Part-Pair Representation for Part Localization

Jiongxin Liu, Yinxiao Li, and Peter N. Belhumeur

Columbia University, USA
{liujx09,yli,belhumeur}@cs.columbia.edu

Abstract. In this paper, we propose a novel part-pair representation for part localization. In this representation, an object is treated as a collection of part pairs to model its shape and appearance. By changing the set of pairs to be used, we are able to impose either stronger or weaker geometric constraints on the part configuration. As for the appearance, we build pair detectors for each part pair, which model the appearance of an object at different levels of granularities. Our method of part localization exploits the part-pair representation, featuring the combination of non-parametric exemplars and parametric regression models. Non-parametric exemplars help generate reliable part hypotheses from very noisy pair detections. Then, the regression models are used to group the part hypotheses in a flexible way to predict the part locations. We evaluate our method extensively on the dataset CUB-200-2011 [32], where we achieve significant improvement over the state-of-the-art method on bird part localization. We also experiment with human pose estimation, where our method produces comparable results to existing works.

Keywords: part localization, part-pair representation, pose estimation.

1 Introduction

As a fundamental problem in computer vision, object part localization has been well studied in the last decade. Previous methods have been applied to different tasks, such as facial landmark detection [22,9,16,10,3,7], human pose estimation [26,19,34,24,25], and animal part localization [2,6,20,8]. In this paper, we use birds and humans as the test cases to design a unified framework for object detection and part localization, further improving the performance.

Existing works mainly focus on two directions: one is to build strong part detectors, and the second is to design expressive spatial models. To model the appearance of local parts that are variable and inherently ambiguous, mixture of components [34,36], and mid-level representations [24,25] are used. As for the spatial model, pictorial structure [18] and its variants have been proved to be very effective in different domains including human pose estimation. However, the pair-wise constraints in pictorial structure are sometimes not strong enough to combat detection noise, as shown in [20]. As a non-parametric spatial model, exemplar [3,35,20] has great success on the human face and birds. But as shown in Sec. 6.3, [20] does not work very well on the human pose, presumably due to

D. Fleet et al. (Eds.): ECCV 2014, Part II, LNCS 8690, pp. 456–471, 2014.

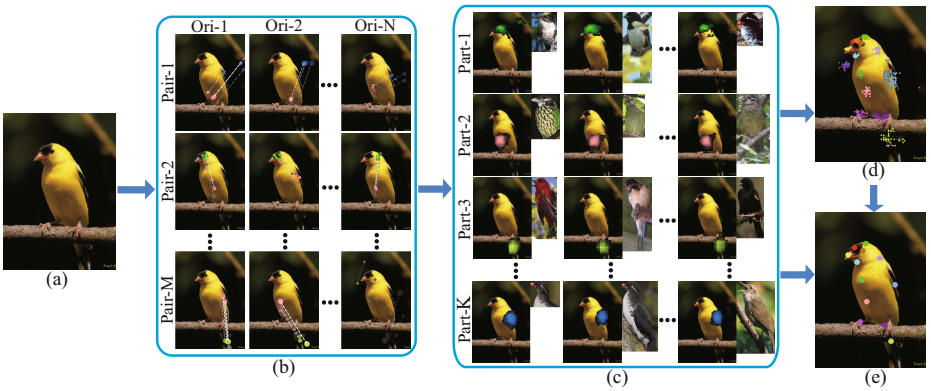


Fig. 1. Illustration of our method. (a) A testing image. (b) Pair detections for different pairs in different orientations. Each dashed line denotes a detected part pair. (c) Exemplar-conditioned part response maps. The exemplar is shown on the right side of the testing sample. (d) Candidate part detections. (e) Predicted part configuration. The brightness of colors indicates the confidence. Please see Fig. 6(a) for the color code

insufficient training exemplars and the limitations of the Consensus of Exemplars approach.

To better model the appearance and shape of an object, we propose part-pair representation where the object is represented as a set of part pairs. For the appearance, we build an ensemble of pair detectors, each of which targets a pair of parts. For the shape, we use the orientations and scales of the pairs to represent the global part configuration. The part-pair representation has two benefits. First, the ensemble of pair detectors cover overlapping regions on an object at different levels of granularities, thus capturing rich visual information. In spirit, pair detectors share some similarities with Poselet detectors [5]. But as pair detectors explicitly target pairs of parts, they are more suitable for our part-pair representation. Second, as the part-pair representation uses a complete graph to connect the parts, we have the luxury of adjusting the set of pairs to be considered when enforcing geometric constraints, making it either stronger (more rigid) or weaker (more flexible). Such flexibility is missing in the original exemplar-based models [3,20].

Using the part-pair representation, our method harnesses non-parametric exemplars and parametric models. As shown in [3,20], the exemplars help enforce relatively strong geometric constraints to suppress false part detections. But our instantiation of the idea is quite different, as we only expect to obtain an accurate estimation for a particular part, rather than for the global shape. After obtaining such part-centric hypotheses, the parametric regression models score pair-wise hypotheses to select the best ones to infer the global configuration. Such composition is flexible, as it only uses a subset of part predictions from each hypothesis that are likely to be correct.

In this paper, we use birds as the example to explain our method. Fig. 1 illustrates the pipeline of our method. Given a testing image, pair detectors scan the images over scales (Sec. 3). Exemplar-conditioned super part detectors are constructed, generating hypotheses for each part (Sec. 4). These hypotheses are then integrated to predict the global part configuration through parametric regression models (Sec. 5).

Our work makes the following contributions:

1. We propose a novel part-pair representation to model the rich visual and geometric features of an object.
2. We show how to apply the part-pair representation to localize individual parts using an exemplar-based framework. It generates reliable part hypotheses, facilitating the subsequent procedure.
3. We design a flexible strategy to integrate the part hypotheses.
4. Our method produces state-of-the-art results on bird part localization, as well as comparable results on human pose estimation.

2 Related Work

An important component in part localization is the appearance model, which has been studied in the context of object detection. Haar-like wavelets have been used in AdaBoost classifier [31] for human face detection. Subsequently, the paradigm of Linear SVMs trained on HOG features proved very popular [11,5,17]. A sufficiently fast non-linear detector which combines soft cascade with integral channel features is studied in [15,14,13]. Higher-level features have also shown promising results on object detection [27]. Recently, deep neural network has been applied to pedestrian detection and general object detection [23,30]. In our work, we follow [14,13] to build pair detectors that capture the appearance of geometrically rectified pair of parts. Note that our pair detectors differ from "pairs" of detectors in [28] which capture the mutual position of two independently learned part detectors.

Various shape models have been proposed in facial landmark detection. Statistical shape models [22,9,21] use multivariate Gaussian distribution to model the shape and appearance of a face. To better capture the shape and appearance variations, Constrained Local Models [3,1,35] constrain the global shape on top of local part detections, while tree-structured models jointly optimize the appearance and spatial terms to infer the part locations [18,16,36]. Shape regression [7] also works well on the human face, which is attributed to the strong correlation between low level features like pixel values and the shape increment.

As for human pose estimation, the tree-structured model has gained favor due to its generalization ability and efficiency. More importantly, the tree structure fits the kinematic skeleton of the human body, enabling effective modeling of the spatial relations. Starting from the work of [18], variants of the method have been developed [26,19,34,29,24,25]. To learn the model with large-scale datasets, a fast structured SVM solver is introduced in [6]. Recently, Poselet detectors are incorporated to capture additional mid-level information [33,24,25]. [33] designs

a complex hierarchical model by organizing the Poselets in a coarse-to-fine hierarchy. [24,25] extends the pictorial structure by using Poselet dependent unary and pairwise terms. Our pair detectors share some similarities with the original Poselet detectors [5], but as pair detectors explicitly target pairs of parts rather than a random set of multiple parts, they can be easily manipulated to predict the part locations under the part-pair representation.

However, the tree-structured model does not work very well in the case where the parts to be estimated do not follow a kinematic tree, and the object resides in a cluttered scene with unknown position and size, as shown in [20]. The reason is that the first-order spatial constraints from tree-structured models are not strong enough to combat noisy detections. [20] manages to impose stronger and more precise constraints through exemplar-based models. But the rigidity of the models and the requirement of sufficiently large number of training samples limit its efficacy in human pose estimation, which will be shown in Sec. 6.3.

To combine the merits of tree-structured models and exemplars, we propose a novel part-pair representation. Under such representation, we employ exemplars to generate high quality hypotheses for each part. Then we design parametric models that exploit part-pair scores to combine these hypotheses in an optimal way. Our method demonstrates good performance on two challenging datasets [32,19].

3 Part-Pair Representation

Unlike part-based models that treat an object as a collection of parts, the part-pair representation breaks down the object into part pairs, forming a complete graph connecting the parts. Under such representation, the shape and appearance modeling focuses on the pairs (*i.e.*, the edges of the graph).

3.1 Shape Modeling

Assuming an object X has n parts with x^i denoting the location of part i, then part-pair representation treats X as a set of $n(n-1)/2$ part pairs $\{(x^i, x^j)|i, j \in [1, n], i \neq j\}$. For each pair (i, j) of X, we record its center location c^{ij}, orientation θ^{ij}, and length l^{ij}. Ideally, as any set of $n-1$ pairs that span all the parts uniquely define the global part configuration, the other pairs seem to be redundant. In practice, such redundancy allow us to adjust the strength of the enforced geometric constraints by changing the set of pairs to be considered, which will be addressed in Sec. 4 and Sec. 5.

3.2 Appearance Modeling

We build pair detectors to model the appearance of each pair (pair detectors can be seen as specialized Poselet detectors [5], aiming at localizing two parts simultaneously). These detectors cover different regions on an object at different levels of granularities, with possibly significant overlap. For this reason, we have a rich representation of the object appearance.

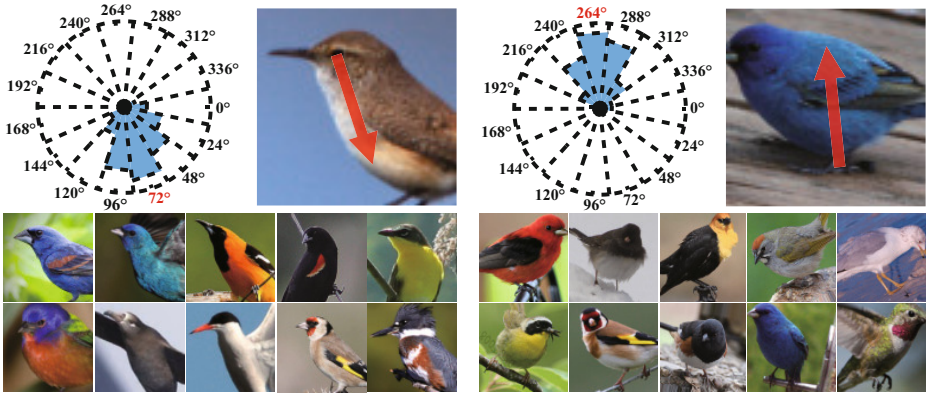


Fig. 2. Training samples after normalization. The left figure is for the pair (Left Eye, Belly), and the right figure corresponds to (Left Leg, Back). In each figure, sample frequencies over 15 orientations are visualized as blue sectors in the pie chart. The red arrow superimposed over the sample image of each figure indicates the pair orientation

Mixtures of Pair Detectors. To deal with rotation variations of the pairs, we discretize the rotation space in 15 different bins, corresponding to a span of 24 degrees. We then build one detector for each pair and each orientation[1]. For efficiency, we measure the sample frequencies in each orientation bin, and ignore the bins with frequencies smaller than 1%. So we have 776 rather than 1,575 detectors altogether for the bird dataset [32] where the number of parts is 15.

Inspired by POOF [4], we normalize the samples for each pair detector by rotating and rescaling the images, so that they are aligned at the two corresponding parts. Please see Fig. 2 for some aligned examples. For rotation, the rotation angle is determined based on the center of the target orientation bin. For rescaling, we rescale the samples of different pairs to different reference sizes, as they contain different granularities of information. For example, (Eye, Forehead) pair is typically much smaller than (Eye, Tail) pair in an image, resizing the (Eye, Tail) samples to a very small size may lose useful information.

To automate the process of deciding the reference sizes, we first estimate the average length $\bar{l}$ for each pair from the training data. After that, we know the minimum and maximum average lengths $\bar{l}_{min}$ and $\bar{l}_{max}$ among all the pairs. Assuming that the normalized length lies in the range $[\hat{l}_{min}, \hat{l}_{max}]$, we use a linear function $f(l)$ to map the range $[l_{min}, l_{max}]$ to $[\hat{l}_{min}, \hat{l}_{max}]$. Therefore, the reference size for pair (i, j) is $f(\bar{l}^{ij})$. We empirically set $\hat{l}_{min} = 24$ and $\hat{l}_{max} = 52$ to ensure reasonable image quality and avoid up-sampling the images too much.

[1] In our work, we use a single non-linear detector to handle pose & appearance variations within the same orientation bin. Alternatively, one can build multiple linear detectors (*e.g.*, Linear SVM + HOG) to explicitly decompose the visual complexity.

Training and Testing. After normalization, we use the toolbox [12] to extract the first-order integral channel features within an outer bounding box (*i.e.*, feature window) that contain both parts inside. Note that the feature window is placed at the center of the corresponding pair. We randomly generate up to $2,000$ rectangles to compute the features, and follow [13] to build a soft cascade detector with constant rejection thresholds. The details are as follows.

We build a cascade with $T = mT_0$ weak classifiers, and each weak classifier is a depth-two decision tree. $m = 30$ is the number of rounds of bootstrapping. After each round, we mine up to 400 hard negatives, and increase the number of weak classifiers by $T_0 = 50$ to build an AdaBoost classifier. Instead of performing a rejection test at every weak classifier, we check it after every T_0 weak classifiers (to accumulate enough observations). Assuming the score of a sample s at the kT_0-th weak classifier is $H_k(s) = \sum_{j \leq kT_0} \alpha_j h_j(s)$ where $\alpha_j > 0$ and $h_j(s)$ is the output of the j-th weak classifier, then the threshold is set as $\tau_k = b \sum_{j \leq kT_0} \alpha_j$ ($b = 0.45$ in our experiment).

At the testing stage, we build an image pyramid with 6 scales per octave, and apply the pair detectors in a sliding-window paradigm (with stride 4 pixels). To facilitate the following procedures, we normalize the scores so that an early rejected sample will not be penalized too much. To do this, we use $\bar{H}_k(s) = \frac{H_k(s)}{\sum_{j \leq kT_0} \alpha_j}$, and the normalized score $\bar{H}_k(s)$ is within the range $[0, 1]$ (early rejected samples will have scores below 0.45). Note that we do not apply Non-Maximum Suppression to the detection results; instead, we cache them as response maps at each scale.

4 Super Part Detector

Our method of part localization follows a bottom-up paradigm, and an important step is to generate reliable estimations for each part. In detection and localization tasks, the output from a single detector is usually very noisy. Therefore, additional contextual information such as the output from other related detectors is needed. Part-pair representation allows us to exploit such context, featuring the use of exemplar-based models. The motivation is that there are multiple pair detectors sharing the same part, and exemplars guide which pair detectors should be used (a way of imposing geometric constraint). In [3,20], the basic element of an exemplar is the part, and exemplars are used to dictate plausible global configuration of parts. In building super part detector, however, the basic element of an exemplar is the part pair, and exemplars provide an example of relevant pairs to a particular part (please see Fig. 3(a)).

4.1 Part Response Maps

Given the detection output from the pair detectors (in the form of pair response maps), our goal is to generate the response map for each part. The idea is similar to Hough Voting: a part pair activation votes for the positions of its two related parts. However, to gather the votes for a particular part i, exemplars are used

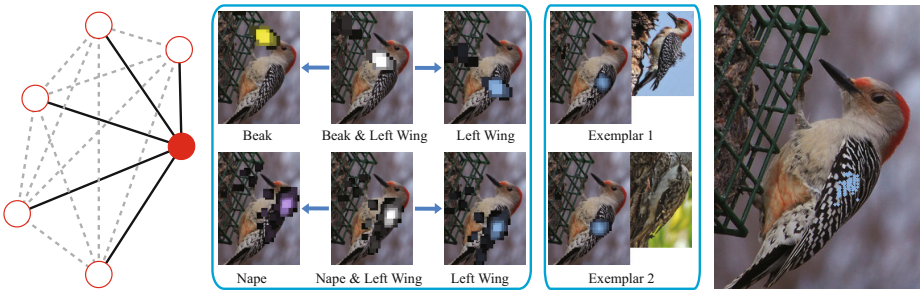


Fig. 3. (a) shows the part-pair representation as a complete graph for an object with 6 parts. To build the super part detector for a part (solid circle), only the pairs sharing the part are considered (solid lines). (b) illustrates the shifting of pair response maps. (c) shows the response maps for Left Wing conditioned on two exemplars. (d) shows the candidate detections for Left Wing.

to specify which pairs, orientations, and scales should be used. Assuming X_k is an exemplar being scaled to a particular size, we can obtain the response map for part i conditioned on X_k as follows.

Let $R^{ij}(x)$ denote the response maps for pair (i, j) where x is the pixel location in the testing image, then the exemplar X_k specifies the particular response map to use (at certain scale and orientation), which is denoted as $R^{ij}_k(x)$. To vote for part i based on the detections of pair (i, j), we can simply shift $R^{ij}_k(x)$ to obtain the corresponding map for part i, as illustrated in Fig. 3(b):

$$r^{ij}_k(x) = R^{ij}_k(x + o), \quad o = c^{ij}_k - x^i_k, \tag{1}$$

where o is the shifting offset, computed from $c^{i,j}_k$ – the center location of pair (i, j), and x^i_k – the location of part i. In our implementation, we quantize the offset o based on the discretization of pair rotations after normalizing the scales (please refer to Sec. 3.2). During testing, we pre-shift and cache the response maps using the quantized offset o. Therefore, given an exemplar X_k, we can easily retrieve its corresponding map $r^{ij}_k(x)$.

As exemplar X_k tells all the visible pairs (*i.e.*, both parts of a pair should be visible) sharing part i, the part response map for part i is then estimated as

$$R^i_k(x) = \frac{1}{N^i_k} \sum_j r^{ij}_k(x), \tag{2}$$

where N^i_k is the number of visible pairs containing part i. Assuming there is a detector that directly generates such response map, then we name it as *Super Part Detector*, which is conditioned on a particular exemplar. Fig. 3(c) shows two such maps from two exemplars.

4.2 Part Hypotheses

In this section, we will describe how to generate part hypotheses from the super part detectors. As described in Sec. 4.1, different exemplars give us different super part detectors for part i. However, only the detectors from exemplars that match the testing sample at the part are meaningful. By "match at the part", we mean that the exemplar has similar configuration of parts in the neighborhood of the target part; by "meaningful", we mean that the detector has reasonably high score at the correct part location rather than the background. Therefore, we simultaneously find the good exemplars and possible part locations.

A reasonable indicator about the goodness of an exemplar is the peak value of its corresponding part response map. So, for part i, score of X_k is

$$S_k^i = \max_x R_k^i(x). \tag{3}$$

To search for good exemplars, a naive way is to go through all the training exemplars, rescale them to each possible scale, evaluate their scores with Eq. 3 and keep the top-scoring ones. This process can be made faster using a heuristic strategy: we compute the upper bound of S_k^i with much lower cost, and obtain an initial set of promising exemplars. Then we use Eq. 3 to recompute their scores. The upper bound is computed as $\hat{S}_k^i = \frac{1}{N_k^i} \sum_j \max_x r_k^{ij}(x)$, where the addend can be reused to evaluate different exemplars. In our experiment, we keep the best 100 exemplars for part i, and extract up to five local maximas from each corresponding part response map. The locations and scores of these maximas form the candidate part detections as in Fig. 3(d).

We have a by-product from the above procedure. As the candidate part detections indicate where to place the exemplar in the image, we also obtain the predictions for the other parts. For instance, given X_k and a candidate detection of part i at x_0, the location of part j is $x_0 - x_k^i + x_k^j$ with confidence value $r_k^{ij}(x_0)$.

4.3 Discussion

The super part detector demonstrates one way of using part-pair representation, where a subset of up to $n-1$ pairs are used to impose the geometric constraints (please see Fig. 3(a)). Because multiple pair detections are accumulated, the super part detector is tolerable to the noise from certain pairs. Because of the discretization in the spatial domain, rotation space, and scale space, the super part detector is also tolerable to the displacement between the exemplar's parts and the testing sample's parts, especially for the distant parts with respect to the target part. For these reasons, the strength of geometric constraint from exemplars is weaker than that in [3,20]. In other words, exemplars that do not match the testing sample globally can still be useful in localizing a particular part.

5 Predicting Part Configuration

Recall that in Sec. 4.2, we obtain a set of hypotheses for each part. Each hypothesis consists of the candidate part detection, as well as the corresponding exemplar. Then we need to use the hypotheses to predict the global part configuration. We have two approaches, one is rigid and the other if more flexible.

5.1 Rigid Method

The idea is similar to [20]: assuming we place the exemplar X_k at a position in the testing image, then we evaluate the overall score of the exemplar as $S_k = \frac{1}{N_k}\sum_{i,j} R_k^{ij}(c_k^{ij})$, where N_k is the number of visible pairs.

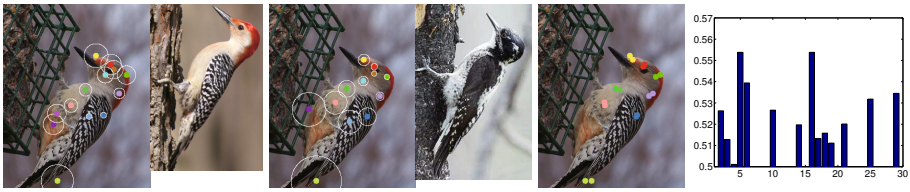
To predict the global configuration, we evaluate the overall scores for all the candidate exemplars (*i.e.*, the exemplars placed in the testing image at the corresponding candidate part locations). Once we obtain the best $N = 30$ exemplars, we follow [20] to predict the visibilities and locations of all the parts using Consensus of Exemplars (CoE). As can be seen here, the method is very rigid, expecting the exemplars to match the testing sample globally; also, the strength of geometric constraints is very strong, as all the pairs in the part-pair representation are used. Therefore, it may fail if good matches to the testing sample do not exist in the training data, which is likely to happen when we do not have a large set of representative training samples.

5.2 Flexible Integration

One limitation of the rigid method is that all the parts from a single exemplar are taken into consideration at the consensus stage, some of which are purely distractors. The simple non-linear consensus operation is likely to fail if such noise is above a certain level. In our flexible method, we attempt to filter out the noise in a more effective way.

To do this, we construct a number of groups of part hypotheses, with at most one hypothesis corresponding to a particular part in each group. We evaluate these groups, and use the best one to predict the global part configuration. As the top-scoring hypotheses already have very high accuracy as shown in Tab. 1, we only keep a few of them (15 in our experiment) for each part.

Grouping Hypotheses. Following the discussion in Sec. 4.3, we first define the UR (uncertainty region) for each part inherited from an exemplar in a part hypothesis. Assuming we have a hypothesis for part i, then the uncertainty region for part j is a circle with radius equal to a fraction (20% in our experiment) of the distance between part i and j. Given this definition, we claim that two part hypotheses agree on a particular part if its two corresponding URs are close enough to each other (based on the center distance divided by the larger radius). To control the strength of geometric constraint, we require that two hypotheses to be paired should agree on at least N parts including themselves, where N can be tuned. Fig. 4 shows two part hypotheses and the parts they agree on.



(a) (b) (c) (d)

Fig. 4. (a) shows a Back hypothesis, which predicts the locations of all the parts in the form of uncertainty regions marked by white circles. The corresponding exemplar is shown on the right. (b) shows a Crown hypothesis. (c) shows eight parts on which the two hypotheses in (a) and (b) agree. (d) shows the features of this pair of hypotheses

To make the problem tractable, we group the hypotheses in a pair-wise manner. To evaluate the goodness of two paired hypotheses for parts i and j, we design a feature vector with $2 \times n$ entries where n is the total number of object parts. The first n entries correspond to the scores of part pairs in the hypothesis of part i, with the k-th value to be the pair score of (i, k). The second n entries are formed in the same way. We zero out the entries for the parts the two hypotheses do not agree on. One example of such feature is shown in Fig. 4(d). Using a held-out validation set, we train a logistic regression model that map the features to the percentage of correctly predicted parts: if two hypotheses agree on m parts, and the ground-truth locations for those parts are close enough to the mean predictions, then the percentage is $\frac{m}{n}$. As the features carry semantic meaning, we build different regression models for different pairs of parts.

The pair-wise grouping procedure is as follows. Initially, each part hypothesis forms a group. Starting from the initial group, we sequentially add another hypothesis that can be paired with an existing hypothesis in the group. We keep track of the parts the newly paired hypotheses agree on (by marking them as detected), and subsequent pairing should have new parts detected. The procedure terminates when there is no more hypothesis to add. In the end, we obtain a number of groups. The score of each group is computed as $S = \frac{1}{M} \sum_p \alpha_p s(p)$, where p is the paired hypotheses, $s(p)$ is the output of the regression model, M is the total number of detected parts and α_p is the percentage of newly detected parts from the parts the paired hypotheses agree on.

Predicting the Part Locations. Given the highest-scoring group, we directly use the candidate part detections from the hypotheses as the final results; for each of the other detected parts, we use the mean prediction from the corresponding paired hypotheses; for each of the undetected parts, if there are hypotheses in this group having a related part pair with scores above 0.5, then we use the predicted location with the highest pair score; Otherwise, the part is marked as invisible. If there does not exist group with more than one hypothesis (which is unlikely to happen), we use the part predictions from the exemplar in the best

part hypothesis. As can be seen here, the parameter N controls the strength of geometric constraints. Smaller N indicates weaker constraints as it allows more dissimilar exemplars to contribute by only using their promising part predictions. In our experiment, we find $N = 5$ gives the best result.

6 Experiments

We evaluate our part localization method on the bird dataset CUB-200-2011 [32] and the human pose dataset LSP (Leeds Sports Poses) [19]. For all the experiments, we use the train/test split provided by the dataset. We withhold 15% of the training data as the validation set.

To evaluate the localization performance, we mainly use the PCP measure (Percentage of Correct Parts). For bird part localization, a correct part estimation should be within 1.5 standard deviation of an MTurk workers click from the ground truth part location. For human pose estimation, correct part should have end points within 50% of the part length from the ground truth end points.

6.1 Performance of Super Part Detector

To have an idea about the importance of the super part detector in our method, we evaluate its performance in localizing a particular part, and compare it with that of regular part detector and our pair detector. In the experiment, we do not try to reach an optimal solution for all the parts jointly. Instead, we predict the location of a single part, assuming it's visible.

For the regular part detector, we use the pose detectors designed by [20], where there are 200 detectors for each part. At the testing stage, the best five activations across all the pose detectors are outputted. As for the pair detector, recall that the activation of a pair detector casts a vote for its related parts. As such, to localize a part, we run all the relevant pair detectors (up to $14 \times 15 = 210$ detectors), and collect the highest-scoring predictions. For the super part detector, we use the best five candidate part detections obtained in Sec. 4.2. Note that we do not use Non-Maximum Suppression for all the detectors, and the activations are just local maximas in the response maps.

The PCPs for each part as well as the total PCP are listed in Tab. 1. We also report the top-5 accuracy, where at least one of the best five predictions is correct. From the comparison between pair detectors and pose detectors, we can see that using the different features and classifiers from [20] does not give us much better raw detectors. However, after building the super part detector from the pair detectors, we achieve significant improvement. This is reasonable as the super part detectors are context-aware. What we want to emphasize is that by imposing geometric constraints at an early stage, we have high quality part hypotheses which make it promising to design effective integration method.

Table 1. Comparison of different detectors in localizing individual parts. The super part detectors produce very reliable part activations. From left to right, the parts are: Back, Beak, Belly, Breast, Crown, Forehead, Eye, Leg, Wing, Nape, Tail, and Throat.

PCP	Ba	Bk	Be	Br	Cr	Fh	Ey	Le	Wi	Na	Ta	Th	Total
Part	23.4	23.8	31.1	28.1	35.0	28.8	11.5	17.3	18.3	29.4	10.0	34.7	23.9
Pair	27.2	28.4	39.7	31.8	21.4	28.4	5.3	14.5	13.2	38.6	17.9	44.7	25.1
SupP	**62.2**	**57.3**	**66.4**	**61.4**	**74.2**	**65.6**	**40.1**	**40.9**	**53.5**	**66.9**	**34.9**	**71.5**	**57.1**
Part-top5	49.1	47.2	56.2	55.7	62.3	51.1	23.9	37.9	43.9	53.5	26.6	59.1	46.7
Pair-top5	50.1	54.0	66.1	57.0	44.5	49.4	15.2	29.6	31.8	64.7	37.0	68.7	46.1
SupP-top5	**76.9**	**75.8**	**79.8**	**77.1**	**86.3**	**81.7**	**66.0**	**56.1**	**66.9**	**81.4**	**48.3**	**83.8**	**72.5**

Table 2. Comparison of part localization results on CUB-200-2011 [32]. Our method outperforms state-of-the-art techniques on all the parts.

PCP	Ba	Bk	Be	Br	Cr	Fh	Ey	Le	Wi	Na	Ta	Th	Total
DPM [6]	34.6	26.0	42.0	37.0	47.9	28.7	48.2	-	55.0	41.8	22.4	42.4	40.7
CoE [20]	62.1	49.0	69.0	67.0	72.9	58.5	55.7	40.7	71.6	70.8	40.2	70.8	59.7
Ours-rigid	59.7	59.0	69.5	67.3	**77.1**	**72.2**	67.9	39.9	69.7	75.2	34.7	76.7	63.1
Ours-flex	**64.5**	**61.2**	**71.7**	**70.5**	76.8	72.0	**70.0**	**45.0**	**74.4**	**79.3**	**46.2**	**80.0**	**66.7**

6.2 Predicting the Part Configuration

We evaluate our rigid and flexible methods (*i.e.*, Ours-rigid and Ours-flex) on predicting the global part configuration, including the visibilities. We compare with DPM implemented by [6] and exemplar-based method [20].

Tab. 2 shows the comparisons. DPM [6] has much lower accuracy possibly for two reasons: there is very large intra-class variability to be captured by few DPM components (14 detectors per part); the first-order spatial constraints in DPM are not strong enough to combat the detection noise. Although Ours-rigid does not outperform CoE [20] by a large margin, the improvement is still remarkable. First, we do not use subcategory labels; Second, the pair detectors does not have better performance than pose detectors as shown in Tab. 1. We attribute such improvement to the aggregation of a much richer set of appearance models that largely suppresses the false detections from individual detectors. Ours-flex further improves the overall PCP over Ours-rigid by about 3.6%. It clearly shows the benefit of adding some flexibility to the estimation of global configuration on top of the part hypotheses. Fig. 5 shows similar comparisons. As exemplars usually sacrifice Tail to match other parts, the improvement of Ours-flex over Ours-rigid on Tail is very large.

Fig. 6(a) shows some qualitative results. We can see that Ours-rigid fails to accurately localize the parts with large deformation. Because the constraints in the rigid method strongly restrict the prediction of part configuration, the estimations from Ours-rigid respect the exemplars much more than the particular

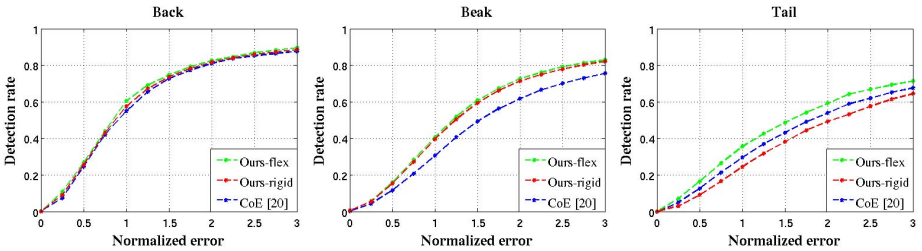


Fig. 5. Detection rates of Back, Beak, and Tail given varying degrees of localization precision. 1.5 is the threshold for a correct detection

Table 3. Comparison of part localization results on LSP dataset [19]. Our flexible method generates comparable results to the state-of-the-art works.

PCP	Torso	Upper leg	Lower leg	Upper arm	Forearm	Head	Total
Strong-PS [25]	**88.7**	**78.8**	**73.4**	**61.5**	**44.9**	**85.6**	**69.2**
Poselet-PS [24]	87.5	75.7	68.0	54.2	33.9	78.1	62.9
CoE [20]	83.4	69.0	61.7	47.5	28.1	79.3	57.5
Ours-rigid	84.2	69.3	61.5	48.7	28.5	79.9	58.0
Ours-flex	87.6	76.4	69.7	55.4	37.6	82.0	64.8

testing image, which is problematic when the exemplars do not match the testing sample well. Our flexible method mitigates this issue by allowing more flexible composition of part hypotheses.

Similar to [20], we conduct the experiment of species classification using the localized parts from our method. On the 200-species dataset, the mAP (mean average precision) is 48.32%; on the 14-species subset, the mAP is 65.18%.

6.3 Human Pose Estimation

We also apply our method to human pose estimation using LSP dataset [19]. Similar to [25], we use observer-centric (OC) annotations. The pair detectors are trained in the same way as those for bird dataset, and altogether we have 796 pair detectors. We also implement [20] with only pose consistency on this dataset.

The quantitative results are listed in Tab. 3. [20] and Ours-rigid do not work well on human pose estimation. Compared with the bird dataset, the number of training samples is much smaller in LSP, and human body is generally more articulated. These factors make the Consensus of Exemplar framework less effective in this experiment. Also note that the rigid method has only marginal improvement over [20]. One possible reason is that the images in the LSP dataset have already been rescaled and cropped (unlike [32]), making the effect of better suppressing false detections not prominent.

Tab. 3 also shows that Ours-flex method significantly improves over Ours-rigid. It also outperforms one state-of-the-art technique [24]. Compared with the

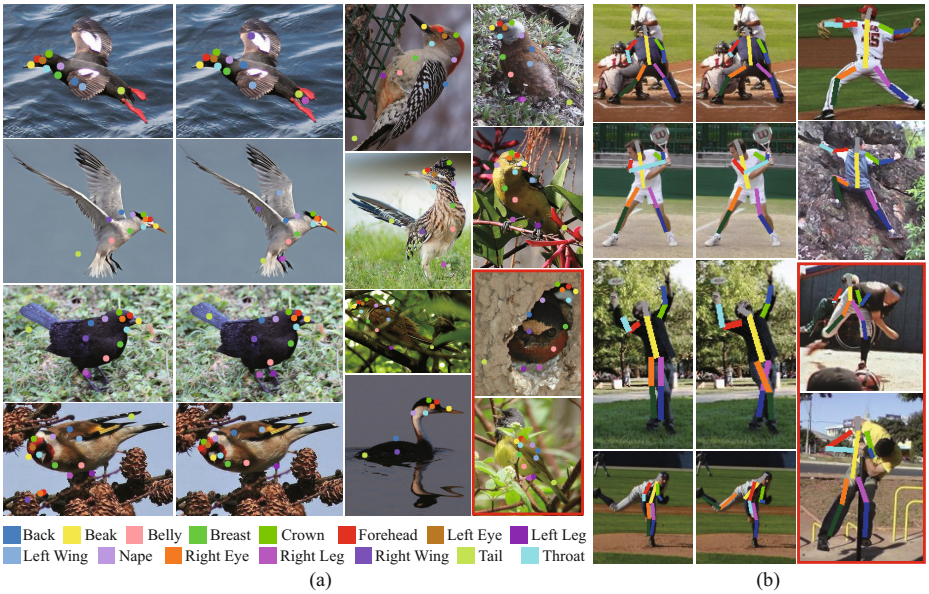


Fig. 6. (a) Qualitative results on CUB-200-2011 [32]. The color codes of the bird parts are at the bottom. (b) Qualitative results on LSP [19]. In both subfigures, the first two columns compare Ours-rigid (left) with Ours-flex (right), the other columns show more examples from Ours-flex. Failures are denoted with red frames.

well-constructed method [25], which employs many strong appearance models (some are tailored to human body), our method produces comparable results. The experiment demonstrates that our part-pair representation can be applied to the categories with large articulated deformation.

Some qualitative results are shown in Fig. 6(b). Similar to the comparison in Fig. 6(a), Ours-flex achieves more accurate localization by balancing the shape prior from exemplars and the detector activations in the testing image.

7 Conclusions

In this paper, we propose a part-pair representation to model an object, and study its application to part localization. Such representation enables us to capture rich visual information of the object, and impose adjustable geometric constraints on the part configuration. By combining part-pair representation with exemplars, we construct very powerful super part detectors, generating reliable part hypotheses. We also show that adding flexibility to the integration of part hypotheses largely improve the performance. Our method produces state-of-the-art results on bird part localization and promising results on human pose estimation.

References

1. Amberg, B., Vetters, T.: Optimal landmark detection using shape models and branch and bound. In: Proc. ICCV (2011)
2. Azizpour, H., Laptev, I.: Object detection using strongly-supervised deformable part models. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 836–849. Springer, Heidelberg (2012)
3. Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.: Localizing parts of faces using a consensus of exemplars. In: Proc. CVPR (2011)
4. Berg, T., Belhumeur, P.N.: POOF: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In: Proc. CVPR (2013)
5. Bourdev, L., Maji, S., Brox, T., Malik, J.: Detecting people using mutually consistent poselet activations. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 168–181. Springer, Heidelberg (2010)
6. Branson, S., Beijbom, O., Belongie, S.: Efficient large-scale structured learning. In: Proc. CVPR (2013)
7. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. In: Proc. CVPR (2012)
8. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: Proc. CVPR (2014)
9. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE TPAMI (2001)
10. Cristinacce, D., Cootes, T.: Feature detection and tracking with constrained local models. In: Proc. BMVC (2006)
11. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR (2005)
12. Dollár, P.: Piotr's Image and Video Matlab Toolbox (PMT), `http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`
13. Dollár, P., Appel, R., Kienzle, W.: Crosstalk cascades for frame-rate pedestrian detection. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 645–659. Springer, Heidelberg (2012)
14. Dollár, P., Belongie, S., Perona, P.: The fastest pedestrian detector in the west. In: Proc. BMVC (2010)
15. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: Proc. BMVC (2009)
16. Everingham, M., Sivic, J., Zisserman, A.: "Hello! my name is... buffy" automatic naming of characters in tv video. In: Proc. BMVC (2006)
17. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. In: IEEE TPAMI (2010)
18. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61(1), 55–79 (2005)
19. Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: Proc. BMVC (2010)
20. Liu, J., Belhumeur, P.N.: Bird part localization using exemplar-based models with enforced pose and subcategory consistency. In: Proc. ICCV (2013)
21. Matthews, I., Baker, S.: Active appearance models revisited. In: IJCV (2004)
22. Milborrow, S., Nicolls, F.: Locating facial features with an extended active shape model. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 504–513. Springer, Heidelberg (2008)

23. Ouyang, W., Wang, X.: Joint deep learning for pedestrian detection. In: Proc. ICCV (2013)
24. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: Proc. CVPR (2013)
25. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Strong appearance and expressive spatial models for human pose estimation. In: Proc. ICCV (2013)
26. Ramanan, D.: Learning to parse images of articulated bodies. In: Proc. NIPS (2006)
27. Ren, X., Ramanan, D.: Histograms of sparse codes for object detection. In: Proc. CVPR (2013)
28. Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 73–86. Springer, Heidelberg (2012)
29. Sun, M., Savarese, S.: Articulated part-based model for joint object detection and pose estimation. In: Proc. ICCV (2011)
30. Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Proc. NIPS (2013)
31. Viola, P., Jones, M.: Robust real-time object detection. IJCV 57(2), 137–154 (2001)
32. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Computation & Neural Systems Technical Report, CNS-TR-2011-001 (2011)
33. Wang, Y., Tran, D., Liao, Z.: Learning hierarchical poselets for human parsing. In: Proc. CVPR (2011)
34. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: Proc. CVPR (2011)
35. Zhou, F., Brandt, J., Lin, Z.: Exemplar-based graph matching for robust facial landmark localization. In: Proc. ICCV (2013)
36. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Proc. CVPR (2012)