture component (Eqn. 3) but predict all mixture components for all windows, which slightly improves recall.

Regularization & constraints. We use cross validation on the VOC_{val} datasets to set the parameter λ . Similar to Felzenszwalb *et al.* [6] we enforce a lower bound of 0.01 on the quadratic terms of the spatial priors in \mathcal{V} to ensure convexity and a ‘not-too-flat’ surface of the deformation cost. We additionally fix the ‘absolute’ components of the spatial priors \mathcal{V} to 0, causing them to be centered on their anchor position. We have found that this constraint does not harm performance and improves convergence of the learning scheme.

4.2. Experiments

For performance evaluation we use the PASCAL VOC2006/2010 [4] datasets and methodology – precision/recall curve with bounding box overlap of 50%. All detectors are learnt on the `trainval` datasets and we report precision/recall plots for the `test` datasets.

Terminology. Throughout this section we compare DPMs using part sharing to DPMs not sharing parts. We generally state that a DPM uses p parts. In case of part sharing there is no strict assignment between mixtures and parts. However, in the case that we do not share parts each DPM essentially represents a model of Felzenszwalb *et al.* [6] and we distribute the parts equally over the mixtures. For example, using 12 parts and 3 mixtures means 4 parts per mixture. Similarly using 8 parts in 2 detectors, each using 2 mixtures, means 2 parts per mixture per detector.

Single class. To evaluate single class performance we learn a detector for the car class of VOC2006 using 3 mixture components. We establish that part sharing provides better accuracy when using models with fewer parameters

by plotting precision/recall curves for models having 6, 12, 24 and 48 parts with and without part sharing in Fig. 2(a). We make the following observations: (i) the detectors using part sharing always provide better Average Precision (AP) than their counterparts, which do not use part sharing; (ii) the detector using 12 shared parts (solid green line) outperforms all other detectors including 24 (dotted magenta) and 48 (dotted blue) non-shared parts. This establishes our idea of performing ‘more with less’ given that the detector using 24 non-shared parts utilizes roughly twice and the detector using 48 non-shared parts roughly four times the number of parameters compared to the detector with 12 shared parts. Fig. 3(a) visualizes the positive detector weights, spatial priors and combination weights for a VOC2006 car detector.

We furthermore train a detector for the car class of VOC2006, fixing the number of parts *per mixture* to 4 while increasing the number of mixture components. Results are presented in Fig. 2(b) and can be summarized as follows: (i) increasing the number of mixtures from 1 to 2 mixture components improves AP significantly (62.7% to 67.0%) while increasing it further does not improve AP for the detectors using part sharing; (ii) AP consistently increases with more mixture components for the detectors not using part sharing, *i.e.* these detectors require more mixture components to reach peak performance *c.f.* detectors using part sharing. (iii) apart from the experiment using a single mixture component, sharing parts is always better than not sharing parts. The detectors with and without part sharing using a single mixture component are essentially the same and only differ in regularization.

Table 1 provides a full set of results for the VOC2010 (`comp3`) challenge for a model with 3 mixtures and 24 parts. We improve AP for 13/20 classes including the ‘rigid’ object classes, such as bicycle, bus, car, motorbike or table. Further improvements in AP could be obtained by increasing the number of mixture components, utilizing a better initialization of mixture components [7] or increasing the number of parts. Fig. 3(b) shows the positive detector weights, spatial priors and combination weights for a VOC2010 bicycle detector.

Multi-class. We evaluate multi-class performance by jointly training VOC2006 car and bicycle detectors with 2 mixture components each and 8 parts overall. Results are presented in Fig. 2(c). We observe an improvement in AP from 63.0% to 64.9% for the car- and 57.4% to 59.9% for the bicycle detector when sharing parts among all mixtures of both object classes.

5. Conclusions

We have proposed extensions to the DPM of Felzenszwalb *et al.* [6] by (i) allowing part appearance to be shared among mixture components while keeping spatial

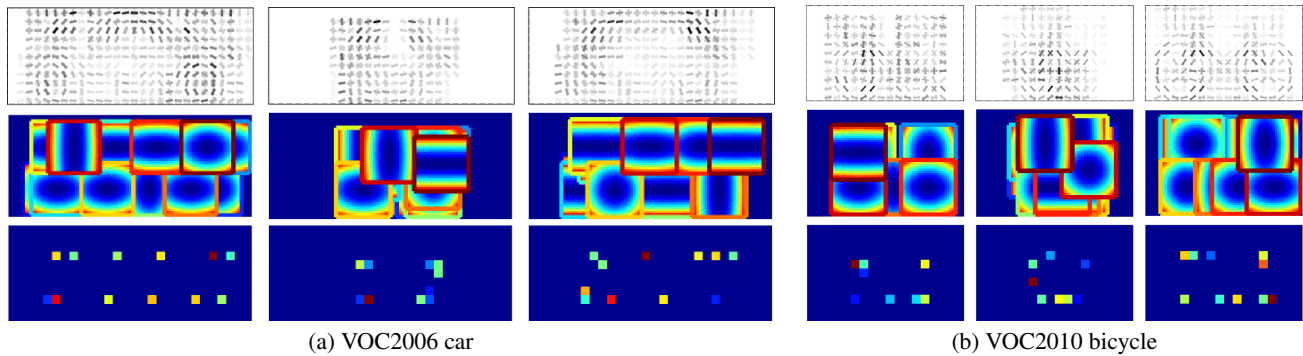


Figure 3. Positive detector weights (top row), spatial priors (middle row) and combination weights β (bottom row) for (a) VOC2006 car detector with 3 mixture components and 24 shared parts and (b) VOC2010 bicycle detector with 3 mixture components and 12 shared parts. Note the different location of the parts in the mixture components as well as the varying spatial priors for one and the same part.

configuration of a part independent w.r.t. each mixture and (ii) introducing an energy function for learning multiple object detectors and their mixture components simultaneously while sharing parts among all of them. We have experimentally demonstrated the positive effects of part sharing on the accuracy of the DPM. Sharing parts also results in better-performing yet more compact models (in terms of number of parameters) compared to models without shared parts. This is important given the extensive training protocol necessary to obtain good performance from the DPM.

In future work we aim to show the applicability of part sharing in enabling scaling to many more classes while maintaining the accuracy achieved by the DPM for the few classes in existing studies.

Acknowledgements. Patrick Ott is supported by EPSRC project EP/E010164/1 and EU FP7 project 214975 Co-Friend. Mark Everingham is supported by an RCUK Academic Fellowship.

References

- [1] S. Andrews, I. Tsochanaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2003.
- [2] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited-memory algorithm for bound constrained optimization. *SIAM Journal on SSC*, 16, 1995.
- [3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
- [4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 88(2), 2010.
- [5] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004.
- [6] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), 2010.
- [7] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. <http://people.cs.uchicago.edu/~pff/latent-release4/>.
- [8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.
- [9] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, 2007.
- [10] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, 22(1), 1973.
- [11] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *PAMI*, 31, Dec 2009.
- [12] K. Mikołajczyk, B. Leibe, and B. Schiele. Multi. object class detection with a generative model. In *CVPR*, 2006.
- [13] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *IJCV*, 80(1), 2008.
- [14] P. Schnitzspan, S. Roth, and B. Schiele. Automatic discovery of meaningful object parts with latent CRFs. In *CVPR*, 2010.
- [15] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 29(5), 2007.
- [16] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.
- [17] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial occlusion. In *NIPS*, 2009.
- [18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.
- [19] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006.
- [20] C.-N. J. Yu and T. Joachims. Learning structural SVMs with latent variables. In *ICML*, 2009.
- [21] L. Zhu, Y. Chen, A. Torralba, W. Freeman, and A. Yuille. Part and appearance sharing: Recursive compositional models for multi-view multi-object detection. In *CVPR*, 2010.
- [22] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010.