Pose Pooling Kernels for Sub-category Recognition

Ning Zhang
ICSI & UC Berkeley
nzhang@eecs.berkeley.edu

Ryan Farrell
ICSI & UC Berkeley
farrell@eecs.berkeley.edu

Trever Darrell
ICSI & UC Berkeley
trevor@eecs.berkeley.edu

Abstract

The ability to normalize pose based on super-category landmarks can significantly improve models of individual categories when training data are limited. Previous methods have considered the use of volumetric or morphable models for faces and for certain classes of articulated objects. We consider methods which impose fewer representational assumptions on categories of interest, and exploit contemporary detection schemes which consider the ensemble of responses of detectors trained for specific pose-keypoint configurations. We develop representations for poselet-based pose normalization using both explicit warping and implicit pooling as mechanisms. Our method defines a pose normalized similarity or kernel function that is suitable for nearest-neighbor or kernel-based learning methods.

1. Introduction

Recognition of fine-grained categories is a significant challenge for contemporary computer vision systems; such categories may be distinguished by relatively localized characteristics which may be difficult to learn from limited amounts of training data in a conventional 1-vs.-all learning framework. When a set of related classes share certain structure it is possible to learn pose estimators from data pooled across the several categories; in general terms, the ability to normalize for pose based on a super-category landmark or pose detector can significantly improve recognition of individual categories with limited amounts of training data.

Approaches to pose normalization have long been used in face recognition [9, 18]; for convex objects pose can be modeled as a rigid motion optionally with a non-rigid deformation. When the more general class of articulated objects is considered, the problem of pose estimation becomes more complex. Recently landmark template or "poselet" based pose estimation has been a topic of increasing interest [7, 5, 4]; In our previous work [12], we exploited such models to construct pose-normalized descriptors that operated on articulated objects. However, this model required the instantiation of 3-D volumetric primitives to form a representation, which is costly to obtain manually and can be problematic in some cases (see Figure 1).

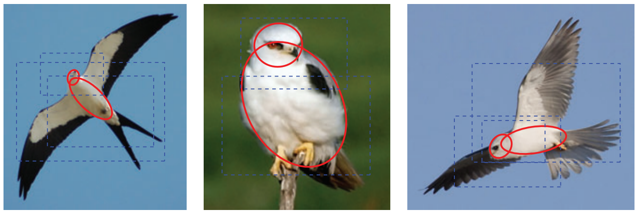
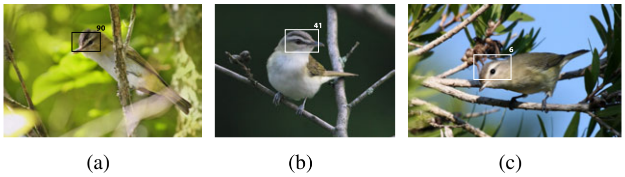


Figure 1. **Limitations of Head/Body Volumetric Representation.** A volumetric representation (red ellipsoids) such as that presented in [12] will be insufficient to determine which of the two birds in flight the perched bird matches. The wings and tail (both color and shape) carry nearly all of the discriminative appearance information, and could be modeled just fine with a poselet ensemble (blue dashed boxes). *Can you tell which bird it matches?*

In this paper we also tackle the issue of geometric normalization for sub-category recognition but advocate for a 2-D rather than 3-D representation. We presume a detection model in the style of [7, 5, 4], which results in a set of poselet-style activations on a given image, and explore the issue of how such sets of detected features should be best compared between two images. We develop similarity functions which take poselet activation "stacks" as input, and are suitable for use in nearest-neighbor classifiers or as SVM kernels.



(a) (b) (c)

Figure 2. **Comparing Poselet Appearances.** For subcategory recognition using discriminative classifiers (or nearest-neighbors) we need a mechanism to compare sets of poselets. Three different poselets may be actually covering the same underlying part in different pose; we therefore need a way to compare appearance based on those poselets. *Can you tell which two birds are the same?*

The key idea behind our similarity function is illustrated in Figure 2, where three different poselets are illustrated firing across different bird instances. The right two images depict instances of the same subcategory; a whole image (or whole-bird) comparison, e.g., using spatially pyramid matching kernel or bag or words, would likely miss the significant correspondence in the appearance of the two birds. However, by exploiting knowledge that the two poselets in the example are actually overlapping the same part (or parts), we can define a comparison function that explicitly compares descriptors formed over the corresponding poselet regions in the two images.

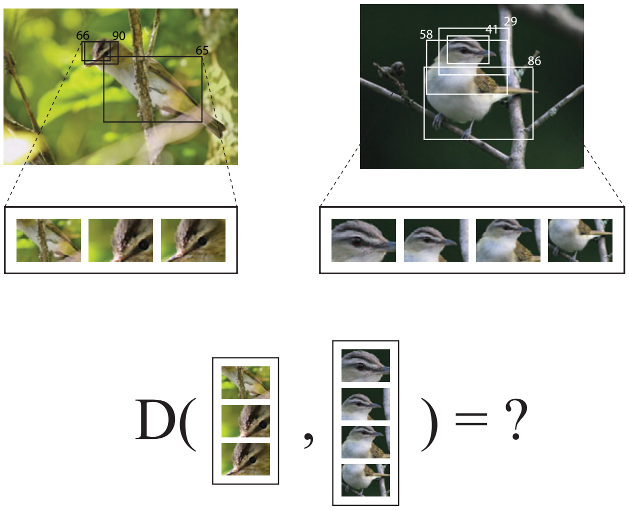


Figure 3. **Image Similarity by Poselet Set Similarity.** We propose to measure image similarity by defining a series of poselet-set similarity measures. Instead of considering image statistics globally within the image, we advocate the use of poselets as a means to tie the object appearance within image patches to that of semantically similar parts found in the training data. This effectively provides a high degree of pose invariance.

We define and compare a series of poselet-set similarity measures, or kernels. One intuitive idea is to use a greedy match kernel with explicit geometric warping based on landmark correspondences, constructing a match kernel that greedily estimates a minimum cost correspondence. This method is elegant, but computationally intractable in most situations. We then consider representations which form a fixed length vector: one variant attempts to normalize within the representation per example using a warping function, while a simpler model pools descriptors over corresponding poselets. Our pooling scheme establishes correspondences between poselets based on the degree of overlap each poselet exhibits: conceptually, the goal is to pool descriptors for poselets that actually are covering the same part or parts.

We evaluate our methods on the recently introduced CUB bird dataset, comparing recognition performance of our various descriptors given noisy detections. Overall, we find a significant boost from our proposed pooling architecture when compared to baseline methods that do not normalize for pose. Our results demonstrate that effective pose normalization is possible even for classes that do not admit a robust volumetric description. While our experiments have been limited to the bird domain, we expect our pose pooling kernels to be useful in a variety of other recognition tasks where there is considerable pose variation yet limited training data per category.

2. Background

Previous work on subordinate categorization includes approaches that learn discriminative image features. Such domains that have been previously considered include: subordinate categories of flowers (Nilsback and Zisserman [25, 26], which introduced the 17- and 102-category Oxford Flowers datasets), two subclasses for each of six basic object categories, e.g., Grand vs Upright Pianos, (Hillel *et al.* [2]), and subordinate categories of stonefly larvae, which exhibit tremendous visual similarity (Martínez-Muñoz *et al.* [23]).

A significant literature seeks to leverage similarities between categories to improve recognition performance. Methods which exploit class taxonomies or hierarchies range from constructing latent topic hierarchies [3] to sharing appearance [16, 27], classifiers [1] or visual parts [28] to constructing efficient classification trees [17, 22], and other references too numerous to mention here. Each such approach provides insights or advances toward efficiently solving basic-level classification. These unsupervised approaches, however, cannot be readily applied to the problem of distinguishing closely-related subordinate categories which, by definition, share a common set of parts and yet can have both subtle and drastic appearance variations.

Several authors have investigated attribute-based recognition, which are highly relevant for the general problems of subcategory recognition, see for example [10, 11, 19, 20, 33]. These techniques often learn discriminative models from attribute-labeled training data and subsequently apply the learnt models to estimate the appropriate visual attributes present in a test image. While attribute-based models are suitable for addressing the one-shot learning problem (previously considered in [13, 14, 15, 24] among others), they typically focus on relatively coarse grained attributes. Our focus is on representations suitable for the fine-scale distinctions needed for subordinate categorization.

The work of Branson *et al.* [8] proposes improving recognition accuracy by interleaving computation with attribute queries made to a human subject. Their method eval-

[1] Images in Figure 1 (CC) Jeff Whitlock, Ingrid Taylor and Bill Bouton (http://goo.gl/Kpw4z, http://goo.gl/wf4rO, and http://goo.gl/PtNFS respectively).

uates recognition in a large 200-category bird dataset [34] which also the subject of our experimentation.

We base our method on the poselet framework, as described in [7, 5], see also the related technique of [4]. We explore the idea of pose-normalization for sub-category appearance descriptors in this framework, a topic previously considered in [12]. The paradigm explored there was to employ volumetric pose normalization using 3-D primitives, following the line of work established by [9, 18] for the case of face recognition. However, in contrast to [12], we explore a method that has fewer representational assumptions: in particular our method does not employ volumetric representations, and therefore is applicable to object classes which do not strictly admit such a model. Additionally, and more significantly, our method does not require 3-D pose annotation, as does the method in [12]. Recent work in the poselet community has considered the task of activity recognition and attribute description [6]; this work computes feature vectors comprised of poselet detector activations. In contrast, our method (and that of [12]) forms *descriptors* over the localized poselet detections; the contribution of this paper is to define and analyze various 2-D schemes for comparing sets of poselet-based descriptors in such a way that poselets which correspond to the same underlying part or region of an object are aligned so that the corresponding descriptors can be effectively compared.

3. Pose Normalization Kernels

Given an ensemble of learned poselets, poselet detection methods (reviewed above) can infer a set of detections for each image. Our goal is to use these detections to compute sub-category level descriptors that are effective at discriminating, e.g., individual species. In particular, we would like to explore schemes for comparing the pose-normalized appearance of two detected instances of a particular poselet model. We compute descriptors at each poselet activation, and consider various strategies for comparing these sets of descriptors in the following subsections.

In order to use discriminative classifiers for subcategory recognition, we therefore need a mechanism to compare two sets of poselet detections. The poselet detection process provides estimates of part locations; our conjecture is that comparing the image descriptors which correspond to the same physical part (or collection of parts) will improve classification performance when compared to using just the whole image without any pose normalization. In general, sub-category recognition depends on the subtle appearance variations of some parts: two different poselets may be actually covering the same underlying part just in different poses or views, so it is desirable to have a similarity function which can properly relate descriptors from various poselet detections when comparing sets of responses. We consider various approaches to this problem below, including schemes which compute a poselet to poselet normalization via geometric warping prior to comparing descriptors, and those which pool descriptors across corresponding (semantically similar) poselets.

To directly apply nearest neighbor and kernel-based classifiers to our sub-category recognition problem, we define kernel functions based on sets of detected poselets. These functions can be used e.g., in SVM or Gaussian Process based classifiers or regression schemes.

3.1. Preliminaries

Each image X_i has a set of poselet activation windows with the corresponding activation scores $\mathbf{t^i} = \{t_1^i, \cdots, t_N^i\}$ where N is the number of poselets. Suppose we extract a d-dimensional image descriptor $\phi(X_u^i)$ from each poselet u's activation window, such as bag of words SIFT or spatial pyramid histogram. Then each image can be represented as $X_i = \{t_1^i, t_2^i, \cdots, t_N^i, \phi(X_1^i), \cdots, \phi(X_N^i)\}$. Also, between each pair of poselets u and v, we compute the transformation function T_{uv} from poselet u to poselet v and the confidence score λ_{uv} of this transformation.

The affine transformation function T_{uv} is computed based on the keypoints locations of two poselets. If two poselets have less than three common keypoints, there would not be an appropriate affine transformation between the two sets of keypoints. In that case the $T_{uv}$is set to be empty and the confidence score λ_{uv} is set to be zero. Otherwise, we compute the affine transform T_{uv} from the keypoint sets of poselet u to keypoint sets of poselet v and the confidence score is set based on the number of overlapping keypoints, i.e. $\lambda_{uv} = \frac{K}{min\{K_u, K_v\}}$, where K is the number of the common keypoints and K_u is the number of keypoints of poselet u.

Ideally, we first consider a match kernel in the spirit of [32], which would compare two sets of poselet activations by transforming each poselet detection in one image to another poselet detection in a second image, and then comparing the corresponding image descriptors. A greedy warp match kernel would be defined as follows

$$\begin{aligned} &K_G(X_i, X_j) = \\ &\sum_{u,v} t_u^i \cdot t_v^j \cdot \frac{1}{2}\{\lambda_{uv} \cdot \widetilde{K}(\phi(T_{uv}(X_u^i)), \phi(X_v^j)) \\ &+\lambda_{vu} \cdot \widetilde{K}(\phi(X_u^i), \phi(T_{vu}(X_v^j))\} \end{aligned} \tag{1}$$

where $\widetilde{K}$ is the base kernel between aligned poselets, $\phi(T_{uv}(X_u^i))$ is the image descriptor after warping the activation window from poselet u to poselet v; taking the average of both warping directions makes the kernel symmetric. The weights λ_{uv} and λ_{vu} are the confidence of the transformation, based on the number of overlapping points.

As described in more detail below an appropriate kernel function for the aligned poselets could be a simple linear kernel or a nonlinear kernels such as the chi-squared distance, computed between histogram-of-gradient descriptors extracted at each detected poselet location.

With this kernel, the similarity between each pair of images is just the weighted sum of similarities between the pose-normalized image descriptors. This kernel function can be effective, but suffers high computational costs when the number of detected poselets is large. It takes $O(n^2N^2)$ time where n is the number of images and N is the number of trained poselets. This method therefore may not scale well in cases where large datasets are involved. In the following sections we therefore consider intermediate fixed-length representations, yet which employ warping or more directly, pooling to align corresponding poselets.

3.2. Warped Feature Kernel

To overcome the quadratic complexity of a naive match kernel which compares sets of detections explicitly, we consider fixed-length representations that capture the set of poselet views of an object. As this defines a vector-space, it can be directly used as a feature vector in a e.g., chi-square or a radial-basis-function (RBF) kernel.

The most straightforward representation simply concatenates the image descriptor of each poselet to a long fixed length feature vector. This trivially represents the image's appearance under different poses, and serves as a baseline method. However, with no geometric normalization, this feature vector will perform poorly unless available training data cover all possible poselet activations for all classes.

Following the notation above, the simple fixed length representation is

$$\Psi(X) = [t_1 \cdot \phi(X_1), \cdots, t_u \cdot \phi(X_u), \cdots, t_N \cdot \phi(X_N)] \quad (2)$$

where $\phi(X_u)$ is the image descriptor of poselet u's activation window and t_u is the activation scores. This feature vector has length Nd where d is the dimension for image descriptor. Figure 4 illustrates this method.

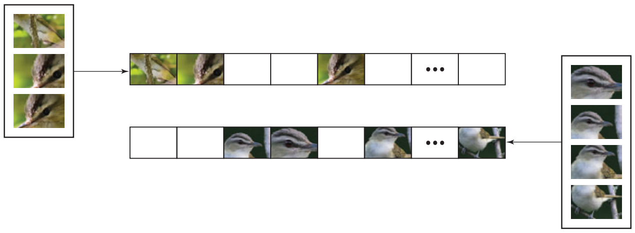


Figure 4. **Fixed-length representation**. Concatenated descriptors without warping.

A significant issue with this feature representation is that the feature vector is sparse as only a small number of poselets are detected in a typical image ($\sim$ 10 in our experiments on the data described below). Also, the representation may be redundant, since distinct poselets are often overlapping and therefore are describing the same object region in different poses and views. To overcome this, we consider ways to pose-normalize this representation.

Our first approach follows in the spirit of the fixed-length representation considered above, and explicitly warps poselet appearance within the fixed-length representation to fill in poselets that have not fired on an image. Effectively, this fills in blank feature blocks in the fixed length representation. As an example, suppose poselet #19 and poselet #23 both capture the left side of the bird's head with only slightly different orientation. For one image showing the left side of a bird's head, it might just fire poselet #19, whereas in another image, poselet #23 would fire. Both poselets represent the same part of the bird (the left side of the head) and it will improve the classification if this correspondence can be captured in the feature vector representation.

Thus, for each $\phi(X_u)$ in the $\Psi(X)$ in Eq. 2 that has not been detected but for which there exists another detected poselet which is similar enough to it, we use the image descriptor of the fired poselet and warp it to the non-fired poselet. With this approach the feature representation is

$$\begin{aligned}\Psi_{warp}(X) \quad = \quad & [t_1 \lambda_{p_1} \phi(T_{p_1}(X_1)), \cdots, t_u \lambda_{p_u} \phi(T_{p_u}(X_u)) \\ & \cdots, t_N \lambda_{p_N} \phi(T_{p_N}(X_N))] \end{aligned} \quad (3)$$

where p_u is the index of most similar fired poselet that should be warped to the non-fired poselet u. If this poselet already fires, it sticks to Eq. 2 and if there is no appropriate fired poselets to warp, the corresponding feature for the non-fired poselet is set to zero. We use the residual error after transformation as the measurement of two poselets' similarity. Figure 5 illustrates this method.

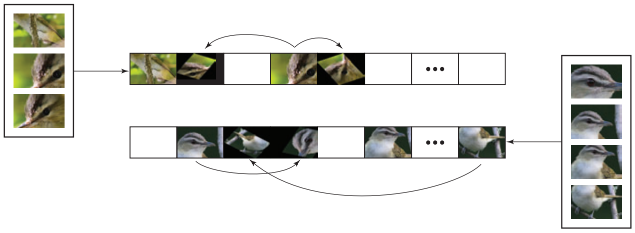


Figure 5. **Warped Feature Kernel**. Concatenated descriptors with warping.

3.3. Pooled Feature Kernel

The intuition behind the fixed length warping kernel is to have a pose-normalized way to compare images which have the correspondences in different parts. A further extension of this model is to group or pool poselets which represent the same underlying part into a cluster of parts.

By design, groups of learned poselets exhibit redundancy: several poselets will represent the same part or parts in slightly different configurations with respect to the camera. For recognition, it is desirable to group them together when comparing representations. We therefore consider a pooling stage on top of our base representation, which groups together the descriptors computed on poselets that are identified as being in correspondence. This strategy is particularly effective for additive kernels such as bag-of-word representations formed over local features, but can also work to a degree on non-additive representations.

We consider two criteria for grouping the poselets into clusters, each containing poselets that representing the same part of an object. One could treat this in a fully supervised fashion, based on provided part annotations; however, we chose to consider an unsupervised approach that discovered clusters in a data-driven fashion.

As illustrated in Figure 6 our pooling scheme forms a cluster feature vector, whose length is equal to the number of clusters times the length of the poselet descriptor. For each cluster, the descriptors are pooled across the poselets assigned to the cluster, producing a single descriptor for the cluster. The final cluster feature vector is the concatenation of the cluster descriptors, as given in the following equation:

$$\Psi_{pool}(X) = [avg_{i\in C_1}\Psi_{warp}(i), \cdots, avg_{i\in C_P}\Psi_{warp}(i)]$$

where C_i is the i-th poselet cluster.

Each poselet cluster should ideally correspond to a coherent part or part group and all the poselets within each group are similar to each other. Using such a clustering scheme, the output pooling image descriptor is much more representative in describing the image features of different parts.

We compute poselet clusters using a greedy clustering scheme, which first forms a graph over the learned poselets with edge distances computed to reflect a measure of inverse poselet correspondence. We have used two different measurements for edge distance:

1. warp distance — using the residual error of the affine transformation between keypoints corresponding to two poselets.
2. keypoint distance — $1/\lambda$ as defined above, based on the number of keypoints common to two poselets.

which lead to distinct clustering results. Below we compare the two pooling results in terms of classification performance. We randomly pick poselets as candidate cluster centers, grouping together a sufficient number of neighbors under one of the two criteria above. We repeat until all poselets are iteratively assigned to a cluster center. Specifically, the clustering algorithm first randomly picks one poselet as the cluster center then groups the rest of poselets which have a distance within a set threshold. Then it iteratively picks another unselected poselet as the new cluster center and repeats the process until there are no good clusters. This method has the benefit of not requiring knowledge of the number of clusters a priori. Other clustering schemes may prove superior to this greedy method and will be an area considered in future work. The method described above worked well in practice.

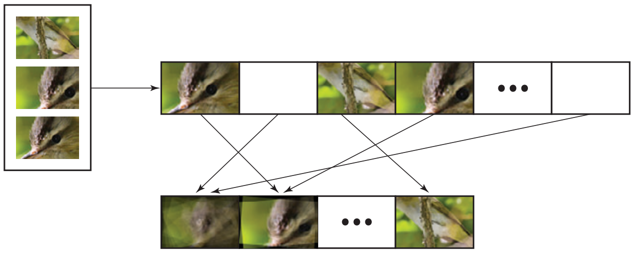


Figure 6. **Pose Pooling Kernel.** Corresponding poselets are grouped.

4. Experiments

We now present experiments validating the effectiveness of our approach for fine-grained object categorization.

4.1. Dataset

Following [8] and [12], we conduct experiments on the 200-category Caltech UCSD Birds Dataset [34], one of the most complete datasets for fine-grained object categorization. We utilize the extended version of the dataset that was recently released [31] which provides approximately 60 images per category, twice what the initial dataset provided.

We use this dataset primarily because of the 15 part annotations (e.g. beak, crest, throat, left-eye, right-wing, nape, etc.) that it provides per image/object. These part annotations are important for our approach as they facilitate the generation of poselets following the 2D keypoint-based paradigm presented in [5].

4.2. Implementation Details

To improve the clarity of the earlier technical sections (Sections 3.2 and 3.3), we omit implementation details that are nonetheless important to the experiments. These details include the computing of canonical poselet activations per image and the descriptors used to encode activation patch appearances.

4.2.1 Poselet Activations

In an effort to evaluate the subcategory classification performance independent of detection errors, we implement a poselet-style detector and train several templates using the training data, finally computing "ground truth" activations on the test set. Each poselet detector is trained as follows:

1. A training image is selected at random and a rectangular window overlapping a subset of the object's keypoints is randomly chosen.
2. A selection of similar images from the training set (those with locally similar keypoint configurations) is collected.
3. Distributions for the relative location of each relevant keypoint are computed and stored.

Once a large set of such poselets (1000 in our experiments) is trained, we use a beam-search based selection strategy to prune this large randomly generated set. The large set will be heavily biased toward the frequently occurring poses. The selection criteria are defined such that the pruned poselet set better covers the full pose space of the training set. Without this step, there will be images (both in the training and presumably the test sets) there will be images with a disproportionately small number of poselets firing, simply because the subject is in a less frequently observed pose.

Next, we use this smaller poselet set (100 in our case) to calculate a set of "ground truth" activations for each test image, accomplished by comparing each poselet's keypoint distributions with the locations of the respective keypoints (if present) in the test image. This comparison is performed by finding the best procrustean fit for the keypoints shared by a given trained poselet and a given test image. As poselets are not invariant to orientation, we only declare activations as valid if the transformation produced by the procrustean analysis has a small deviation in orientation (we use a tolerance of $\pm 10°$).

4.2.2 Patch Appearance Descriptors

We consider a few different measures for describing the appearance of the image patch underlying a given activation. Ultimately, the descriptors are concatenated into a single vector per image and are passed to a support vector machine (SVM) for classification (using a 1 vs. all policy). We consider the following two appearance descriptors.

- Bag of Words (BOW-SIFT) - This descriptor is generated by densely computing SIFT features (at multiple scales) and vector quantizing them against a codebook.

- Pyramidal Histogram of Words (PHOW) - Following Spatial Pyramid Match [21], the SIFT features are quantized and then binned into regions defined by a spatial subdivision pyramid.

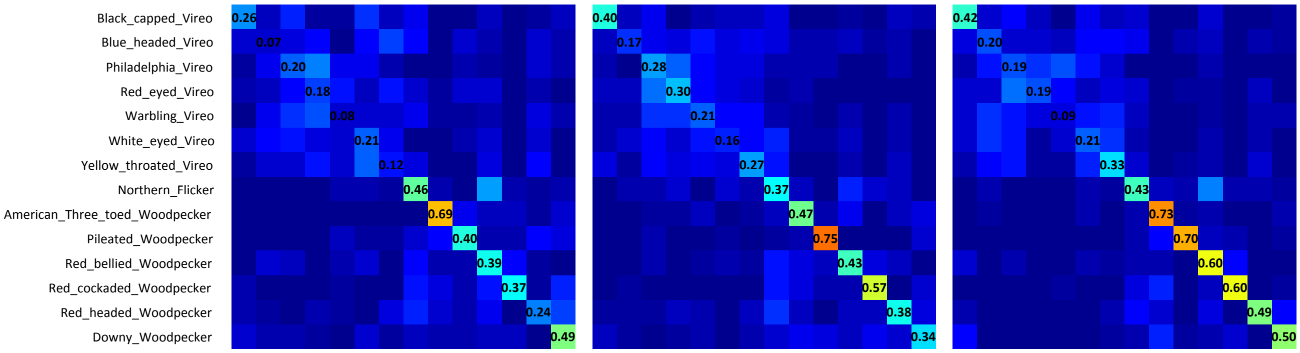
In our experiment, we use the bag of words (BOW-SIFT) and pyramidal histogram of words (PHOW) as our appearance features. Specifically, we use the VLFEAT toolbox [29] to compute the patch descriptors with a codebook of 1024 elements. For the spatial pyramid, following standard convention we subdivide the image at three different levels of resolution. For each level, we concatenate the histogram of each spatial bin and the weight for the lth pyramid level is set to $\frac{1}{2^{(L-l)}}$ where L is the the total number of layers (3 in our experiment). Given the activation windows and image descriptors, we can compute the warped and/or pooled features as discussed in Section 3. Then, we use SVMs for classification and using either a linear kernel or the efficient additive kernel map in [30] for χ^2 and Intersection kernels.

4.3. Results

We now present our experimental evaluation and begin by defining a baseline for comparison. As noted previously, there are three approaches (to our knowledge) that have presented categorization results on the CUB200 dataset. The authors of [8] leverage attributes provided by a human-in-the-loop to supplement a machine vision back end for classification. In [12], categorization is performed in a pose-normalized space on a two family (14-category) subset of the full CUB200 dataset. The authors in [35] proposed a fine-grained classification approach using random forests with discriminative decision trees, tested on all 200 categories. We evaluate our methods in both the 14 category and 200 category settings. We use the VLFEAT toolbox [29] as a baseline, which applies a linear SVM to vector quantized SIFT features from within the bounding box.

Figure 7 depicts the confusion matrices for categorization on these two families using a linear SVM with 15 training examples per category (plus their reflections to yield 30 training examples). The warped feature kernel uses a linear SVM to classify the features described in Section 3.2 while the pose pooling kernel follows the method in Section 3.3 also using a linear SVM. Both feature kernels have the same bag of word SIFT descriptors used by the baseline method. The confusion matrices show that both the warped feature kernel and the pooled feature kernel improve the baseline methods of using just the bounding box image. From the additional results presented in Table 1, we find that for both training settings ($N = 15$ and $N = 30$ training images), the warped feature kernel using linear SVM improves both and pose pooling kernel outperforms the warped feature kernel. Warping poselets also helps the pooling stage and both cluster schemes work well and warping distance based clustering works slightly better than overlapping keypoints based clustering.

We also apply a spatial pyramid to the quantized SIFT features; the results are shown in Table 2. Here we observe that for the two different training settings, pose pooling kernels outperform the baseline and the χ^2 kernel usually outperforms the intersection kernel. All these results are similar to the previous results using BOW-SIFT features, but using the spatial information in the image descriptor improves the categorization results.



(a) Baseline-VLFEAT (29.73% MAP) (b) Warped Feature (36.33% MAP) (c) Pose Pooling Kernel (40.60% MAP)

Figure 7. Confusion matrices on 14 categories across two bird families (for comparison with [12]) using 15 images per training category. All three methods a linear SVM for classification; the pose pooling kernel uses the distance-based clustering described in section 3.3.

Method	Linear Kernel (N=15)	χ^2 Kernel (N=15)	Linear Kernel (N=30)	χ^2 Kernel (N=30)
Baseline (VLFEAT)	29.73	36.61	33.39	42.68
Fixed-length Feature(no warping)	33.61	36.10	45.08	46.10
Warped Feature Kernel	36.33	31.85	40.71	42.32
Pose Pooling (warping, distance)	**40.60**	**43.35**	44.61	52.44
Pose Pooling (warping, keypoints)	39.79	41.40	**46.12**	**52.75**
Pose Pooling (no warping, distance)	32.24	42.25	40.40	51.78
Pose Pooling (no warping, keypoints)	31.82	42.22	39.77	52.72

Table 1. Mean precision on the 14 categories from [12] using a bag of words model on SIFT features. N denotes the number of examples used for training per category and two different kernels (linear and χ^2) are used for the SVM. The distance/keypoints and warping/no warping refer to the distance- or keypoint-based pooling and pooling with or without descriptor warping.

Method	Intersection Kernel (N=15)	χ^2 Kernel (N=15)	Intersection Kernel (N=30)	χ^2 Kernel (N=30)
Baseline (VLFEAT)	40.06	41.03	48.61	49.11
Pose Pooling (warping, distance)	45.36	**46.91**	54.08	55.87
Pose Pooling (warping, keypoints)	**45.76**	45.98	**56.76**	**57.44**
Pose Pooling (no warping, distance)	43.73	44.10	54.09	55.09
Pose Pooling (no warping, keypoints)	43.22	43.88	55.00	54.99

Table 2. Mean precision on the same 14 categories using a spatial pyramid. The χ^2 and intersection kernels were used here due to the poor performance of the linear kernel.

Method	Linear	χ^2
Baseline(VLFEAT)	14.14	18.60
Pose Pooling(warp, distance)	23.44	**28.18**
Pose Pooling(warp, keypoints)	**24.21**	27.74
Pose Pooling(no warp, distance)	17.74	23.06
Pose Pooling(no warp, keypoints)	17.68	22.69

Table 3. Mean precision on the full 200 categories using BOW SIFT features. These results are not directly comparable to the results in [35], as an earlier version of the dataset was used there.

We also test our methods on the full 200 categories of the CUB dataset. We split the training/test according to the default split provided in the dataset and use the BOW SIFT feature as the image descriptor. Table 3 presents these results demonstrating that pose pooling kernel outperforms the baseline method; pooling on the warped feature has the best performance.

5. Conclusion

In this paper we demonstrate the ability to normalize pose based on super-category landmarks, and show that this can significantly improve models of individual categories when training data are limited. Our method does not require 3-D training data, and is suitable for categories that do not admit volumetric representations. Our scheme is based on contemporary poselet-based representation schemes which consider the ensemble of detector responses trained for specific pose-keypoint configurations. In contrast to existing 2-D approaches, our method computes a set of local descriptors at detected poselet locations, and uses these to form a fine-grained category model. We achieve pose normalization via explicit warping and implicit pooling; our method defines a pose normalized similarity or kernel function that is suitable for nearest-neighbor methods or kernel-based learning method.

The authors gratefully acknowledge support by DARPA contract W911NF-10-2-0059, by NSF awards IIS-0905647 and IIS-0819984, and support from Toyota and Google.

References

[1] B. Babenko, S. Branson, and S. Belongie. Similarity Metrics for Categorization: From Monolithic to Category Specific. In *ICCV*, 2009. 2

[2] A. Bar-Hillel and D. Weinshall. Subordinate Class Recognition Using Relational Object Models. In *NIPS*, 2007. 2

[3] E. Bart, I. Porteous, P. Perona, and M. Welling. Unsupervised learning of visual taxonomies. In *CVPR*, 2008. 2

[4] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing Parts of Faces Using a Consensus of Exemplars. In *CVPR*, 2011. 1, 3

[5] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting People Using Mutually Consistent Poselet Activations. In *ECCV*, 2010. 1, 3, 5

[6] L. Bourdev, S. Maji, and J. Malik. Describing People: Poselet-Based Approach to Attribute Classification. In *ICCV*, 2011. 3

[7] L. Bourdev and J. Malik. Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations. In *ICCV*, 2009. 1, 3

[8] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual Recognition with Humans in the Loop. In *European Conference on Computer Vision (ECCV)*, 2010. 2, 5, 6

[9] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active Appearance Models. In *PAMI*, volume 23, 2001. 1, 3

[10] A. Farhadi, I. Endres, and D. Hoiem. Attribute-Centric Recognition for Cross-category Generalization. In *CVPR*, 2010. 2

[11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing Objects by their Attributes. In *CVPR*, 2009. 2

[12] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate Categorization Using Volumetric Primitives and Pose-Normalized Appearance. In *ICCV*, 2011. 1, 3, 5, 6, 7

[13] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories. In *ICCV*, 2003. 2

[14] L. Fei-Fei, R. Fergus, and P. Perona. One-Shot learning of object categories. *PAMI*, pages 594–611, 2006. 2

[15] A. Ferencz, E. G. Learned-Miller, and J. Malik. Building a Classification Cascade for Visual Identification from One Example. In *ICCV*, 2005. 2

[16] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba. Semantic Label Sharing for Learning with Many Categories. In *ECCV*, 2010. 2

[17] G. Griffin and P. Perona. Learning and Using Taxonomies for Fast Visual Categorization. In *CVPR*, 2008. 2

[18] M. J. Jones and T. Poggio. Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes. *IJCV*, 29, 1998. 1, 3

[19] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *ICCV*, 2009. 2

[20] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In *CVPR*, 2009. 2

[21] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006. 6

[22] M. Marszałek and C. Schmid. Constructing Category Hierarchies for Visual Recognition. In *ECCV*, 2008. 2

[23] G. Martínez-Muñoz, N. Larios, E. Mortensen, W. Zhang, A. Yamamuro, R. Paasch, N. Payet, D. Lytle, L. Shapiro, S. Todorovic, A. Moldenke, and T. Dietterich. Dictionary-Free Categorization of Very Similar Objects via Stacked Evidence Trees. In *CVPR*, 2009. 2

[24] E. Miller, N. Matsakis, and P. Viola. Learning from One Example Through Shared Densities on Transforms. In *CVPR*, 2000. 2

[25] M.-E. Nilsback and A. Zisserman. A Visual Vocabulary for Flower Classification. In *CVPR*, 2006. 2

[26] M.-E. Nilsback and A. Zisserman. Automated Flower Classification over a Large Number of Classes. In *ICVGIP*, 2008. 2

[27] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to Share Visual Appearance for Multiclass Object Detection. In *CVPR*, 2011. 2

[28] S. Todorovic and N. Ahuja. Learning Subcategory Relevances for Category Recognition. In *CVPR*, 2008. 2

[29] A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms. `http://www.vlfeat.org/`, 2008. 6

[30] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2010. 6

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-UCSD Birds 200-2011. In *FGVC*, 2011. 5

[32] C. Wallraven, B. Caputo, and A. B. A. Graf. Recognition with local features: the kernel recipe. In *ICCV'03*, pages 257–264, 2003. 3

[33] G. Wang and D. Forsyth. Joint Learning of Visual Attributes, Object Classes and Visual Saliency. In *ICCV*, 2009. 2

[34] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 3, 5

[35] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011. 6, 7