Factorized Appearances for Object Detection

Josep M. Gonfaus[a], Marco Pedersoli[b], Jordi Gonzàlez[a,*], Andrea Vedaldi[c], F. Xavier Roca[a]

[a] *Computer Vision Center and Universitat Autònoma de Barcelona, Catalonia (Spain)*
[b] *PSI/VISICS, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium*
[c] *Department of Engineering Science, Oxford University, Oxford, UK, OX1 3PJ, UK*

Abstract

Deformable object models capture variations in an object's appearance that can be represented as image deformations. Other effects such as out-of-plane rotations, three-dimensional articulations, and self-occlusions are often captured by considering mixture of deformable models, one per object aspect. A more scalable approach is representing instead the variations at the level of the object parts, applying the concept of a mixture locally. Combining a few part variations can in fact cheaply generate a large number of global appearances.

A limited version of this idea was proposed by [1] for human pose detection. In this paper we apply it to the task of generic object category detection and extend it in several ways. First, we propose a model for the relationship between part appearances more general than the tree of [1] which is more suitable for generic categories. Second, we treat part locations as well as their appearance as latent variables so that training does not need part annotations but only the object bounding boxes. Third, we modify the weakly-supervised learning of [2, 3] to handle a significantly more complex latent structure.

Our model is evaluated on standard object detection benchmarks and is found to improve over existing approaches, yielding state-of-the-art results for several object categories.

Keywords: Object recognition, deformable part models, learning and sharing parts, discovering discriminative parts.

[*] Corresponding author. E-mail: poal@cvc.uab.cat. Phone: +34 93 581 15 19. Fax: +34 93 581 16 70

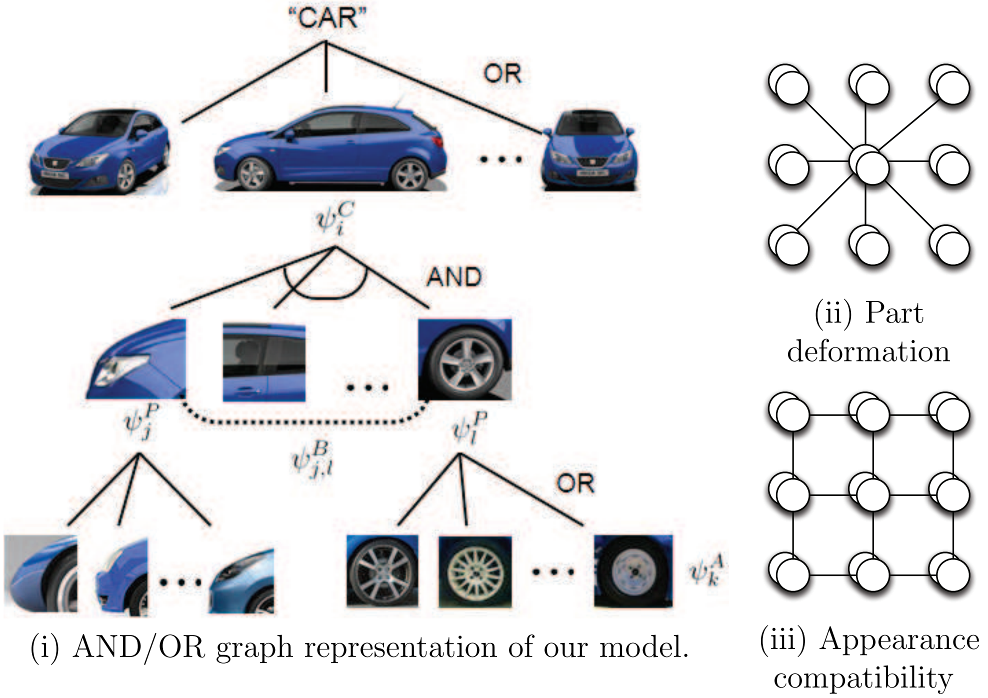
1. Introduction

Pictorial Structures (PSs) [4, 5] and their modern variants such as the Deformable Part Models (DPMs) [2] are probably the most popular models for object category detection. A PS is a collection of independent object parts whose spatial configuration is constrained by a system of elastic connections (springs). A DPM is a particular example of a PS that is learned by a discriminative method (latent SVM) and that uses linear classifiers on top of HOG features to describe the part appearance.

By design, DPMs model variations of the object that can be expressed as an independent motion of the object parts, which excludes, in particular, all the effects that cannot be expressed as an image deformations. An example are appearance variations due to the self occlusion of a three dimensional object rotating out-of-plane. Another example are three dimensional articulations or deformations: the appearance of a horse tail or of a scarf can change quite dramatically with motion. Since the linear HOG filters used in DPMs represent, by their very nature, a *unimodal* distribution of appearances, none of these variations can be modelled effectively by a DPM.

A simple way of incorporating multi-modal statistics in a DPM is to give up the linearity of the filters. For a discriminatively trained model, this means using a kernel other than a linear one, for example a radial basis function (RBF) kernel [6, 7]. Unfortunately, non-linear kernels have a major impact on the learning and testing complexity of the model [6]. In fact, if the bottleneck of a standard DPM is searching object parts at all image locations and scales [8], with a non-linear kernel this is further exacerbated by the need of comparing each candidate part appearance to a large number of support vectors (typically in the order of thousands [6]). Recent techniques for the efficient "linearization" of non-linear kernels [9, 7] do not help much here because they are limited to *additive* kernels, which, unlike the RBF ones, cannot be used to express multi-modal functions. Approximating RBF kernels very efficiently is still an open issue [10].

The alternative and more common approach for modelling multi-modal statistics with a DPM is to use a *mixture* of multiple DPMs [2, 11, 12], one for each object aspect (*e.g.*, the front, three-quarter, and side views of a car, as in Fig. 1). The multiple DPMs are "glued" together by a latent variable that selects which component to use for each given candidate object



(i) AND/OR graph representation of our model.

(ii) Part deformation

(iii) Appearance compatibility

Figure 1: **Structure of the object model.** (i) Our model can be interpreted as a OR-AND-OR tree, where aspect, parts and local appearance of each part are represented. (ii) As in DPM each part it is constrained to a center, in a star model. (iii) In contrast to [1] our appearance compatibility is learnt with a grid-like structure to adapt to any class.

instance. Compared to using non-linear kernels, the increase in complexity is bounded (linear in the number of components), and the latent variable explicitly captures *which appearance variant* is active, which may have a well defined semantic (*e.g.*, the object viewpoint).

Mixtures of DPMs are usually learned jointly to calibrate their scores and to determine which component to use for each training object instance [2, 11]. Other than that, the components are independent computationally and statistically. The latter issue is particularly severe as it limits the number of components that can be added to the model before overfitting starts to kick in. In practice, mixtures of DPMs can only model a handful of different object aspects. A more effective modelling scheme must exploit the fact that the various object aspects are by and large statistically *dependent*.

In this paper we extend the mixture-of-parts proposed in [1] for pose estimation to general object class detection. In essence, we investigate the simplest extension to DPMs that allows exploiting the statistical dependencies between different object aspects. In particular, we apply the notion of a *mixture of object appearances at the level of the object parts*, rather than to the object as a whole. In object detection, the class structure for any object and the part locations are generally unknown, and only bounding boxes are available. Therefore, the fully supervised method of [1] can not be used. In contrast, our model considers the object parts and their appearance as latent variables that should be jointly estimated during training. In order to properly constraint the latent variables, we adapt the weakly-supervised latent SVM algorithm [2, 3], with a hierarchical regularization as explained in Sect.3. In this way, local part appearances can be learned in an unsupervised way.

To illustrate our model, consider a standard mixture of DPMs [2]. Graphically, this can be represented by the AND-OR tree of Fig. 1(i) . The root node represents an OR node, and entails selecting one of a number of possible DPM models (corresponding to the three-quarter, side, and front views of the car). Each of these nodes is in turn connected to a small number of parts by an AND node, meaning that all those parts should be detected for the corresponding DPM. Our extension associates to each part a pool of different appearances to choose from, connected by an OR node. These multiple part appearances can represent *local* variations such as different styles of the wheel of a car, different shapes of the tail of a horse, or different rotations of the head of a person.

The key insight is that the model can now represent a much broader range of object variations *combinatorial rather than linear in the number of model components*, with a very modest increase in the number of model parameters (*e.g.*, just twice as many if two appearance models per part are considered). As we will see in Sect. 5, the impact on the inference and learning costs is also very modest.

Nevertheless, selecting parts independently from each other can yield unreasonable configurations (*e.g.*, two different wheel styles for the same car). To improve the model specificity and ultimately its precision, we consider on top of the AND-OR graph a mechanism to constrain the part activations to be *pairwise compatible*. (see Fig. 1(iii)). While in [1] the structure of the compatibility constraints have the same structure used for deformations, i.e. a tree, since our goal is to generic object categories whose structure may be unknown *a-priori*, local appearance compatibility is enforced on a planar graph instead, where each part is connected to its neighbourhoods. This structure is a loopy conditional random field (CRF) and it can be optimized efficiently with combinatorial techniques [29]. In this way, the actual structure of the object is learned during training by associating a weight to each pairwise term.

1.1. Related work

This section briefly summarises some of the main development in the vast literature on object detection, highlighting the methods that are most related to our contribution, see [13] for an extensive survey on object detection.

The simplest approach to improve the quality of an object detection system such as DPM is to improve the underlying image features. For example, [14] adds LBP features on top of the standard HOG representation, [15] incorporates color and [16] integrates local bag-of-features models and an object mask. However, a conscious study on the effect of adding more data and changing the structure model based only on HOG [17], reflects the necessity that more complex structures can better represent the objects and therefore increase the recognition performance. A counterexample is found in [18], which allows sharing of parts between different components, an approach orthogonal to ours. Unfortunately their results are well below the state-of-the-art in some international benchmarks. A possible reason is that, in our experience, sharing the same linear part filters between different DPMs yields serious calibration issues.

The concept of mixtures-of-parts is first introduced in [1]. Here the authors propose a tree-structured model for human pose estimation using multiples interchangeable mixtures for each part. Unfortunately, their model is valid only for articulated objects, where the structure and the degree-of-freedom of the parts is known. Furthermore, part locations are known which make the problem easier and a standard learning procedure, like SVMs can be used. Recently, other methods have also explored the case of fully-supervised training, where the part location is known [19, 20]. These seem to trade a higher cost of annotations for a better detection performance.

Other works have proposed to use multiple part appearances in contexts other than DPMs, but they usually require a significant amount of supervision. [21] use AND-OR graphs to parse articulated objects, but the position of the parts (limbs) is known beforehand. Similarly, in [22], the authors make use of production scores to capture the co-occurrence costs. Poselets [23] learn a large mixture of human parts, each with his own appearance, and associate them to “fragments of pose”. These methods have some interesting properties but require a very large quantity of annotated data.

In [24], the authors introduced multiple instance learning for object modelling by learning automatically the object parts and their locations from a set of object bounding boxes. The same mechanism, but implemented by means of latent variables, has been used extensively in the learning of DPMs [2], including determining object bounding boxes, parts, and aspects, and is further extended in this work to capture multiple part appearances. Finally, the layout of our baseline model is a simplified version of [11] where we use a single layer of parts in a regular grid, still obtaining similar performance.

The grammar framework described in [3] does not require ground-truth annotations on the position of the parts. However, that grammar needs to be carefully hand-tuned to represent the object of interest (humans). Since grammars cannot yet be learned automatically, we prefer to choose a model that can be adapted to any type of class, so we select a general structure based on simple pairwise connections between the parts, forming a CRF over parts appearance.

CRFs and latent variables have been used in the modelling of object categories in [25]. There the authors model an object as a set of patches and activate them by computing a minimum-spanning tree. However, the representation is too weak to obtain satisfactory performance on challenging international benchmarks such as the PASCAL VOC.

While in our work we use multiple appearances to render complex object configurations, rank constraints during learning [26] or a sparse representation on the learned model [27] are used to represent the object parts as a linear combination of a reduced set of basis. These methods contribute to make the inference faster having to evaluate a reduced set of parts, but does not help on improving detection, as instead our model does. A similar idea is used in [28] to speed up multi-class object detection, by using a coarse-to-fine taxonomy of parts among classes.

2. Object Model

This section introduces our deformable object model combining: (i) a small number of global components that capture radically different object viewpoints (*e.g.*, the front and side of a car), (ii) a number of movable parts for each component to model deformations and (iii) a number of appearance models for each part, to represent multiple variations of their appearance.

Next, we give a formal definition of the model, and we specify the score obtained by matching the model to an image for a given configuration of the parts.

AND-OR model. Let $\mathbf{x}$ be an image. The score $A_j(\mathbf{y};\mathbf{x},\mathbf{w})$ of matching a single part given its location/scale at rest $\mathbf{y} = (y_x, y_y, y_s)$ is obtained by trading off the cost of a part displacements $\mathbf{z} = (z_x, z_y, z_s)$ with the quality of the resulting appearance match:

$$A(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_{\mathbf{z}} \left\langle \psi^A(\mathbf{w}), \phi^A(\mathbf{x},\mathbf{y}+\mathbf{z}) \right\rangle + \left\langle \psi^D(\mathbf{w}), \phi^D(\mathbf{z}) \right\rangle . \tag{1}$$

Here $\phi^A(\mathbf{x},\mathbf{y}+\mathbf{z})$ is the HOG descriptor extracted from image $\mathbf{x}$ at location $\mathbf{y}+\mathbf{z}$ and $\phi^D(\mathbf{z})$ is a descriptor of the deformation (for example defining $\phi^D(\mathbf{z})$ as the vector of the squared displacements implements a quadratic spring). The vector $\mathbf{w}$ collects the parameters for the part and the operators ψ^A and ψ^D simply extract the blocks of parameters corresponding respectively to the appearance and the deformation.

Next, we extend $\mathbf{w}$ to include multiple part parameters (appearance and deformation) and introduce corresponding operators $\psi^A_k(\mathbf{w})$ to extract them. We can therefore associate each $\psi^A_k(\mathbf{w})$ to a learned appearance for a certain part as represented in Fig. 1. The appearance with the highest score is used to match the part to the image:

$$P(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_{k} A(\mathbf{y};\mathbf{x},\psi^A_k(\mathbf{w})). \tag{2}$$

This has the function of a OR node as shown in Fig. 1. Summing over a number of parts $j \in \mathcal{P}$ results in the score for the aspect:

$$C(\mathbf{y};\mathbf{x},\mathbf{w}) = \sum_{j} P_j(\mathbf{y}+\mathbf{h}_j;\mathbf{x},\psi^P_j(\mathbf{w})). \tag{3}$$

This is equivalent to an AND node in Fig. 1. ψ^P_j contains therefore the model parameters of the multiple appearances and deformations of a part j, while $\mathbf{h}_j = (h_x, h_x, h_s)$ is the part *anchor*, *i.e.* the location of the part with respect to the object centre. Finally, $\mathbf{w}$ is extended one last time to include multiple aspects and the score of the whole model is given by of the best matching aspect:

$$O(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_{i} C_i(\mathbf{y};\mathbf{x},\psi^C_i(\mathbf{w})). \tag{4}$$

Again, this correspond to a logical OR over the object aspects modelled by ψ_i^C as represented at the top of the AND-OR tree in Fig. 1. To summarise, the score of the model is given by

$$O(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_i \sum_j \max_k A(\mathbf{y}+\mathbf{h}_{i,j};\mathbf{x},\psi_{i,j,k}(\mathbf{w})) \tag{5}$$

where for compactness we defined $\psi_{i,j,k}(\mathbf{w}) = \psi_i^C(\psi_j^P(\psi_k^A(\mathbf{w})))$ and we denoted by $\mathbf{h}_{i,j}$ the anchor of the part j of the aspect i.

Loopy CRF model. In order to limit the number of possible part combinations to the ones that are meaningful, a set of additional constraints in the form of a CRF with loops is introduced. These constraints encourage neighbour parts to be assigned a compatible appearance, as automatically estimated from the frequency of co-occurrences on the training set. This set of part relations is modelled by a graph $\mathcal{G} \subset \mathcal{P} \times \mathcal{P}$ with an edge per constraint. For each constraint, consider a matrix $\mathbf{v}_{mn}$ where $\mathbf{v}_{k_1,k_2}$ is the cost of activating the appearance k_1 of the first part together with the the appearance k_2 of the second part. Consider also the scoring function

$$B(k_1,k_2;\mathbf{v}) = \sum_m \sum_n \mathcal{I}(k_1=m)\mathcal{I}(k_2=n)\mathbf{v}_{m,n}, \tag{6}$$

where $\mathcal{I}$ is the indicator function of an event. Instead of maximising independently over each part appearance as in (2), now the model optimises jointly over all parts, while accounting for the pairwise constraints:

$$\begin{aligned} C^{CRF}(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_{\mathbf{k}} & \sum_{j\in\mathcal{P}} A(\mathbf{y}+\mathbf{h}_j;\mathbf{x},\psi_{j,k_j}(\mathbf{w})) \\ & + \sum_{(j,l)\in\mathcal{G}} B(k_j,k_l;\psi_{j,l}^B(\mathbf{w})) \end{aligned} \tag{7}$$

where $\mathbf{k} = [k_0, k_1, ..., k_n]$ is a vector appearance labels, one for each part, and $\psi_{j,k}(\mathbf{w}) = \psi_j^P(\psi_k^A(\mathbf{w}))$. Finally $\psi_{j,l}^B$ are the parameters of the pairwise constraints between the parts (j,l), represented with a dashed line in Fig. 1.

Rewriting the final score for the formulation with pairwise appearance constraints gives:

$$\begin{aligned} O^{CRF}(\mathbf{y};\mathbf{x},\mathbf{w}) = \max_{i,\mathbf{k}} & \sum_j A(\mathbf{y}+\mathbf{h}_{i,j},\mathbf{x},\psi_{i,j,k_j}(\mathbf{w})) \\ & + \sum_{(j,l)\in\mathcal{G}} B(k_j,k_l;\psi_{i,j,l}^B(\mathbf{w})) \end{aligned} \tag{8}$$

Inferring the model at location $\mathbf{y}$ amounts to maximising (8). To do so efficiently, $\mathcal{G}$ is restricted to have a planar structure, where each part is connected with its horizontal and vertical neighbours (as in Fig. 1 (iii)). Dynamic programming is used to estimate the optimal displacement of each part first, and sequential reweighted trees [29] is used to solve the loopy CRF model and jointly estimate the optimal appearance of the parts. Considering that the number of parts is generally quite small, this does not compromise detection speed compared to a standard DPM.

3. Weakly-Supervised Learning

Learning uses weak supervision and, similarly to [2], requires only bounding boxes around instances of the object category of interest. The aspect, part locations, and part appearance components are *not* provided and are instead estimated automatically during learning as latent variables.

In detail, given a set of input images $\mathcal{X} = (\mathbf{x}_0, \mathbf{x}_1, .., \mathbf{x}_l)$, a set of object locations $\mathcal{Y} = (\mathbf{y}_0, \mathbf{y}_1, .., \mathbf{y}_p)$, and the locations of the negative samples $\mathcal{N} = (\mathbf{n}_0, \mathbf{n}_1, .., \mathbf{n}_n)$ (*i.e.*, locations that do not overlap with the ground truth object bounding boxes), the goal is to optimise the empirical risk

$$\begin{aligned} f(\mathbf{w}) = \frac{1}{2}\mathcal{R}(\mathbf{w}) \, + \, C \sum_{i=0}^{p} \mathcal{L}\left(\max_{\mathbf{s}\in\mathcal{S}_i} O(\mathbf{s}; \mathbf{x}_{l(i)}, \mathbf{w}) \right) \\ + \, C \sum_{i=0}^{\mathbf{n}} \mathcal{L}\left(-O(\mathbf{n}_i; \mathbf{x}_{l(i)}, \mathbf{w}) \right), \end{aligned} \tag{9}$$

where $\mathcal{L}(z) = \max\{0, 1-z\}$ is the hinge loss, $\mathbf{x}_{l(i)}$ is the image corresponding to the object location $\mathbf{y}_i$, and $\mathbf{s}$ denotes a small correction applied to the ground truth location estimated to better fit the model to the training data, similar to [2]. In particular, the adjustment is encoded by the (latent) variable $\mathbf{s}$, which is constrained to be in the vicinity of the ground truth locations, *i.e.*

$$\mathcal{S}_i = \{\mathbf{s} \in \mathbf{x}_{l(i)} : \mathrm{ovr}(\mathbf{s}, \mathbf{y}_i) > T\}, \tag{10}$$

where $\mathrm{ovr}(\mathbf{s}, \mathbf{y}) = \frac{area(B_\mathbf{s} \cap B_\mathbf{y})}{area(B_\mathbf{s} \cup B_\mathbf{y})}$ is the overlap score between the bounding boxes at location $\mathbf{s}$ and $\mathbf{y}$ respectively, and T is a threshold. Note that besides $\mathbf{s}$, the other latent variables are not shown in Eq.(9), but are still maximized inside O, as shown in the derivation of O in section 2.

3.1. Optimisation

Since the objective (9) is equivalent to a standard linear SVM (except for the treatment of the latent variables, as discussed below), optimisation uses the fast stochastic gradient descent technique of [30]. However, since the number of negative examples is extremely large (there is one negative for each image location that does not contain the object), the model is learned in stages, by collecting more and more hard negative examples based on the current version of the model. This procedure, known as constraint generation, cutting plane, or mining of hard negatives [2], can be shown to converge to the optimum of the objective function (9) in polynomial time.

The scoring function $O(\mathbf{y}; \mathbf{x}, \mathbf{w})$ of the model implicitly maximises over a number of parameters (aspect, part locations, part appearance selections) energies that are, ultimately, linear in $\mathbf{w}$. Since $O(\mathbf{y}; \mathbf{x}, \mathbf{w})$ is the max of convex functions, is itself convex in $\mathbf{w}$, and so is the composition with the hinge loss $\mathcal{L}(-O(\mathbf{n}_i; \mathbf{x}_{l(i)}; \mathbf{w}))$ *for the negative examples*. Unfortunately, for the *positive* examples the loss turns the sign the other way around and the composition is *not* convex. To address this issue, we follow the standard approach of converting the parameters that $O(\mathbf{y}; \mathbf{x}, \mathbf{w})$ marginalises over (aspect, part locations, part appearances) into latent variables and use the Concave-Convex Procedure (CCP) [31, 2, 11] to find a model $\mathbf{w}$ which is at least locally optimal. The CCP alternates estimating the latent parameters of the positive object instances and the model $\mathbf{w}$; in particular, the latent estimation step can be seen as hallucinating/estimating the model parameters that would be provided by an annotator in case of strong supervision.

3.2. Regularisation

In our model the latent variables are applied at two different levels. For the parts location, the latent variable is applied at *feature level*. That is, the model displaces each part to select the features that maximize its score. Instead, for aspect and part appearance, the latent variable is applied at *model level*, because the model selects which component for aspect and parts appearance better describe the features (*i.e.* maximises the score).

While latent variables at the feature level can be regularized with standard SVM $\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|^2$ regularization, for latent variables at model level the standard approach would fail. This is because when a latent variable at model level selects the best component, the others would be set to zero to force them to not contribute to the scoring. This procedure allows the model to represent OR-like nodes, but it is intrinsically unstable. Imagine that a component, during an iteration of latent variables estimation get assigned more samples than another. This would produce a new model, where the corresponding component has gained importance (*i.e.* its norm is higher). Thus, in the next iteration of latent variables estimation the component will probably get assigned even more samples. This would tend to produce a sparse representation with few strong components and the rest set to zeros and thus ineffective.

We can counterbalance this instability by using as regularizer the maximum of the squared norm of the parameters of each component rather than their sum. In [32] this procedure was used to better balance the final score among the different aspects of a model. Here, as the object parts are totally free to choose any appearance, this procedure becomes fundamental. We found that when using latent variables, balancing the various model components (aspects, part appearances) is very important. If the latent, using the standard SVM regulariser $\mathcal{R}(\mathbf{w}) = \|\mathbf{w}\|^2$ tends in fact to kill entire components by pushing their parameters to zero, ultimately lowering the performance of the model. [32] alleviate this problem by using as regulariser the maximum of the squared norm of the parameters of each component rather than their sum. In this way, there is no advantage in lowering the weights of any of the components with respect to any other. Since our model includes components at two levels (object and parts), we found that the appropriate extension of this idea involves maximising over components at both levels, as follows:

$$\mathcal{R}(\mathbf{w}) = \max_i \sum_j \max_k \langle \psi_{i,j,k}(\mathbf{w}), \psi_{i,j,k}(\mathbf{w}) \rangle \,. \tag{11}$$

Due to the recursive definition of $\psi_{i,j,k}(\mathbf{w})$, (11) must be computed recursively, for example by using dynamic programming. Other than that, incorporating it in the SGD solver is trivial as it suffices to compute a sub-gradient with respect to $\mathbf{w}$.

3.3. Initialization

The CCP procedure is a local optimization method therefore the initialization is very important in order to obtain a good solution. This amounts to finding a good initial value for the latent variables. As in the proposed model the latent variables are extended also to part appearance, their initialization is fundamental for good results. We propose a two steps approach to produce a good initialization for the parts appearance.

The location of the positive instances (**s** in (9)) is chosen to maximise the overlap between the ground truth bounding box and the one associated to the model. Initially deformations are set to be null. As in [32], the model has a flag indicating whether the object is facing left or right; this is an additional latent variable which is initialized by pre-clustering the training examples (denoted as FLIP).

We explore two initialization procedures for learning the local appearances. In the first, we learn all latent variables at the same time, by randomly assigning each local appearance to one of the labels. With this naive approach, we found that the model can easily get stuck in a local minima.

A better strategy is to using a two step sequential procedure (denoted as SEQ). A standard (one appearance) DPM model is first learned. Then, the learned model is applied again to the training images so that a precise and *aligned* localization of the parts can be obtained. For each part a k-means clustering on the feature space is effectuated, where k is the number of appearances that we want to model. Each cluster is then used as initialization for the appearance of the multi-appearance model.

The appearance compatibility parameters are initialized to zero so that initially, any appearance can be chosen. After this, their value is estimated by the SVM optimization so that compatible parts will obtain a positive weight while incompatible parts a negative one.

4. Implementation Details

We implement our model using HOG features for the object appearance and quadratic cost for the deformation features. Specifically, we define the features of an object part as:

$$\phi^I(\mathbf{x}, \mathbf{y} + \mathbf{z}) = H(\mathbf{x}, \mathbf{y} + \mathbf{z}) \tag{12}$$

where H is a function that given the image $\mathbf{x}$ extracts a vector of HOG features [2] from the given location $\mathbf{y} + \mathbf{z}$. The deformation features are defined as:

$$\phi^D(\mathbf{z}) = [z_x^2, z_y^2, z_x, z_y] \tag{13}$$

to account for the displacement magnitude and direction. Due to these choices, the maximisation in (1) can be done efficiently by using the distance transform [4]. Local Appearances are selected in the same maximisation, after applying its own displacement penalties.

For detection, the score $O(\mathbf{y}; \mathbf{x}, \mathbf{w})$ is evaluated at a discrete set of locations $\mathbf{y}$ which match to the layout of the underlying HOG features. To speed-up the evaluation of the appearance compatibility we produce an initial set of detection hypotheses without considering the pair-wise compatibility scores (5), rank them, and compute the full but more expensive score (8) only at the top 1000 candidates. This reduces the computational cost of the method without affecting the detection accuracy.

To get a final list of candidate detections, non-maxima suppression is run over the candidate list of bounding boxes of the different model aspects sorted by decreased confidence score. This procedure is greedy: after selection a new detection, any other detection that overlaps with it by more than a threshold is removed from the candidate pool.

The time required to detect an object is dominated by the number of part filters that need to be evaluated. For example, a model with two aspects, left-right flipping, and two appearances per part, requires $8 \times N_{\text{parts}}$ filtering operations. On a single core Xeon 2.4GHz a model with $N_{\text{parts}} = 9$, evaluating the cost on a PASCAL VOC image takes an average of ten seconds.

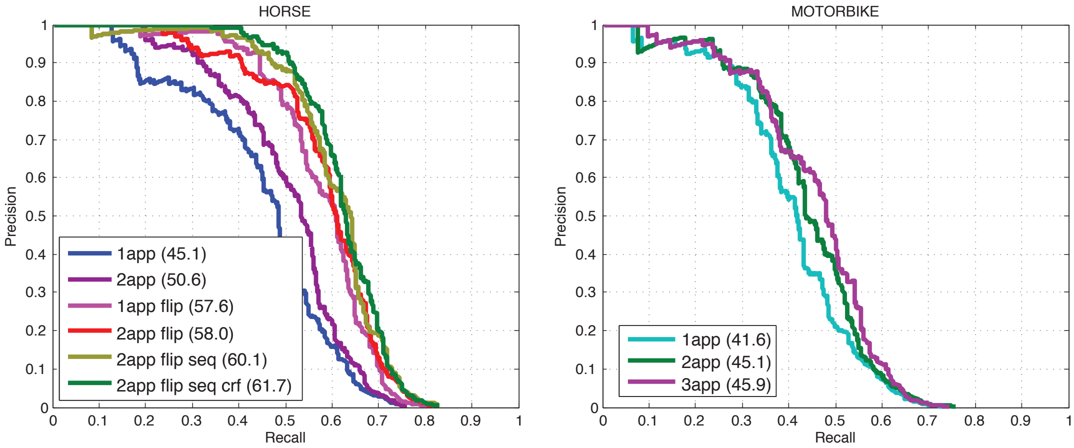
5. Experiments

We evaluate our approach on two standard datasets: INRIA Person Dataset [33] and Pascal VOC 2007 [34]. The variety of the classes helps to identify the classes where more benefit is obtained by the use of multiple local appearances. For evaluation, we use the comparison framework of [35] for INRIA, and the average precision (AP) with the standard Pascal VOC 2007 criterion.

5.1. Initialization

First, we evaluate the two initializations of the appearance explained in section 3 for the horse and motorbike classes. We begin with a model with 2 Components. Although the simplicity of the random initialization, the method is able to find two different appereances per part. As shown in Fig. 2 (i), a model of horse with 2 local appearances (named *2app*) with this random initialization gain 5 points over the 1 appearance model (*1app*).

Using the same initialization with the left-right models, the method gain is not as high as expected, and only improves in 1 point with respect to the flipped version. This is because the left-right orientation and the local appearances compete each other to estimate the same object appearance.



(i) (ii)

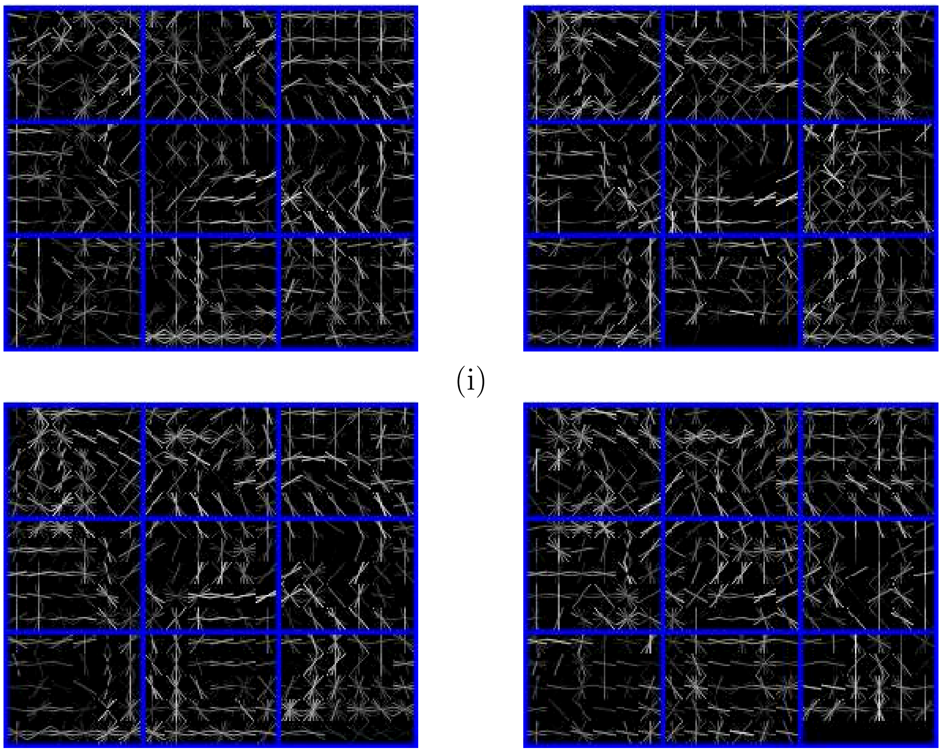
Figure 2: Average precision for the horse and motorbike classes. See explanation in the text.

An interesting example of this is shown in Fig. 3 (i), where it is illustrated the object model of a horse with random initialization. Local appearances and left-right model tries to represent the same appearance, finally resulting in impossible model configurations (i.e. horse with two heads in the top-right model). Instead, with the SEQ initialization, which sequentially learn the left-right prediction and then two latent estimations of local appearances, obtains a much nicer model. In this way, as the model orientation has already been learned, the local appearances can now learn different views of the object (namely, a quiet or a running horse).

This is shown in Fig. 3 (ii). We add the two appearances to the model, once the flip variables has been estimated, which represents the current state of the art for deformable HOG based models. Again, the multiple local appearances increase the performance, pushing the AP up to 60.1% which is already 4 points over the state of the art. Finally, learning the compatibility of the local appearances further increase the AP of more than 1 point reaching an AP of 61.7%. This is mainly due to less false detections are found, hence higher precision is achieved.

5.2. Inria Pedestrian Dataset

We next evaluate our approach on the INRIA Person Dataset [33]. For evaluation, we use the framework of [35].



(i)

(ii)

Figure 3: Effects of different multiple appearance initializations on the horse class. Left and right models represent the two appearances that each part can represent. In (i), all latent variables are estimated from the beginning. In (ii), local appearances are learnt sequentially, after an initial model has been learnt. Note that in (i) the top right horse has modelled two heads in the same model, and that in (ii) the horse is better modelled by its movements (quiet or in movement).

	1	2	3
Global Components	86.8%	86.7%	86.0%
Local Appearances	86.8%	87.8%	88.0%

Table 1: **AP on Pedestrian INRIA Database.** Comparison of the usage of multiple local or global appearances. Notice how overfitting kicks in when increasing the multiple components used, while this does not occur when adding more local appearances.

In Table 1, we evaluate different configurations of our model. The baseline is shown in column one and is a model with only one appearance per part

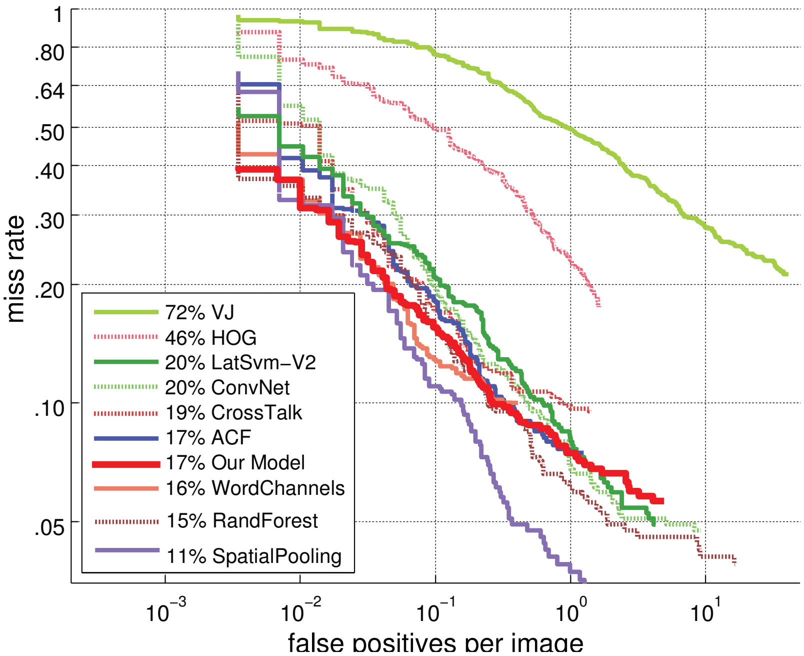


Figure 4: **Evaluation on INRIA.** Comparison with other state of the art methods on the Inria Person Dataset. Our method using only HOG features is on par with most of the other methods that use color and other more sophisticated features. For a complete explanation of the evaluation criteria and the methods see [35].

and one global component with left-right facing (like a traditional DPM). In the first row we show the effect of increasing the number components. It produces a slight decrement on AP, probably due to the statistical independence of each component. In practice increasing the global components reduces the number of samples available for each component and therefore the generalization capability of the learned model. In contrast, using more local appearances yields better accuracy, and the model reaches an AP of 88% when using a model with 3 appearances for each part.

In Fig. 4 we compare the model with 3 appearances with the current state of the art in pedestrian detection. As pedestrians assume different poses, local deformations are quite important. For this reason the DPM model (in the table is referred as LatSvm-V2) performs relatively well, even if other models use multiple and more expensive features like color or convolutional features. However, pedestrians can also wear different clothes or assume very

cmp	ap	crf	aero	bike	bird	boat	bott	bus	car	cat	chair	cow	tab	dog	hors	moto	pers	plant	sheep	sofa	train	tv	mean
2	2	Y	35.5	59.6	9.6	11.3	30.0	54.3	55.8	13.8	20.5	30.2	23.0	10.3	58.4	45.6	36.2	12.3	26.0	18.7	42.2	39.2	31.6
2	2	N	32.8	60.5	4.9	11.9	29.6	52.9	53.9	10.5	19.9	30.4	23.3	10.4	58.4	44.8	35.9	11.7	25.8	18.6	43.4	39.9	30.9
4	1	-	32.1	57.6	4.5	11.2	26.8	56.0	49.4	11.0	18.0	23.3	13.1	3.7	55.2	41.2	34.9	12.3	24.9	12.7	42.1	37.5	28.4

Table 2: Different configurations on PASCAL VOC 2007. First row reports AP values with our standard method with 2 components and 2 appearances per part. Second row report results for a model with exactly the same configuration but without using the appearance constraints introduced in section 2. The last row reports results for a model with 4 different components but a single appearance.

specific positions that cannot be explained with simple parts displacement. In this conditions the proposed model is better indicated. This is reflected on the evaluation, where our method combining deformation and a multimodal representation of the object parts clearly outperform DPM and is on par with most of the state of the art approaches which are specifically optimized for the task of pedestrian detection.

5.3. PASCAL VOC 2007

Our method is general enough to be used to learn any object class, not only pedestrians. In this sense we perform several experiments on the challenging VOC 2007 [34], where 20 different classes should be learned using the same settings.

We evaluate for each class the importance of the two main contributions of this work: (i) we compare local parts versus global components and (ii) we evaluate the effect of the pairwise constraints on the parts appearance. As reference we consider the AP of our model trained using 2 components and 2 local appearances as reported in the first row of table 2.

The second row of the table reports the AP for a new model trained with exactly the same configuration of our reference model but without using the pairwise appearance constraints. In average the model without appearance constraints is inferior to the complete one. This confirms our hypothesis that learning the pairwise compatibility between parts helps to improve the detection accuracy, see Fig. 5. Although the average difference between the two models is relatively small (0.7) for certain classes enforcing compatibility among parts can provide a neat improvement of more than 2 points.

In the third row of the table we train a new model with 4 global components. Doing so, the number of parameter to learn is similar to the model with 2 components and 2 appearances. However, in the 4 components model, each component is totally separate from the other. This can be considered

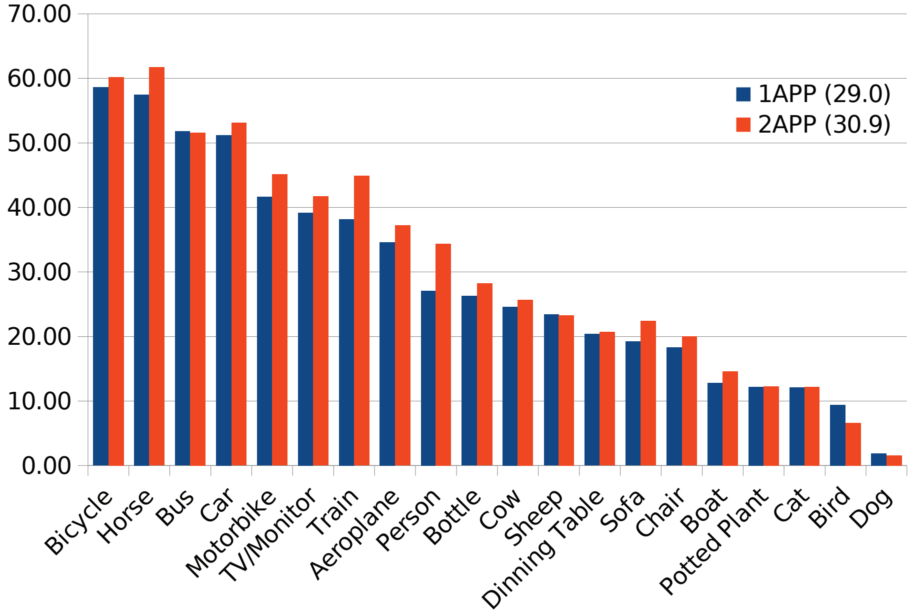


Figure 5: AP for different configurations of our model.

an advantage because the model avoids to mix-up appearances and it will probably generates fewer false positives. Still, the model cannot share parts which reduces its capability to generalize. In this experiment the advantage of using multiple local parts is evident. The 4 components model obtains a lower AP in almost every class and it has a mAP more than 3 points lower than the model with multiple appearances.

In Fig. 6 we visualize the occurrence of each possible configuration of parts for the class car. As the model is composed by 9 parts and each has 2 local appearances, a total of 2^9 different configurations can be expressed. This is much higher than the 4 representations of a traditional DPM with independent components. From the histogram we can see that there are few configurations that are the most used. However, most of the configurations are used at least one time. This shows that the model is really using its capability to combine different part appearances to represent the different instances of a class.

In table 3 we report the AP of the DPM model as reference and the AP of our deformable model with one, two and three appearances. Our baseline is around 2 points below the DPM score form [32]. This is mainly due to

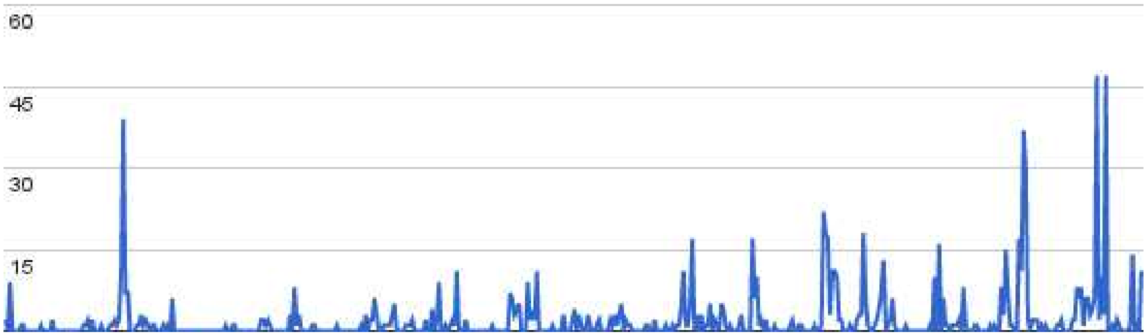


Figure 6: **Appearance distribution.** Each of the part configuration can be represented by 2^9 possible combinations. Here we show the number of times each configuration has been used to represent a car. Without the appearance factorization only two configurations can be selected.

	aero	bike	bird	boat	bott	bus	car	cat	chair	cow	tab	dog	hors	moto	pers	plant	sheep	sofa	train	tv	mean
DPM[32]	29.6	57.3	**10.1**	**17.1**	25.2	47.8	55.0	**18.4**	**21.6**	24.7	**23.3**	**11.2**	57.6	**46.5**	**42.1**	12.2	18.6	**31.9**	44.5	**40.9**	31.8
1 App	34.3	57.5	9.2	13.2	26.4	50.7	52.3	11.2	19.1	27.2	21.5	5.2	58.1	44.6	34	11.3	22.6	16.6	37.0	40.5	29.6
2 App	35.5	59.6	9.6	11.3	**30.0**	**54.3**	**55.8**	13.8	20.5	30.2	23.0	10.3	58.4	45.6	36.2	12.3	26.0	18.7	42.2	39.2	31.6
3 App	**37.6**	**61.9**	9.5	12.8	28.9	53.8	55.0	13.8	20.5	**31.0**	23.0	10.6	**60.4**	46.0	36.0	**13.0**	**27.1**	19.1	**44.6**	39.5	**32.2**

Table 3: PASCAL VOC 2007 Detection results. The first row reports results from [32] without bounding box estimation. Our model results are reported for 1 2 and 3 local appearances.

our implementation choice. A strategic placement of the parts can highly enhance the performance of the detector. However, as we use connections among nearby parts, we prefer to use a uniform distribution of the parts. Instead, in DPM they use a greedy procedure to find the best placement for the parts. Furthermore, the DPM is composed of a low-resolution and rigid model and on top of it several parts.

In our simplified model, we do not use the low resolution representation because we are mainly interested in the role of the parts and their interaction. Assuming that, we consider ours a strong baseline. Our baseline obtains a mAP of 29.6. The performance of the same model with 2 appearances per each part and pairwise compatibility constraints scores a mAP of 31.6 such that the gap with DPM is already almost cancelled.

Moving to 3 appearances per part leads to an additional improvement of 0.6 points as reported in the last row of the table. In Figs. 7 and 8, we show the different appearances learned for the parts of cars and horses,

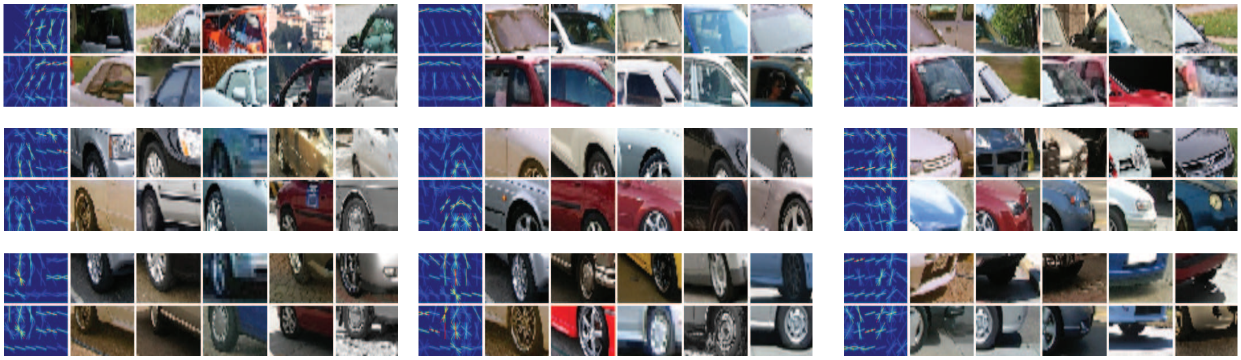


Figure 7: Top scoring detection for each appearance of each part of a car. Note that the two appearances are interchangeable.

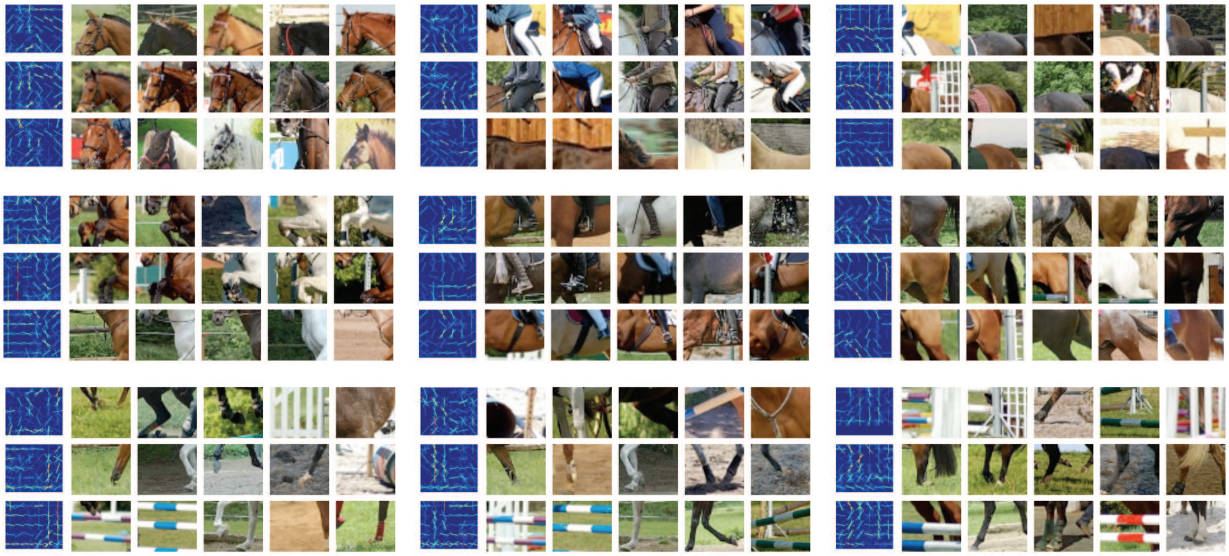


Figure 8: Top scoring detection for each appearance of each part the class horse. Note the differences that we capture in the appearance of the head and the legs, or in the rider.

respectively, together with the top 5 best scoring detections for each part. We can see how, despite describing the same object, each appearance learns a quite different model.

5.4. Discussion

In this paper we have shown how to increase the representational capability of DPM by adding multiple local part appearances that can be combined in a exponential number of possible representations with a limited computational cost. However, to obtain this model to work properly some important parts of the DPM algorithm had to be modified and improved.

First of all, as explained in Sect. (3.2), when dealing with multiple competing representations, especially at multiple levels, (as aspects and parts in our case), it is fundamental to apply a regularization that tends to keep the corresponding models balanced, so that one does not "steal" all samples. We notice that this problem becomes more and more important while increasing the number of appearances. In our setting we limit our experiments to 3 part appearances mostly for resources reasons (i.e. memory). However, we believe that further increasing the number of appearances can give a limited improvement also due to the competing representations problem.

Another very important factor for good results is initialization. In particular, clustering aligned parts can make a big difference in the final results (see Fig. 2(i) in Sect.5). Applying the clustering to fixed part locations (before alignment) would produce splits that represent the different displacement that the part can assume in different object instances. Therefore the resulting multi-appearance model of each part would represent almost the same appearance multiple times (would learnt displaced parts) which lead to a poor initialization, and the model would get also stuck. Instead, with our aligned initialization we assure that the split in the clustering would model different appearances of the same part. Even though the proposed initialization performs already much better than a naive one, we still believe that the initialization of the parts is one of the key points to further improve results. Specifically, for classes where the body deformation are relevant, like cats and dogs, a better initialization based on the real part location can produce much better results as shown in [19].

Finally, it is interesting to notice that, as the model capacity increases, for example in our case allowing combination of parts, the space of search of the negative examples also increases, which directly translates into a slower convergence. For example a training of a deformable model with 1 appearance needs an average of $4-5$ iterations of negatives to converge in the first iteration. If we move to 2 local appearances the number of iterations grows to $5-15$ while for 3 appearances it is necessary from 10 to 20 iterations. Despite the training time increases, during testing time, the method grows linearly with the number of appearances. In this sense methods like [26, 27, 36, 37, 38] can be used to reduce the computational cost for detection.

6. Conclusions

We have presented a new extension of the deformable parts model that can be used to learn multiple local appearances at a reasonable computational cost.

Compared to a traditional mixture of DPMs, our model (i) can express a very large set of different object appearances with a very small increase in the number of parameters, (ii) can learn the same amount of variation from far less training data by better exploiting the statistical dependencies between different object appearances, and (iii) is still very discriminative because the CRF constraints can reject unlikely part configurations.

Compared with multiple independent models, our approach can approximate an exponentially rich combination of appearances maintaining the same model representation. In addition, to limit our representation to only the feasible configuration of local parts, we introduce pairwise potential between appearances. We are also investigating the possibility to introduce the concept of occluded parts to the model as another local appearance, which can help on learning clearer parts.

Acknowledgments

Andrea Vedaldi was supported by the EU Project FP6 VIDI-Video IST-04554, ONR MURI N00014-07- 1-0182, and the Violette and Samuel Glasstone Research Fellowships in Science. The other authors would acknowledge the support of the Spanish projects TIN2009-14501-C02-02 and TIN2012-39051.

References

[1] Y. Yang, D. Ramanan, Articulated pose estimation with flexible mixtures-of-parts, in: CVPR, 2011.

[2] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, PAMI 32 (9).

[3] R. Girshick, P. Felzenszwalb, D. McAllester, Object detection with grammar models, in: NIPS, 2011.

[4] M. Fischler, R. Elschlager, The representation and matching of pictorial structures, IEEE Transactions on Computer 22 (1973) 67–92.

[5] P. Felzenszwalb, D. Huttenlocher, Pictorial structures for object recognition, IJCV 61 (1).

[6] A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, Multiple kernels for object detection, in: ICCV, 2009.

[7] S. Maji, A. Berg, J. Malik, Classification using intersection kernel support vector machines is efficient, in: CVPR, 2008.

[8] M. Pedersoli, A. Vedaldi, J. Gonzàlez, A coarse-to-fine approach for fast deformable object detection, in: CVPR, 2011.

[9] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, in: Proc. CVPR, 2010.

[10] V. Sreekanth, A. Vedaldi, A. Zisserman, C. V. Jawahar, Generalized RBF feature maps for efficient detection, in: BMVC, 2010.

[11] L. Zhu, Y. Chen, A. Yuille, W. Freeman, Latent hierarchical structural learning for object detection, in: CVPR, 2010.

[12] K. Zimmermann, D. Hurych, T. Svoboda, Non-rigid object detection with localinterleaved sequential alignment (LISA), PAMI 36 (4) (2014) 731–743.

[13] X. Zhang, Y.-H. Yang, Z. Han, H. Wang, C. Gao, Object class detection: A survey, ACM Comput. Surv. 46 (1).

[14] J. Zhang, K. Huang, Y. Yu, T. Tan, Boosted local structured hog-lbp for object localization, in: CVPR, 2011.

[15] F. Khan, R. M. Anwer, J. van de Weijer, A. Bagdanov, M. Vanrell, A. Lopez, Color attributes for object detection, in: CVPR, 2012.

[16] Y. Chen, L. Zhu, A. Yuille, Active mask hierarchies for object detection, in: ECCV, 2010.

[17] X. Zhu, C. Vondrick, D. Ramanan, C. C. Fowlkes, Do we need more training data or better models for object detection?, in: BMVC, 2012.

[18] P. Ott, M. Everingham, Shared parts for deformable part-based models, in: CVPR, 2011.

[19] H. Azizpour, I. Laptev, Object detection using strongly-supervised deformable part models, in: ECCV, 2012.

[20] X. Chen, R. Mottaghi, X. Liu, N.-G. Cho, S. Fidler, R. Urtasun, A. Yuille, Detect what you can: Detecting and representing objects using holistic models and body parts, in: CVPR, 2014.

[21] L. Zhu, Y. Chen, C. Lin, A. Yuille, Max margin learning of hierarchical configural deformable templates (HCDTs) for efficient object parsing and pose estimation, IJCV.

[22] B. Rothrock, S. Zhu, Human parsing using stochastic and-or grammar and rich appearance, in: ICCV Workshops (SIG), 2011.

[23] L. Bourdev, S. Maji, T. Brox, J. Malik, Detecting people using mutually consistent poselet activations, in: ECCV, 2010.

[24] P. Viola, J. Platt, C. Zhang, Multiple instance boosting for object detection, in: NIPS, 2005.

[25] A. Quattoni, M. Collins, T. Darrell, Conditional random fields for object recognition, in: In NIPS, 2004.

[26] H. Pirsiavash, D. Ramanan, Steerable part models, in: CVPR, 2012.

[27] H. Song, S. Zickler, T. Althoff, R. Girshick, M. Fritz, C. Geyer, P. Felzenszwalb, T. Darrell, Sparselet models for efficient multiclass object detection, in: ECCV, 2012.

[28] S. Fidler, M. Boben, A. Leonardis, A coarse-to-fine taxonomy of constellations for fast multi-class object detection, in: ECCV, 2010.

[29] V. Kolmogorov, Convergent tree-reweighted message passing for energy minimization, PAMI.

[30] Y. Singer, N. Srebro, Pegasos: Primal estimated sub-gradient solver for svm, in: ICML, 2007, pp. 807–814.

[31] A. Yuille, The concave-convex procedure, in: NIPS, 2002.

[32] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, Discriminatively trained deformable part models, release 4, http://people.cs.uchicago.edu/ pff/latent-release4/ (2010).

[33] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, 2005, pp. 886–893.

[34] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL visual obiect classes (VOC) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338.

[35] P. Dollár, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, Vol. 99, 2011.

[36] D. Levi, S. Silberstein, A. Bar-Hillel, Fast multiple-part based object detection using kd-ferns, in: CVPR, 2013.

[37] R. Benenson, M. Mathias, R. Timofte, L. Van Gool, Pedestrian detection at 100 frames per second, in: CVPR, 2012.

[38] X. Wang, M. Yang, S. Zhu, Y. Lin, Regionlets for generic object detection, in: ICCV, 2013.