Part Discovery from Partial Correspondence

Subhransu Maji Gregory Shakhnarovich
Toyota Technological Institute at Chicago, IL, USA

Abstract

We study the problem of part discovery when partial correspondence between instances of a category are available. For visual categories that exhibit high diversity in structure such as buildings, our approach can be used to discover parts that are hard to name, but can be easily expressed as a correspondence between pairs of images. Parts naturally emerge from point-wise landmark matches across many instances within a category. We propose a learning framework for automatic discovery of parts in such weakly supervised settings, and show the utility of the rich part library learned in this way for three tasks: object detection, category-specific saliency estimation, and fine-grained image parsing.

1. Introduction

Many visual categories have inherent structure: body parts of animals, architectural elements in a building, components of mechanical devices, etc. Discovery of such structure may be highly useful in design of recognition algorithms applicable on a large scale. Prior work in computer vision has addressed this task as either unsupervised, or highly supervised, with painstakingly annotated image sets. In this paper, we study the problem of discovering such structure with only a weak form of supervision: *partial correspondence* between pairs of instances within an object category.

Notion of parts is important to computer vision because much of recent work on visual recognition relies on the idea of representing a category as a composition of smaller fragments (or parts) arranged in variety of layouts. The parts act as *diagnostic* elements for the category; their presence and arrangement provides rich information regarding the presence and location of the object, its pose, size and fine-grained properties, e.g., architectural style of a building or type of a car.

Structure discovery may be especially important for categories of man-made objects, such as buildings, furniture, boats or aeroplanes. Examples of such categories are shown in Figure 1. Presence or absence of parts, or the number of their appearances, varies across instances; e.g., a church building may or may not have a spire, an airplane may have four, two or no visible engines. Furthermore, instances of these parts could differ drastically in their appearance, e.g., shape of windows for buildings, form of armrests for chairs. Still, despite this structural flexibility and appearance variability, humans can reliably recognize corresponding points across instances, even when the observer does not have a name for the part and does not precisely know its function.

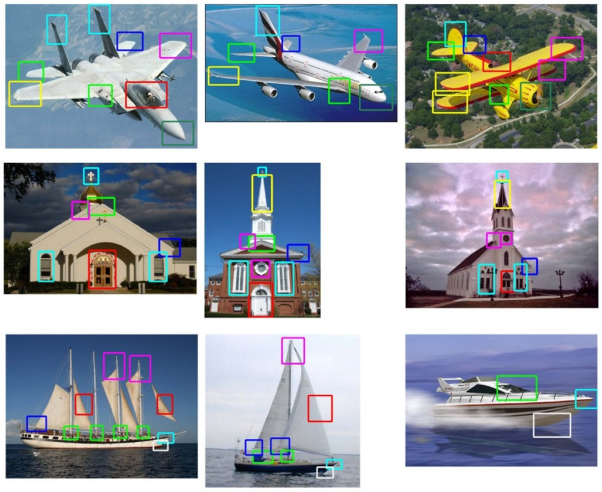
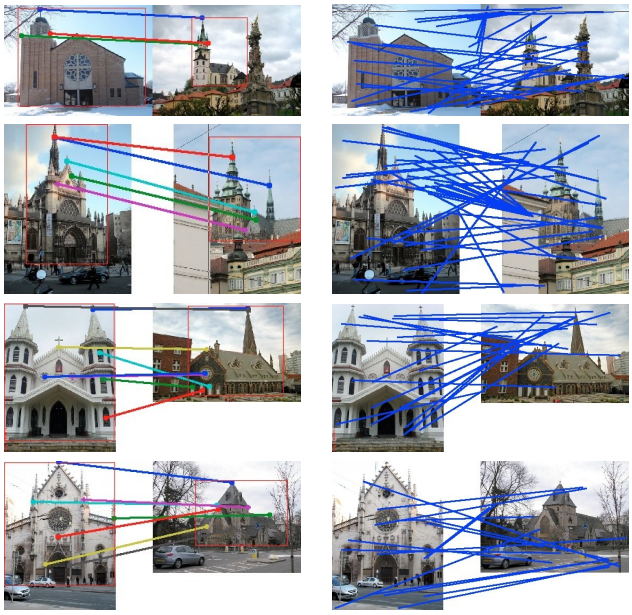


Figure 1. Objects and their diagnostic parts.

We leverage this ability through a recently introduced annotation paradigm that relies on people marking such correspondences, and propose a novel approach to construction of a library of parts driven by such annotations. Such annotations can enable discovery of parts that are aligned to human-semantics for categories that are otherwise hard to annotate using traditional methods of named keypoints, and part bounding boxes. We show the utility of the rich part library learned in this way for three tasks: object detection, category-specific saliency estimation, and fine-grained image parsing.

1.1. Prior Work

Most modern object detection methods rely on the notion of parts. These approaches differ on two important axes:



Semantic correspondence SIFT correspondence

Figure 2. Example annotations collected on Amazon's Mechanical Turk (*left*), which are much more semantic in nature than matches obtained using SIFT descriptors (*right*).

complexity of part "library", and the level of supervision in part construction or discovery. Poselet models [1, 2] rely on highly supervised annotations in which a set of 10 to 20 keypoints per category, defined by the designer of the annotation task, are marked. A large library of parts (poselets) is then formed by finding repeatable and detectable configurations of these keypoints.

In contrast, in many models the parts are learned automatically. This idea goes back to constellation models [22, 21] where the parts were learned via clustering of patches. More recent work has exploited the idea of pictorial structures [9] and deformable part models [8]. In such models parts are learned as a byproduct of optimizing the discriminative objective, involving reasoning about part appearance as well as their joint location relative to the object. Such approaches are usually limited to a handful of parts per model; in contrast we learn much larger libraries of parts.

A very different approach is taken in [19, 4], where parts are learned and selected in an iterative framework, with the objective to optimize specificity/sensitivity tradeoff. Our work differs in its use of the correspondence annotations, used very efficiently via semantic graph defined in the next section. This is a much more efficient strategy for constraining the search space.

The idea of using pairwise correspondences as source of learning parts was introduced in [15], along with an intuitive interface for collecting such correspondences. However, in [15] parts were learned in a rather naïve fashion, and no framework for selecting the parts was proposed, nor was the utility of the learned parts demonstrated on any task. In this paper we show how we can leverage the pairwise correspondence data to multiple fundamental tasks.

One can contrast this approach to using correspondences between detected "interest points". Methods that rely on such interest points use them as a kind of parts, and compare them with either universal descriptors like SIFT, or descriptors learned for the task. Examples of such approach include [17, 14, 20]. The latter work describes learning correspondence between patches that described the same element (part) of an urban scene. In contrast to our work, the points are detected using interest point operator, and the training relies on 3D correspondences obtained from structure from motion. As we show in Section 5, using generic interest point operators is inferior to using category-specific parts learned using our proposed approach.

Finally, a relevant body of work [5, 18, 7] addresses learning a good set of parts or attributes–which are often parts in disguise. The focus there is usually either on unsupervised learning, or on learning nameable parts; our work, in contrast, occupies the middle ground in which we rely on semantic meaning of parts perceived by humans without forcing a potentially contrived nameable nomenclature.

In the sections below we describe the procedure for learning a basic library of parts for a category using pairwise correspondences, and then proceed with a description of applying the part library to three tasks: object detection, landmark prediction and fine-grained image parsing.

2. From partial correspondence to parts

In this section we describe the framework for learning a library of parts using the correspondence annotations. We describe the annotation framework which was used to collect annotations in Section 2.1; how these can be used to define a "semantic graph" between images that enables part discovery in Section 2.2; the discriminative learning framework to learn part appearance models in Section 2.3.

2.1. Obtaining correspondence annotations

Following [15] we obtain correspondence annotations by presenting subjects with pairs of images, and asking them to click on pairs of matching points in the two instances of the category. People were recruited to annotate images on Amazon Mechanical Turk. They were given concise instructions, asking them to annotate "landmarks", defined as "any interesting feature of a church building". They were given a few visual examples, along with an emphatic clarification that these are not exhaustive; no further instructions were provided. Then, each person was presented with a sequence of image pairs, each containing a prominent church building. They can click on any number of *landmark pairs* that they deem corresponding between the two images.

Using this interface, we have collected annotations for 1000 pairs among 288 images of church buildings downloaded from **Flickr**. Landmark pairs, a few examples of which are shown in Figure 2 (*left*), include a variety of semantic matches: identical structural elements of buildings (windows, spires, corners, and gables), and vaguely defined yet consistent matches, the likes of "the mid-point of roof slope". Compare these to matches obtained using SIFT features as seen in Figure 2 (*right*).

2.2. Semantic graph of correspondence

Figure 3 illustrates how landmark correspondences between instances can be used to estimate the corresponding bounding boxes of parts in the two images. We estimate the similarity transform (translation and scaling) that maps the landmarks within the box from one image to another. If there are less than two landmarks within the box we set the scale as the relative scale of the two objects (determined by the bounding box of the entire set of landmarks in each image). The correspondence can be propagated beyond explicitly clicked landmark pairs using the *semantic graph* [15]. In this graph, there is an edge connecting every pair of landmarks identified as matching in the annotation. In this way, we can "trace" a part along a path in semantic graph from an image in left column to an image in the right column, even though we do not have explicit annotation for that pair of images.

Figure 4 shows various parts found from the source image by propagating the correspondence in the semantic graph in a breadth first manner. There are multiple ways to reach the same image by traversing different intermediate images and landmark pairs and we maintain a set of non-overlapping windows for each image. Doing so enables us to find multiple occurrences of a part in an object.

2.3. Learning a library of parts

The main idea of our algorithm for learning parts is to start with a possible single example of a part sampled from the data, augment it by examples harvested over the semantic graph, and construct a robust appearance model for the part that can explain sufficient fraction of these examples.

Sampling seed windows. We sample parts around the clicked landmarks in each image. The landmarks represent parts of the whole that are partially matched across instances. However the scale of the part around each landmark is unknown. We sample a large number of "seed windows" centered at each landmark, uniformly at random. The uniform sampling respects the underlying frequency of each part, i.e., parts that are matched frequently across each image are likely to be sampled frequently.

It is useful to contrast this method to alternative methods for sampling seeds. Sampling uniformly over image pixels would clearly be wasteful. Sampling using responses of a generic interest point operator such as Harris corner or DoG operator [14] might seem like a plausible alternative. However, as we show in Section 5, it is inferior to the landmark-driven saliency.

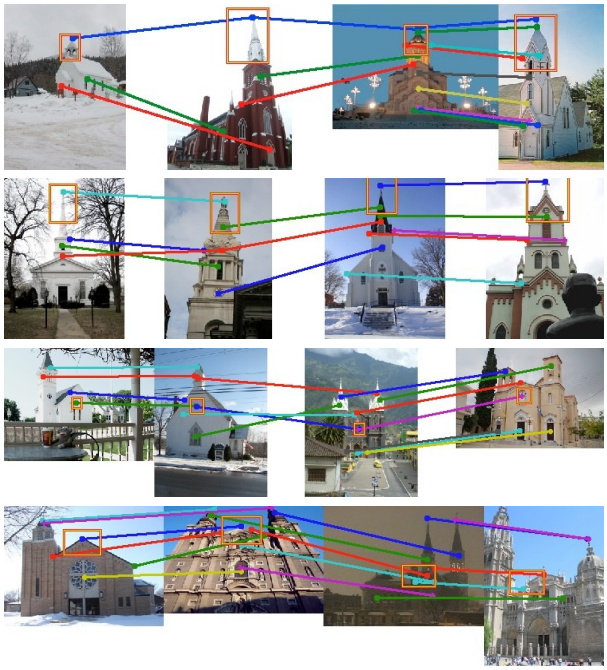


Figure 3. Correspondence propagation in the semantic graph from the image on the left to the image on the right in each row.

Learning an appearance model. We use HOG features [3] to model part appearance. Given a sampled "seed", we initialize the model by training the HOG filter $\mathbf{w}^{(0)}$ to separate the seed patch from a set of background patches; this step resembles the exemplar-SVM of [16]. Next, we propagate the correspondence from the seed window using breadth-first search in the semantic graph as shown in Figure 4. This provides a set of hypothesized locations for the part in other images. We denote them $\mathbf{x}(I_i, L_i^{(0)}, s_i^{(0)})$, for $i = 1, \ldots, k$, where $\mathbf{x}(I, L, s)$ is the patch extracted from image I at location L and with scale s; these locations and scales are estimated as explained in Section 2.2 and shown in Figure 3. We would like to use these additional likely examples of the part to retrain the model.

Since the correspondence is sparse, the estimated location and scale of these initial hypothesized matches is likely noisy. Furthermore, some of these matches may belong to a different visual sub-type of the part, e.g., a different kind of window or door. Therefore we treat the unknown location and scale of the matches as latent variables, and train the model using the following iterative algorithm.

In iteration t, we find for each hypothesized match the

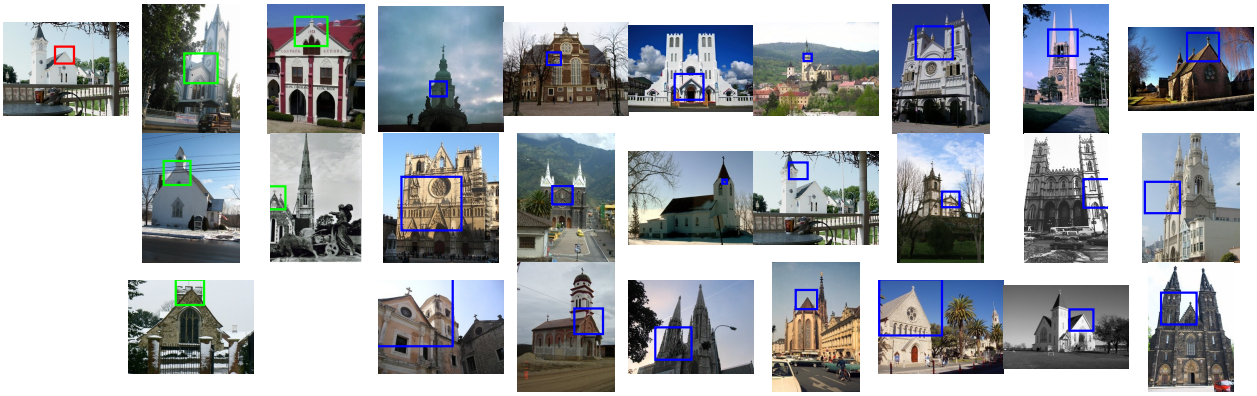


Figure 4. Breadth-first correspondence propagation in the semantic graph. The source part in shown in red in the leftmost figure. Parts found at depth one and two are shown in green and blue respectively.

location and scale *near* the initial estimate obtained using the semantic graph that maximize the response of $\mathbf{w}^{(t-1)}$:

$$(L_i^{(t)}, s_i^{(t)}) = \underset{L,s \in \mathcal{N}(L_i^{(0)}, s_i^{(0)})}{\operatorname{argmax}} \langle \mathbf{w}^{(t-1)}, \mathbf{x}(I_i, L, s) \rangle \quad (1)$$

Where, $\mathcal{N}(L, s)$ denotes all the locations and scales for which the corresponding rectangles have an overlap (defined as the intersection over union of areas) greater than τ=0.5 with the rectangle at (L, s). Then, we retrain $\mathbf{w}^{(t)}$ using the updated list of matches $\mathbf{x}(I_i, L_i^{(t)}, s_i^{(t)})$ as positive examples, and continue to next iteration until convergence. To make the process robust under visual diversity, we only retain $\mathbf{w}^{(t)}$ using the k matches with the highest score under $\mathbf{w}^{(t-1)}$. In practice the process converges in a few iterations.

This procedure is illustrated in Figure 5. For each of three parts shown, the top row contains the initial hypothesized matches found using semantic graph (ordered by depth at which they were found). The bottom row shows the refined matches after the training converges, with location and scale at which the response to the filter $\mathbf{w}$ is maximized. The ordering now reflects the response in (1).

We use the Linear Discriminant Analysis (LDA) method of [11] to learn $\mathbf{w}$. The method replaces the entire negative set by a single Gaussian distribution estimated from a large number of images. This significantly speeds up the learning procedure as it avoids the hard-negative mining step commonly during training. However, we still have to perform the latent updates described in Equation 1 during training.

3. Experimental setup

Datasets for training and testing. For our experiments we divided the set of 288 annotated images as described in Section 2.1 into a training set of 216 images, and a test set of 72 images. We call this dataset *church-corr*. During training we only use the semantic graph edges entirely contained in the training set (*church-corr-train*), resulting in 617 correspondence pairs, each labelled with an average of five landmarks. The test set (*church-corr-test*) is used to evaluate the utility of parts for predicting the location of the human-clicked landmarks, a "semantic saliency" prediction task described in Section 5.

Since the *church-corr* dataset contains church buildings that occupy most of the image, we collected an additional set of 127 images where the church building occupies a small portion of the image to test the utility of parts for localizing them (Section 4). The chance performance of detection in these images is small. For these images we also obtained bounding box annotations and the set if further divided into a training set of 64 images and a test set of 63 images. We call this dataset *church-loc*.

Methods for training parts. We compare various methods of learning parts: (1) Exemplar LDA (random seeds): randomly sampled seeds w/o graph (2) Exemplar LDA (landmark seeds): seeds sampled on landmarks w/o graph (3) Latent LDA: seeds sampled on landmarks w/ graph (4) Discriminative patches [19].

The first ignores the annotations completely and can be thought of as a simplified version of [19]. It lacks the careful cross-validation and multiple rounds of training with k-means like clustering in between iterations. The second simply uses the landmarks to bias the seed sampling step, hopefully resulting in fewer "wasted" seeds. The third (our proposed method) additionally uses the correspondence annotations to find "similar" patches in the training set using the procedure described in Section 2.3. In comparison to [19], this step is computationally much more efficient since the search for "similar" patches is restricted to a small fraction windows in the entire set using the semantic graph.

We trained a set of 200 parts for various methods on the *church-corr-train* subset. For our graph-based learning we restricted the maximum depth of our breadth-first search to two. For [19] we used publicly available code provided by the authors which takes as input the number of desired

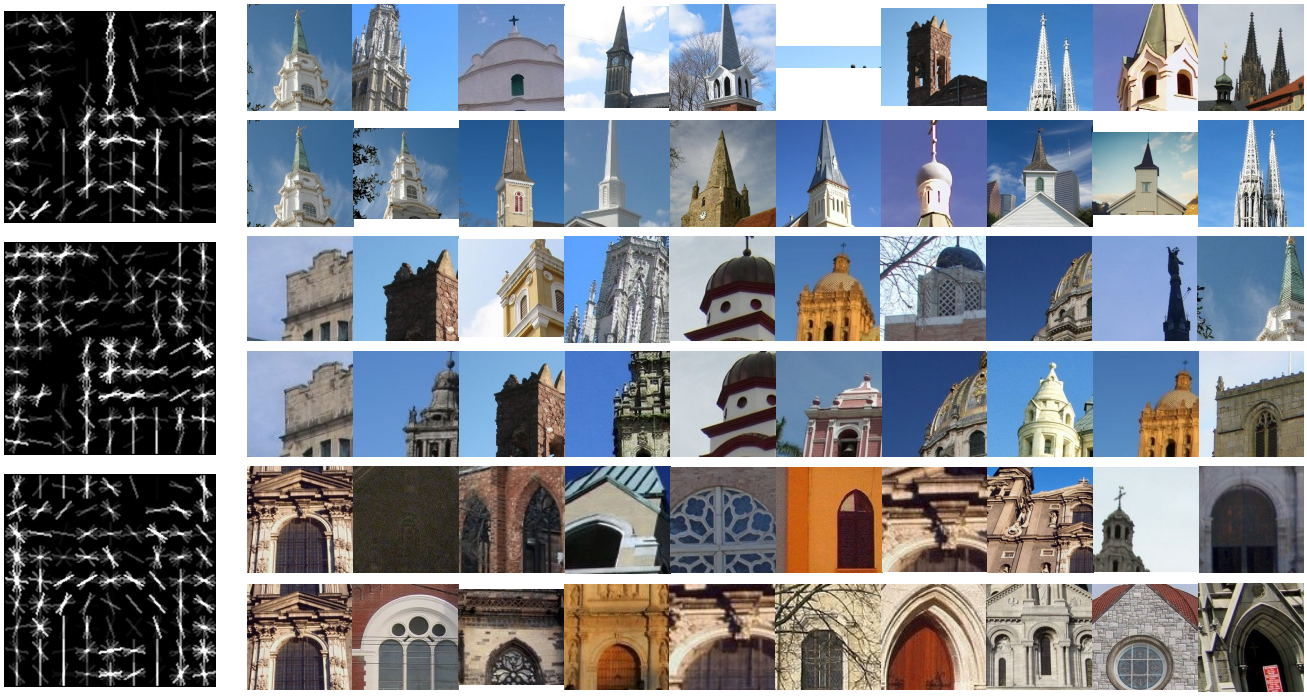


Figure 5. (*Left*) Learned HOG filter along with the top 10 locations of each part found using the semantic graph (*top row for each part*) and the latent search procedure (*bottom row for each part*) described in Section 2.3.

patches. During training however some of these are dropped resulting in a fewer number of trained patches, hence we trained a larger number of "discriminative patches" and selected a random subset of 200 for a fair comparison.

4. Detecting church buildings

The parts learned in the previous step can be utilized for localizing objects. On the training data we can estimate the spatial distribution of the object relative to the part and use this to predict the location of the object. Specifically, we use the top 10 detections on the *church-loc-train* set to estimate the mean offsets in scale and location of the object bounding box relative to the part bounding box. Figure 7 shows some parts, their estimated offsets, and top few detections.

We use a simple Hough voting based detector [13] for combining multiple parts. Votes from multiple part detections are combined in a greedy manner. For each image, part detections are sorted by their detection score (after normalizing to $[0, 1]$ using the sigmoid function) and considered one by one to find clusters of parts that belong together (based on the overlap of their predicted bounding boxes being greater than τ=0.5). We stop after n=500 part detections are considered. Each cluster represents a detection, from which we predict the overall bounding box as the weighted average of the predictions of each member and score as the sum of their detection scores. This is similar to the detection strategy using poselets [1].

Other baselines. In addition we trained the following deformable part models [8] using voc-release5 of the code [10]: (1) Single "root only" model without parts, (2) Mixture of three "root only" models, (3) Single "root + part" model, and (4) Mixture of three "root + part" models. These were trained on the same subset of images (*church-corr-train*) used for training our parts. As a reference the last model is nearly the state-of-the-art on the PASCAL VOC object detection challenge [6].

4.1. Results

We adopt the PASCAL VOC setup for evaluating detections. Bounding box predictions that overlap the ground truth bounding box (defined by the intersection over union) greater than τ are considered correct detections, while multiple detections of the same object are considered false positives. We refer the readers to [6] for details. We compare various methods for training parts *individually* and as a *combination* for localizing church buildings on the *church-loc-test* set. From our experiments we make the following observations:

Landmark seeds are better than random seeds. This can be seen in Figure 6 (*left*) which plots the performance of various parts sorted by the detection AP. The performance of the "exemplar LDA" method using "landmark seeds" is significantly better than using "random seeds", as can be seen by the difference between the red curve and the dashed black curve in 6 (*left*).

Using the semantic graph leads to further improvements. This can be seen as the difference between the solid black curve and the dashed black curve in Figure 6 (*left*). Moreover, the performance is better than the parts obtained using [19]. We used the same seeds for the "exemplar LDA" and "latent LDA" parts during training, hence we can compare the performance of each part individually for both these methods. This can be seen in Figure 6 (*middle*) which plots the performance of the 200 parts individually. "Latent LDA" improves performance 63.5% of the time.

Our part-based detectors compare favorably to state-of-the-art. We combine the predictions of the top 30 parts using the method described in Section 3 and evaluate it on *church-loc-test*. Figure 6 (*right*) plots the detection AP of various methods as a function of overlap threshold (τ) used for evaluation. Typically, τ=0.5 is used in a variety of object detection benchmarks (e.g. PASCAL VOC), but one can obtain a better insight about the performance by looking at the entire "AP vs. overlap" tradeoff curve. At τ=0.5, the "latent LDA" obtains an AP=39.90%, outperforming the DPM detector which obtains an AP=34.75%. The performance of "discriminative patches" is also quite good at AP=38.34%, while that of "random patches" (AP=16.67%) and "exemplar LDA" (AP=19.95%) is not very competitive. Out of the various DPM detectors we found that the single "root only" detector performed the best, hinting that a simple tree model of the parts is inadequate for capturing the variety in part layouts.

The difference between the various part-based methods and the DPM detector is more stark at a looser overlap threshold of τ=0.4, where there is a 15% gap between the "latent LDA" part-based detector and the DPM detector. This shows that many of the detections are near misses at τ=0.5. We believe that a better modeling of the part layouts can help with the bounding box prediction task.

Figure 8 shows high scoring detections on the *church-loc-test* set along with the locations of parts shown in different colors. The part activations reflect the variety in the layouts of different buildings. In addition to using the parts as a building block for a detector, we are interested in exploring their role in other scene parsing tasks. We describe two such experiments next.

5. Landmark saliency prediction

A landmark saliency map is a function $s(x, y) \rightarrow [0, 1], \sum_{x,y} s(x, y) = 1$, which is a likelihood that a location of the image is a landmark. We can evaluate the likelihood of a given set of ground truth landmark locations under the saliency map as a measure of its predictive quality. Assume a set of n images are all scaled to contain the same number of pixels m. Let $S_k, k = 1, \ldots, n$, denote the set of landmarks in the k^{th} image. The Mean Average Likelihood (MAL) is defined as:

$$\mathsf{MAL} = \frac{1}{n} \sum_{k=1}^{n} \left(\sum_{(x,y) \in S_k} \frac{m s(x, y)}{|S_k|} \right) \qquad (2)$$

According to this definition, the *uniform* saliency map has $\mathsf{MAL} = 1$ since $s(x, y) = 1/m, \forall x, y$.

Our saliency detector uses the top 30 parts sorted according to their part detection accuracy on the training set. Given an image, the highest scoring detections above the threshold, up to a maximum of 5 detections, are found for each part. Each detection contributes saliency proportional to the detection score to the center of the detection window. The contributions are accumulated across all detections to obtain the initial saliency map. This is then smoothed with a Gaussian with $\sigma = 0.01d$, where d is the length of the image diagonal, and normalized to sum to one, to obtain the final saliency map. We set the number of pixels $m = 10^6$.

Our approach can be seen as "category-specific interest points", and we compare this approach to a baseline that uses standard unsupervised scale-space interest point detectors based on Differences of Gaussians (DoG) and the Itti and Koch saliency model [12]. Table 1 shows the MAL scores for various approaches on the *church-corr-test* subset of our dataset. According to our saliency maps, the landmarks are $6.4\times$ more likely than the DoG saliency, and $4.2\times$ more likely than the Itti and Koch saliency. The "latent LDA" parts outperform both the "exemplar LDA" parts and "discriminative patches" [19] based saliency. Figure 9 shows example saliency maps for a few images for a variety of methods. As one might expect, our part-based saliency tends to be sharply localized near doors, windows, and towers.

Method	MAL
Difference of Gaussian	1.23
Itti and Koch	1.86
Discriminative patches	6.14
Exemplar LDA (Landmark seeds)	5.79
Latent LDA on the graph	**7.84**

Table 1. Mean Average Likelihood (MAL) of landmarks according to various saliency maps.

6. Fine-grained image parsing

Beyond the standard classification and detection tasks, the rich library of correspondence-driven parts allows us to reason about fine-grained structure of visual categories. For instance, we can attach semantic meaning to a set of parts at almost no cost by simply showing a human a few high-scoring detections. If the parts appear to correspond to a coherent visual concept with a name, say, "window" or "tower", the name for the concept is recorded. Figure 10 (*top row*) shows such labels assigned to various such

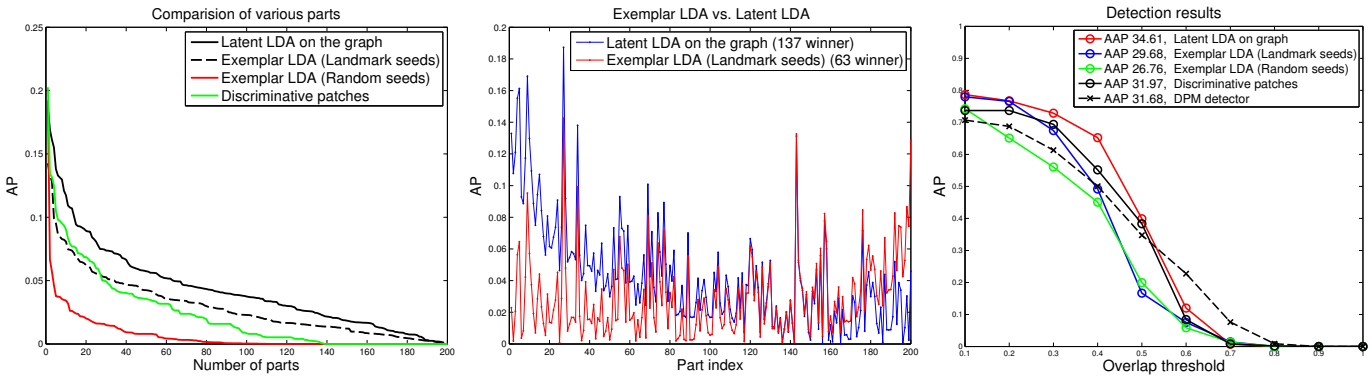
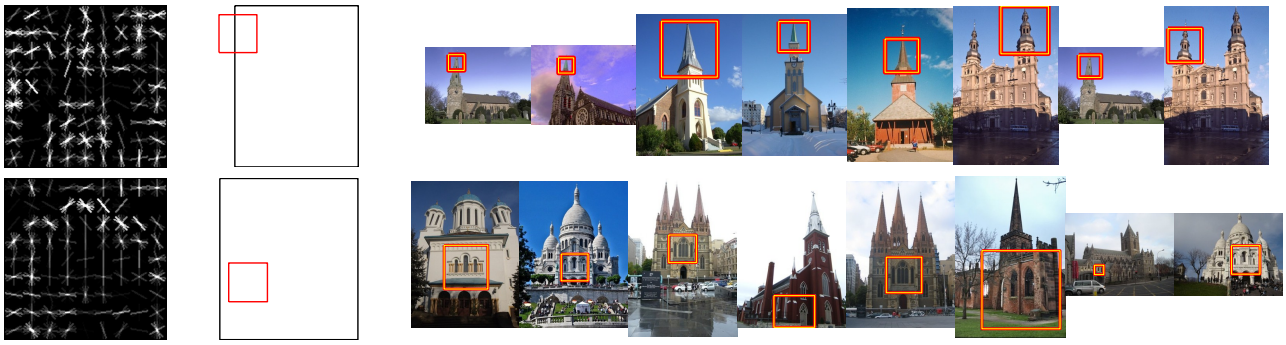


Figure 6. (*Left*) Detection performance of the 200 parts individually. (*Middle*) Comparison of parts using "latent LDA" and "exemplar LDA" using the same seeds. (*Right*) Detection performance of the combination of parts, and other baselines; AAP stands for average AP over thresholds.



Part filter Relative position Top detections on the test set

Figure 7. (*Left*) Learned part filters. (*Middle*) The vote associated with each part. (*Right*) Example part detections on test images.

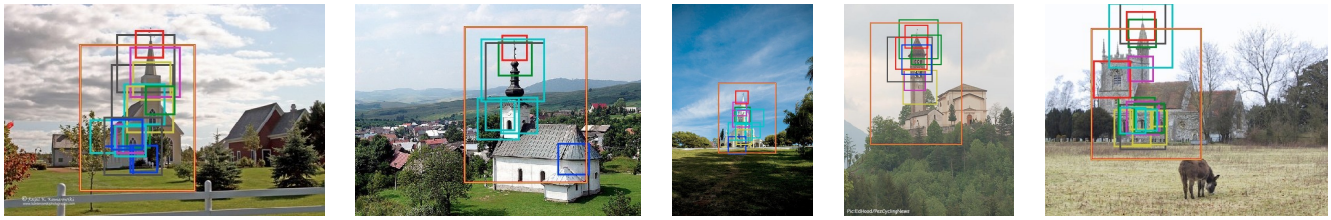


Figure 8. **Example church detections.** The corresponding parts in each detection are shown in different colors.

parts. These semantic labels can be visualized on new images by pooling the part detections across models that correspond to the same label. Figure 10 (*bottom row*) shows example images from the SUNS dataset [23], where we have visualized each image with labels positioned at the center of the detection window. Such parsing may be used for search and retrieval of images based on attributes such as "churches with windows on towers", "churches with two towers", etc.

7. Conclusions and discussion

We have described a method for semi-supervised discovery of semantically meaningful parts from pairwise correspondence annotations: pairs of landmark in images that are deemed matching. A library of parts can be discovered from such annotations by a discriminative algorithm that learns an appearance model for each part. On a category of church buildings, these parts are useful in a variety of ways: as building blocks for a part-based object detector, as category-specific interest point operators, and as a tool for fine-grained visual parsing for applications such as retrieval by attributes.

To exploit the rich part library discovered with the proposed framework for detection and segmentation, one likely needs an appropriate *layout* model connecting many parts into a coherent category model, beyond the simplistic star-graph model used in our experiments. Such a layout model is the subject of our future work on this topic.

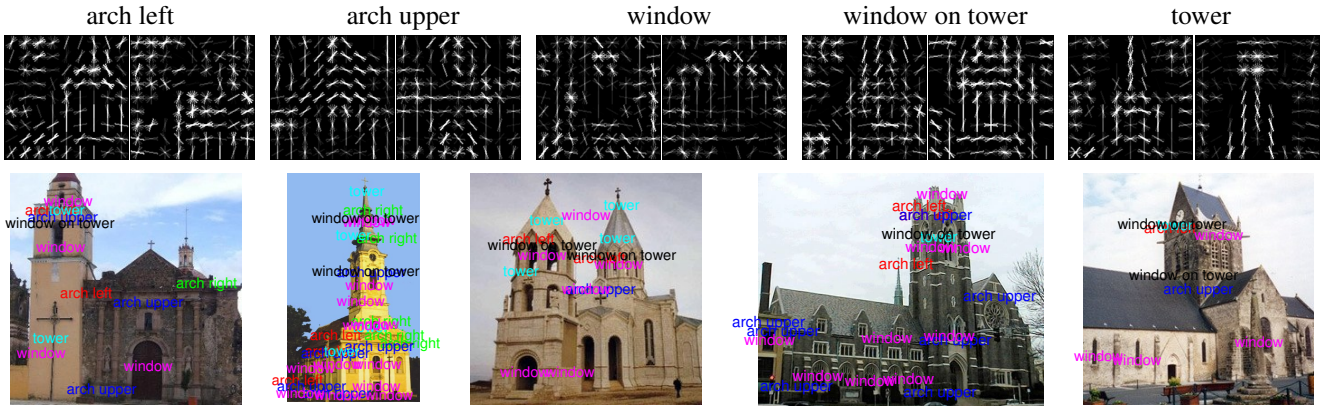


Figure 10. **Fine-grained parsing of images.** On the *top row* are labels assigned to parts by humans and on the *bottom row* are localized labels obtained by pooling the corresponding part detections on images.

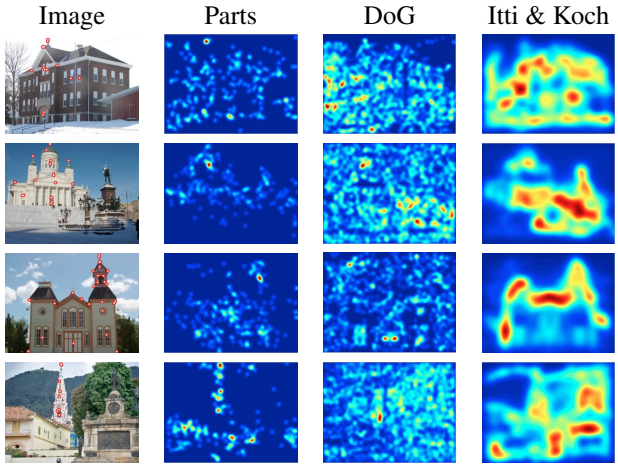


Figure 9. From *left* to *right* – images shown with the landmarks; saliency maps from our parts, Difference of Gaussian (DoG) interest point operator, and the Itti and Koch model.

References

[1] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010. 2, 5

[2] L. Bourdev and J. Malik. Poselets:body part detectors trained using 3D human pose annotations. In *ICCV*, 2009. 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3

[4] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes Paris look like Paris? In *ACM SIGGRAPH*, 2012. 2

[5] K. Duan, I. Bloomington, D. Parikh, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012. 2

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) challenge. *IJCV*, 88(2), jun. 2010. 5

[7] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010. 2

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 32(9), 2010. 2, 5

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61, January 2005. 2

[10] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/. 5

[11] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012. 4

[12] L. Itti and C. Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3), 2001. 6

[13] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1), 2008. 5

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004. 2, 3

[15] S. Maji and G. Shakhnarovich. Part annotations via pairwise correspondence. In *Workshop on Human Computation, AAAI*, 2012. 2, 3

[16] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *ICCV*, 2011. 3

[17] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR*, 2006. 2

[18] D. Parikh and K. Grauman. Interactive discovery of task-specic nameable attributes. In *Workshop on Fine-Grained Visual Categorization, CVPR*, 2011. 2

[19] S. Singh, A. Gupta, and A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 2, 4, 6

[20] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua. LDA-Hash: Improved matching with smaller descriptors. *IEEE PAMI*, 34(1), 2012. 2

[21] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *CVPR*, 2000. 2

[22] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, 2000. 2

[23] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 7