POOF: Part-Based One-vs-One Features for Fine-Grained Categorization, Face Verification, and Attribute Estimation

Thomas Berg
Columbia University
tberg@cs.columbia.edu

Peter N. Belhumeur
Columbia University
belhumeur@cs.columbia.edu

Abstract

From a set of images in a particular domain, labeled with part locations and class, we present a method to automatically learn a large and diverse set of highly discriminative intermediate features that we call Part-based One-vs-One Features (POOFs). *Each of these features specializes in discrimination between two particular classes based on the appearance at a particular part. We demonstrate the particular usefulness of these features for fine-grained visual categorization with new state-of-the-art results on bird species identification using the Caltech UCSD Birds (CUB) dataset and parity with the best existing results in face verification on the Labeled Faces in the Wild (LFW) dataset. Finally, we demonstrate the particular advantage of POOFs when training data is scarce.*

1. Introduction

Fine-grained visual categorization has become a popular area over the past several years. In contrast to basic-level recognition, in which we need to distinguish basic-level categories such as *chair* and *car* from each other, the fine-grained categorization problem asks us to distinguish subordinate-level categories such as *office chair* and *kitchen chair* from each other. One relatively well-studied example of fine-grained visual categorization is species or breed recognition.

Many of the most accurate approaches to fine-grained visual categorization are based on detecting and extracting features from particular parts of the objects. For example, in dog breed classification one may extract features from the nose and base of the ears [16, 23]. Face recognition is an extreme case of fine-grained visual categorization in which the "subcategories" are individual instances, and the best face recognition methods extract features from locations determined by finding facial landmarks such as the corners of the eyes [3, 31, 36]. Intuitively, we expect fine-grained visual categorization to require part-based approaches because the differences between subcategories are small and not noticeable from global, image-level features. Fine-grained visual categorization also conveniently *enables* part-based approaches, because objects within the same basic-level category will often have the "same" parts [25], allowing for easier comparison. For example since all dogs have noses, it is natural in dog breed recognition to attempt to detect and extract features from the nose. In basic-level categorization this approach is more difficult, as there is no natural corresponding part among instances of dogs, motorboats, and staplers.

Computer vision has produced a wide array of standard features, including SIFT [17], SURF [1], HOG [7], LBP [20], etc. A straightforward approach to part-based recognition is to extract some of these features at the part locations and build a classifier. In general, however, these standard features are unlikely to be optimal for any particular problem; what is best will likely vary both by domain (the best features for dogs are different from the best features for birds) and by task (the best features for face recognition are different from the best features for gender classification).

In this work, we build a framework for learning a large set of discriminative intermediate-level features, which we call *Part-based One-vs-One Features (POOFs)*, specialized for a particular domain and set of parts. The process of learning these features is illustrated in Figure 1. We start with a dataset of images in the domain, labeled by class and with part locations. For any pair of classes, for any pair of parts, we extract some low-level features in a grid of cells that covers the two parts, and train a linear classifier to distinguish the two classes from each other. (In our experiments we use histograms of gradient direction or color as the low-level features.) The weights assigned by this classifier to different cells of the grid indicate the most discriminative region around these parts for this pair of classes. We fix the support region for our feature based on these weights, and then retrain the classifier to find a discriminative projection. The combination of the two parts, the low-level feature, the learned support region, and the final projection form a POOF, which can produce a scalar

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.128

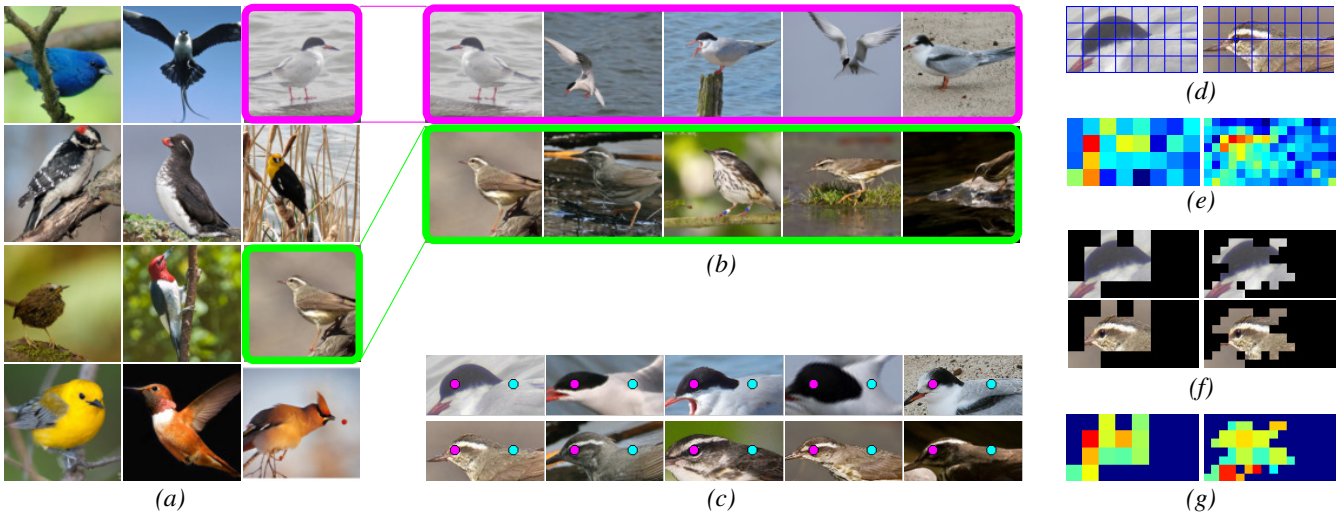

Figure 1. Learning a Part-based One-vs-One Feature (POOF) for bird species identification. Given (a) a reference dataset of images labeled with class (species) and part locations, a POOF is defined by specifying two classes, one part for feature extraction, another part for alignment, and a low-level "base feature." (b) Samples of the two chosen classes are taken from the dataset and (c) aligned to put the two chosen parts in fixed locations. (d) The aligned images are divided into cells at multiple scales, from which the base feature is extracted. A linear classifier is trained to distinguish the two classes, giving (e) a weight to each cell. We threshold the weights and find the maximal connected component contiguous to the chosen feature part, setting this as (f) the support region for the POOF. Finally, a classifier is trained on the base feature values from just the support region. The output of this classifier is our one-vs-one feature.

score (the decision value from the classifier) for any test image with locations for the two parts. This score is our intermediate-level feature. If our dataset does not have part locations, the straightforward simplification of taking the grid over the whole image or the object's bounding box produces OOFs rather than POOFs, but we do not analyzed their performance here.

This paper makes the following contributions:

- We present a fully automatic method for constructing a library of *Part-based One-vs-One Features (POOFs)* – discriminatively trained intermediate-level features – from a set of images with class and part location labels
- We demonstrate that POOFs significantly advance the state of the art on the Caltech-UCSD Birds dataset, obtaining a classification accuracy of 73.30% on the localized species categorization benchmark, quadrupling the accuracy reported in [27].
- We demonstrate that POOFs reduce the need for large training sets, showing that in the face domain they can be used as extremely effective intermediate features for tasks such as attribute labeling.

While each POOF is only known to be discriminative for the two classes used in its definition, we find that collections of POOFs are useful not only for classification into the classes in the reference dataset, but for other tasks in the same domain. We show examples in two domains, bird species and faces.

2. Related Work

Fine-grained visual categorization has seen a lot of work recently. Instance-level recognition can be seen as the "finest-grained" categorization, and is most commonly seen as face recognition or image search. Face recognition often takes a part-based approach, either by explicitly extracting features at landmark points (*e.g.* [6, 30, 34]) or by performing an alignment step based on parts, then extracting features from fixed locations (*e.g.* [3, 14, 31]). [3] in particular takes an approach similar to ours in using binary one-vs-one classifiers trained on a reference set as discriminative features for the domain. In all of these cases, however, the locations and shapes of the regions from which the features are extracted are designed and tuned by hand. In our work, the selection of regions for feature extraction is fully automatic, allowing the method to be applied easily to any domain.

Moving up a step from instance-level recognition, subordinate category recognition has been explored mostly in the context of species or breed recognition. Many authors have reported bird species identification results on the Caltech-UCSD Birds Dataset [27], using the idea of parts in one way or another. [8, 32, 33, 35] in different ways attempt to find parts of the image that are discriminative without explicit part labels, but cannot achieve the accuracy of a supervised part-based approach. [4, 26] propose interactive approaches which include the system requesting the location of the most discriminative parts from the user. [9] defines a set of just

two coarse parts (the head and body) used to align the images, but do not use fine-scale part locations to define their features. Additional fine-grained recognition work has been demonstrated on trees [13], flowers [19], butterflies [28, 8], and dogs [16, 21, 23].

Although much recognition work continues to use fixed features such as SIFT, HOG, and LBP, there is also work which, like ours, attempts to some degree to learn the feature from a dataset. One branch of this is the work on attributes. [14] and [15] train attribute classifiers based on a set of classes with labeled attributes, then apply the attribute classifiers to novel classes, in the domains of faces and animal species respectively. In both cases, unlike our POOFs, the attributes are chosen by hand. [8] comes closer to our work, automatically finding discriminative attributes, and their support regions, based on a hierarchical oversegmentation of the images. However, by deriving the regions from segmentation, they seem to only find attributes that correspond to single-color regions of the images.

Outside the realm of fine-grained categorization, there is some work in learning discriminative features. [11] applies linear discriminant analysis to remove correlation between HOG features, improving their accuracy. Brown *et al*. [5] describe a formal optimization method for learning a parametrized descriptor based on a set of matching patches, and a convex formulation of this method is presented in [24]. These methods restrict the shape of the feature support region to one of several symmetric configurations, while our method allows any shape of descriptor, up to the resolution of our base feature grid.

3. Part-Based One-vs-One Features

Our method requires as input a reference dataset of images belonging to the domain under study, annotated with class labels and part locations. It is not necessary that all parts be labeled in all images. The output of our method is a set of discriminative features we call *Part-based One-vs-One Features*, suitable for many tasks in this domain. If the task at hand is supervised classification, the reference dataset may simply be the training set, but it need not be. It can also be a separate dataset labeled with classes different from those in the classification task. We show examples of this in Sections 4.2 and 4.3.

Given the reference set, the process of POOF learning is fully automatic. The method is illustrated in Figure 1, and is motivated overall by the goal of building a *discriminative* and *diverse* set of features. Let the reference set consist of images in N classes $\{1, ..., N\}$, each image labeled with P parts. Each POOF we will learn is defined by

- the selection of two distinct reference classes, $i, j \in \{1, ..., N\}$ with $i \neq j$,
- one part for feature extraction, $f \in \{1, ...P\}$,
- one other part for alignment, $a \in \{1, ..., P\}$, with $a \neq f$, and
- a low-level *base feature*, b, which can be extracted from windows in the image. In the current implementation we use two base features: gradient direction histograms and color histograms.

We write $T^{i,j}_{f,a,b}$ for the POOF built based on these parameters; it is a function that extracts a single, scalar score from any image in the domain, and in combination the $T^{i,j}_{f,a,b}$ form a powerful feature space. We learn how to extract $T^{i,j}_{f,a,b}$ by the following procedure.

1. The POOF will be learned based on the reference images of classes i and j. We first take all these images, exclude those in which either part f or part a is missing, and perform a similarity transform to bring points f and a to fixed positions. The transformed image is then cropped to a rectangular region enclosing points f and a. Depending on whether points f and a are close to or far from each other on images in this domain, $T^{i,j}_{f,a,b}$ will learn a fine-scale or coarse-scale feature.
2. We tile the cropped images with a grid of *feature cells*, and extract the base feature from each cell. We do multiple tilings, each using grid cells of a different size, and so extracting features at a different scale.
3. For the tiling at each scale, we train a linear support vector machine to distinguish class i from class j, based on the concatenation of the base feature values over the grid.
4. The trained SVM weight vector gives weights to every dimension of the base feature in every grid cell. We assign to each grid cell in each tiling the maximum absolute SVM weight over the dimensions in the feature vector that correspond to that cell. By thresholding these weights, we obtain a mask on the aligned images that defines the grid cells that are most discriminative between class i and j.
5. Starting with the grid cell containing part f as a seed, we find the maximum connected component of grid cells above the threshold in each tiling. This will act as a mask on the aligned image, defining at each scale a discriminative region around part f. By restricting the region to a connected component of f, we force POOFs with different feature parts to use different regions, encouraging diversity across the set of POOFs.
6. The low-level feature associated with $T^{i,j}_{f,a,b}$ is the concatenation of the base feature at the masked cells in all the tilings. Using this feature and all aligned images of classes i and j, we train another linear SVM. This SVM learns a projection of the masked, multiscale, local feature to a single dimension. This projection is $T^{i,j}_{f,a,b}$.

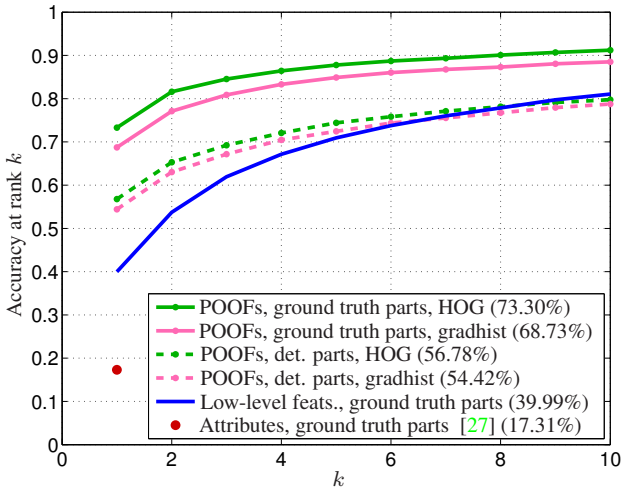

Figure 2. Bird species classification accuracy on the full 200-species CUBS benchmark.

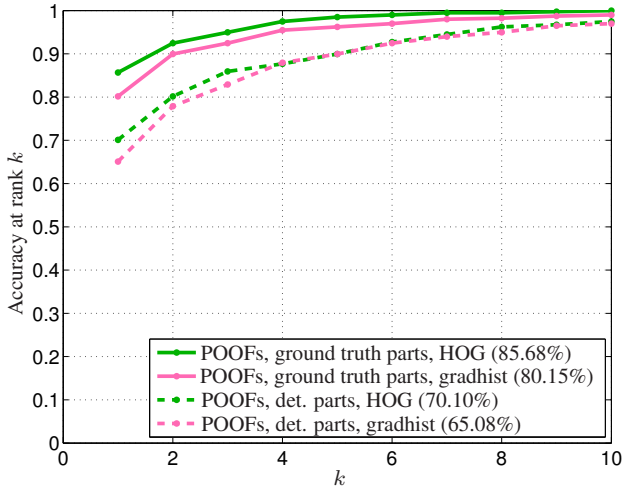

Figure 3. Bird species classification accuracy on the "birdlets" subset of 14 woodpeckers and vireos defined in [9].

To extract feature $T^{i,j}_{f,a,b}$ from a new image with part locations, we proceed through the steps above again. The new image is aligned by similarity to put parts f and a in standard locations, then the base-level feature is extracted from just the masked cells of the tilings at each scale. The resulting vector is evaluated by the SVM to get a scalar projection value, which is the POOF score.

Note that switching i and j simply reverses the sign of the feature (i is taken as the "positive" class when training the SVMs). To avoid redundancy, we restrict ourselves to $i < j$. In contrast, parts f and a play different roles in constructing the POOF, so it may be useful to have both $T^{i,j}_{1,2,b}$ and $T^{i,j}_{2,1,b}$.

3.1. Implementation details

In our current implementation, we use the following settings.

- In the alignment, the two parts are placed in a horizontal line with 64 pixels between them. The crop is centered at the midpoint of the two parts, and is 64 pixels tall and 128 pixels wide.
- We use two scales of grid for the base feature extraction, with 8 x 8 and 16 x 16-pixel cells.
- We use two base features. The first is a gradient direction histogram. This feature comes in two variants. For the "gradhist" variant, we extract an 8-bin gradient direction histogram from each grid cell, then concatenate the histograms over all cells (or in the final $T^{i,j}_{f,a,b}$, over just the masked cells). For the "HOG" variant, we use Dalal and Triggs' histogram of oriented gradients [7] feature, as modified by Felzenszwalb *et al.* [10] to include a dimensionality reduction step and the concatenation of histograms of signed and unsigned gradient. This gives us a nine-bin unsigned gradient direction histogram, an 18-bin signed gradient orientation histogram, and 4 normalization constants, for, in total, a 31-dimensional feature for each grid cell. These are concatenated as in the gradhist variant.
  The second base feature is a color histogram. We use the same grids as for the gradient direction histograms, assigning each pixel to one of 32 color centers to form a histogram of length 32. The color histograms are then concatenated as with the gradient orientation histograms. The color centers are obtained by running k-means in RGB space on the pixels in the aligned and cropped region for all the images in the reference set, so the color centers are a function of f and a.
- For the SVM weight threshold we use the median absolute weight. This has the effect of masking out half of the region in Step 4 (which is further reduced when we restrict the region to a connected component contiguous with part f).

4. Experiments

To demonstrate the value and applicability of POOFs, we apply them to three problems. In Section 4.1, we consider bird species identification, building a set of POOFs from the training set, and applying them to recognition. In Section 4.2 we apply our method to face verification on unseen face pairs, building POOFs on a set of faces of different people than the test faces, demonstrating that our features learn to discriminate over the domain of images in general and not just over the particular classes from which they are built. In Section 4.3, we apply the POOFs built in Section 4.2 to attribute classification, and find that they are useful even when the classification task is on a different type

of classes (attributes) than the classes on which they were learned (subject identities).

4.1. Bird Species Identification

The Caltech-UCSD Birds 200-2011 dataset [27] contains 11,788 photographs of birds spanning 200 species. Each image is labeled with its species, a bounding box for the bird, and the image coordinates of fifteen parts: the back, beak, belly, breast, crown, forehead, left eye, left leg, left wing, nape, right eye, right leg, right wing, tail, and throat. The images are split into training and test sets, with about 30 images per species in the training set, and the remainder in the test set. The authors propose several benchmarks for species recognition and part detection. Here, we evaluate on the "localized species categorization" benchmark, in which the part locations for all images are provided to the algorithm, and the task is, given the species labels on the training images, to determine the species of the test images. We also include results using an automatic parts detector in place of the ground truth positions.

There are very few images in the dataset with all fifteen parts visible. In particular, most birds have only one eye and one wing visible. When a part is not visible, it is labeled as such, with no position given. To better be able to make correspondences between parts, we preprocess the images, performing a left-right reflection on any image in which the right eye is visible but the left is not. This gives us a dataset in which almost all of the images have the left eye labeled (a few images have neither eye visible). We then disregard the (usually missing) right eye, right wing, and right leg parts.

To apply POOFs to this problem, we take the training set as our reference set. There are 200 classes, twelve parts, and two base features, yielding $\binom{200}{2} \cdot 12 \cdot 11 \cdot 2 = 5,253,600$ possibilities if we exhaustively learn features for all (i, j, f, a, b). Instead, we randomly choose 5000 sets of parameters and learn just those features. We then extract the POOF scores from the training and test images, obtaining a feature vector of length 5000 for each image. Using this feature, we train a set of 200 one-vs-all linear SVMs to classify species. For each image, we rank the 200 species from highest to lowest classifier response. Taking the top ranked species for each image, we achieve a classification accuracy of 68.73% using the gradhist variant of the gradient feature, or 73.30% using the HOG variant.

While the localized species categorization protocol defined in [27] uses the ground truth part locations, this does not give *automatic* classification performance. To evaluate automatic classification, we rerun the experiment using automatically detected part locations on the test data in place of the ground truth locations. We use part locations from the part detector of [2] on images cropped to the bounding boxes of the birds, allowing us to compare with previous work that uses the bounding boxes but not the part labels.

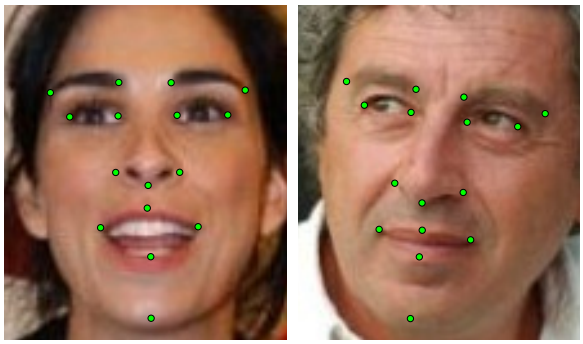

Figure 4. Face parts from the detector of [2].

Using these detected part locations, we obtain a classification accuracy of 54.42% with the gradhist variant or 56.78% with HOG. The rate at which the correct species is in the top k ranked species is shown in Figure 2. For comparison with existing work, we also show our results when restricted to the 14-species "birdlets" subset of the dataset defined in [9] in Figure 3. Our rank-1 classification accuracy on this subset using the gradhist variant is 80.15% using the ground truth parts and 65.08% using the detected parts, or 85.68% and 70.10% using HOG.

To show the benefit of the POOFs, we contrast our one-vs-all species classifiers with classifiers trained in a similar way, but without the POOFs. The POOFs are built using histograms of gradient direction and color over spatial grids covering the parts as the base features. For comparison, we build species classifiers that operate directly on the concatenation of these base features over all twelve parts. As with the POOF-based species classifiers, these classifiers are linear SVMs. These classifiers achieve a rank-1 accuracy of 39.99%.

Baseline accuracy on the localized species categorization benchmark reported in [27] is 17.31%, barely a quarter of our accuracy. To our knowledge, ours is the first subsequent work strictly following this protocol. However there are several pieces of work on this dataset reporting results of different experiments with which we can make comparisons.

Our result of 56.78% based on automatically detected parts uses only the ground truth bounding boxes, as does all the previous work cited here, and is far higher than any existing results on the full 200-species dataset, although there are differences in the experiments that make some of the comparisons imperfect. [4] and [33] report rank-1 accuracies, of 19% and 19.2% using multiple kernel learning and random forests respectively. However they use an earlier version of the dataset [29] with less training data. [32] reports 44.73% mean average precision on the birdlets subset using the earlier version of the dataset (our mAP with HOG on the birdlets subset is 85.57% using ground truth parts or 70.16% using detected parts). Only [8] and [35]

report on the later version of the dataset. The former does not include results on the full 200-species set or the known birdlets subset, however the highest accuracy they report is 55%, on a five-species subset, very close to our automatic result on the much more difficult 200-species set. The latter is the most directly comparable to our work, reporting mean average precision of 28.18% on the 200-species benchmark and 57.44% on the birdlets subset. Our comparable mean average precisions with HOG are 56.89% and 70.16% respectively.

4.2. Face Verification

In face verification, we are given two face images, of people not encountered at any training stage, and must determine whether they are two images of the same person or images of two different people. Because we must deal with previously unseen faces, there is no training set of images belonging to the classes we will be faced with at test time, as there was in the previous example, where we could learn our features based on the training set. Here, we learn the features from a set of face images entirely separate from the evaluation dataset, in the belief that the features we discover are generally applicable to the face domain.

The Labeled Faces in the Wild (LFW) [12] is the standard face verification dataset and benchmark, containing 6,000 face pairs and a ten-fold cross-validation protocol for algorithm evaluation. The best existing result on the "image-restricted" benchmark is 93.30% [3], using a separate reference dataset of images labeled with identity to train a set of "Tom-vs-Pete" classifiers, which are then used as feature extractors feeding a higher level classifier. We use this same reference dataset to learn POOFs, and also transform the images with the "identity-preserving alignment" from that work as a preprocessing step, based on part detections from the detector of [2].

The reference dataset, from [3], consists of 20,639 face images, downloaded from the internet, spanning 120 subject identities. The images are annotated with the location of 95 parts, a fairly dense representation that is useful for alignment, but unnecessary for learning the our features; we use only the sixteen-part subset shown in Figure 4. We learn a random selection of 10,000 POOFs from this dataset, following the steps in Section 3 without modification.

To apply POOFs to the verification problem, we follow the method of [3]. For each verification pair (I, J), we get 10,000-dimensional POOF score vectors $f(I)$, $f(J)$. We then represent the pair by the concatenation of $|f(I)-f(J)|$ and $f(I)\cdot f(J)$ (where the subtraction and multiplication are performed elementwise) to get a 20,000-dimensional pair feature vector. This image pair feature is extracted from the training folds to train a same-vs-different classifier that makes the verification decision.

We obtain an accuracy of 93.13%, with a standard deviation of 0.40%across the ten folds using the gradhist variant, or 92.80%±0.47% using HOG. Our method shares a great deal with the state-of-the-art method of [3], and obtains very similar results. The most important difference is that our method is general, where they carefully choose the support regions for the Tom-vs-Pete classifiers based on knowledge of face recognition. Our method is also more efficient at test time, using a linear rather than an RBF kernel. Our ROC curve is shown in Figure 5, with the four best published results on this benchmark. Figure 6 compares the result from the POOFs with a result using the base features alone, showing, as in Figure 2 for bird species recognition, a substantial boost due to the POOFs.

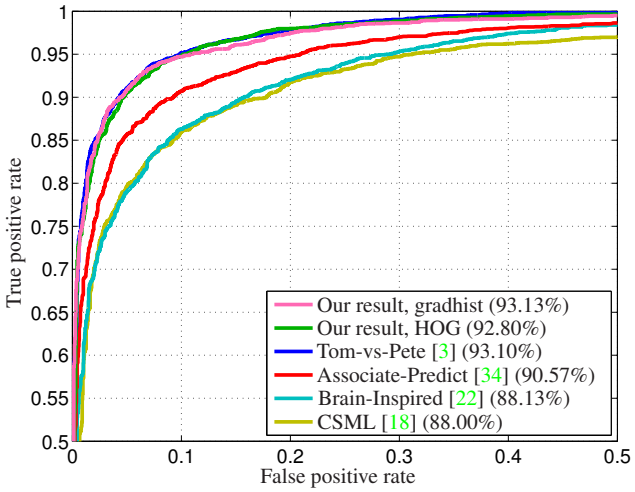

Figure 5. Results on the LFW benchmark. Our result, and the top four previous results.

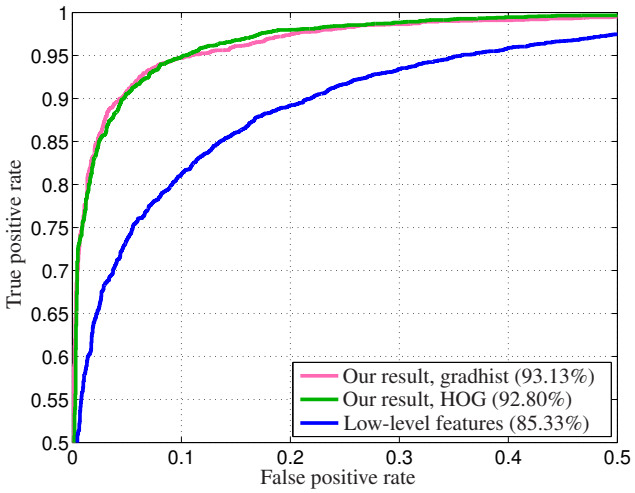

Figure 6. Comparison of POOFs with low-level features on the LFW benchmark.

Attribute	Method	Number of training samples 6	20	60	200	600	Kumar *et al.*
Gender	low-level feat.	50.7	61.0	66.9	81.4	87.8	90.5
	POOFs	**86.2**	**89.9**	**89.7**	**91.3**	**91.7**	
Asian	low-level feat.	53.9	53.9	68.4	78.2	83.2	86.5
	POOFs	**75.2**	**75.8**	**84.3**	**87.6**	**89.8**	
White	low-level feat.	57.0	57.4	68.3	76.7	77.7	**85.5**
	POOFs	**66.3**	**74.9**	**82.6**	**81.7**	**80.5**	
Black	low-level feat.	60.9	68.3	76.7	84.1	87.3	75.4
	POOFs	**74.0**	**84.2**	**87.4**	**88.9**	**90.4**	
Youth	low-level feat.	53.6	56.0	59.8	62.5	66.2	66.1
	POOFs	**71.0**	**62.0**	**67.6**	**67.7**	**70.8**	
Middle Aged	low-level feat.	**49.5**	**51.0**	49.6	53.2	56.0	54.2
	POOFs	47.1	50.9	**51.4**	**57.5**	**59.6**	
Senior	low-level feat.	54.6	60.6	63.7	72.1	74.3	69.5
	POOFs	**70.7**	**75.9**	**73.6**	**80.0**	**79.5**	
Black Hair	low-level feat.	50.3	53.6	62.3	67.9	**68.9**	66.0
	POOFs	**54.6**	**59.3**	**62.9**	**67.9**	66.7	
Blond Hair	low-level feat.	53.7	60.7	69.0	**72.3**	74.6	67.6
	POOFs	**70.5**	**68.8**	**72.6**	71.4	**75.2**	
Bald	low-level feat.	54.4	57.3	**65.4**	**68.7**	**70.9**	**71.8**
	POOFs	**55.4**	**62.2**	64.9	66.3	66.9	
No Eyewear	low-level feat.	51.2	56.6	58.8	75.6	79.5	83.9
	POOFs	**65.9**	**76.9**	**75.9**	**85.6**	**87.0**	
Eyeglasses	low-level feat.	51.7	53.9	61.5	71.4	79.4	86.4
	POOFs	**74.5**	**79.3**	**77.2**	**85.6**	**89.5**	
Mustache	low-level feat.	53.3	61.1	69.0	75.2	81.9	83.1
	POOFs	**70.0**	**82.0**	**73.7**	**81.7**	**85.8**	
Receding Hairline	low-level feat.	55.0	56.3	67.0	70.0	**73.6**	**75.7**
	POOFs	**63.7**	**66.4**	**69.3**	**70.5**	71.8	
Bushy Eyebrows	low-level feat.	49.9	55.8	63.5	67.4	72.1	71.7
	POOFs	**60.0**	**61.8**	**66.0**	**67.7**	**73.5**	
Arched Eyebrows	low-level feat.	53.2	51.1	54.6	63.3	65.9	66.4
	POOFs	**64.5**	**66.9**	**63.5**	**69.1**	**70.9**	
Big Nose	low-level feat.	52.5	52.5	59.0	63.3	66.6	65.4
	POOFs	**55.2**	**63.6**	**61.5**	**64.9**	**68.3**	
No Beard	low-level feat.	57.1	51.2	62.8	**71.2**	75.9	**80.6**
	POOFs	**71.1**	**68.0**	**68.8**	68.7	**76.7**	
Round Jaw	low-level feat.	50.8	49.5	50.0	53.2	**55.7**	50.5
	POOFs	**51.5**	**53.7**	**54.4**	**55.6**	54.8	
Average improvement		12.3	13.4	8.0	4.3	2.7	2.8

Table 1. Attribute classification accuracy. For each attribute, the first row gives the baseline accuracy obtained by training directly on the low-level base features (color and gradient direction histograms), and the second row gives accuracies using our POOFs. The more accurate of the two is in bold. The last column gives accuracies of the classifiers of Kumar *et al*. [14] on the same test images, in bold when better than the POOFs classifier with 600 training samples. The last row shows the average improvement of the POOFs over the low-level features or [14]. As these are binary attributes, chance gives 50% accuracy.

4.3. Attribute Classification

Our third experiment is attribute classification on human faces. For their work on attributes, Kumar *et al*. [14] downloaded face images from the Internet, labeled them with attributes such as gender, race, age, and hair color, and used these labels to train attribute classifiers based on low-level features such as raw pixel color and gradients. We use this same dataset to train a set of attribute classifiers based on POOFs. Kumar *et al*. have made available both human labels and the results of their attribute classifiers for 19 binary attributes on the 7701 images in View 2 of LFW. Restricting ourselves to these 19 attributes, we use these images as our test set.

Although the classes in this task (attributes) are of a different type from those in the previous experiment (identities), we remain in the face domain, and so expect the POOFs we learned there to be useful here. We use the POOFs learned in Section 4.2 without modification. (This means they are trained using our reference set, not the attributes-labeled images.) To build attribute classifiers, we simply extract our 10,000 POOF scores from the attribute training images, and use these feature vectors to train a linear SVM for each attribute. One of the benefits of POOFs is that by incorporating knowledge of the domain learned from the reference set, which is not labeled with attributes, they reduce the need for a large attribute-labeled training set. To demonstrate this, we restrict the number of images we use from the training set.

The results on the test set are shown in Table 1, using the gradhist variant of the gradient orientation base feature. As before, we also show the performance of classifiers built directly on the low-level base features. In almost every case our POOFs outperform the classifier operating directly on the low-level features. The difference is especially large when the amount of training data is small. At six training samples, many of the direct classifiers are at chance accuracy (*e.g*. gender) or even worse; it is easy for the classifier to attach significance to a random peculiarity of the six images it sees. Our POOFs, based on what they have learned is discriminative in a different set of classes (identities) in the same type of image (faces), avoid this noise. The table also shows the results of the classifiers of [14] on this dataset. These classifiers are trained on between 1500 and 5600 samples each. To account for biases in the dataset, the accuracies we report are the means of the accuracies on positive and negative test images. (For example, the test set is 6% Asian, so a direct calculation of accuracy would give a "never-Asian" classifier 94% accuracy, but our calculation would give it 50%.)

5. Conclusions

We have described a method for building a large set of *Part-based One-vs-One Features* based on a dataset of images labeled by class and with part locations, and shown them to have excellent performance in a series of experiments with different tasks on different datasets, outperforming existing work on the CUBS-200 2011 dataset and equaling the best results on the extensively studied LFW dataset. The strength of the POOFs lies in their ability to bring knowledge from an external, labeled dataset to bear on the problem at hand, even when the labels on the exter-

nal set are of different classes from the dataset under study, and the discriminative power and diversity brought about by training each POOF on a single pair of classes and a single part.

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc. ECCV*, 2006. 1

[2] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Proc. CVPR*, 2011. 5, 6

[3] T. Berg and P. N. Belhumeur. Tom-vs-Pete classifiers and identity-preserving alignment for face verification. In *Proc. British Machine Vision Conf.*, 2012. 1, 2, 6

[4] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *Proc. ECCV*, 2010. 2, 5

[5] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *Trans. PAMI*, 33(1), 2011. 3

[6] R. Brunelli and T. Poggio. Face recognition: features versus templates. *Trans. PAMI*, 15(10), 1993. 2

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 1, 4

[8] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *Proc. CVPR*, 2012. 2, 3, 5

[9] 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 *Proc. ICCV*, 2011. 2, 4, 5

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *Trans. PAMI*, 32(9), 2010. 4

[11] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *Proc. ECCV*, 2012. 3

[12] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, 2007. 6

[13] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. Lopez, and J. V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Proc. ECCV*, 2012. 3

[14] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Describable visual attributes for face verification and image search. *Trans. PAMI*, October 2011. 2, 3, 7

[15] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. CVPR*, 2009. 3

[16] J. Liu, A. Kanazawa, D. W. Jacobs, and P. N. Belhumeur. Dog breed classification using part localization. In *Proc. ECCV*, 2012. 1, 3

[17] D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999. 1

[18] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *Proc. Asian Conf. Computer Vision*, 2011. 6

[19] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proc. Indian Conf. Computer Vision Graphics and Image Processing*, 2008. 3

[20] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Trans. PAMI*, 24(7), 2002. 1

[21] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *Proc. CVPR*, 2012. 3

[22] N. Pinto and D. D. Cox. Beyond Simple Features: A Large-Scale Feature Search Approach to Unconstrained Face Recognition. In *Proc. Conf. Automatic Face and Gesture Recognition*, 2011. 6

[23] P. Prasong and K. Chamnongthai. Face-Recognition-Based dog-Breed classification using size and position of each local part, and pca. In *Proc. Int. Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2012. 1, 3

[24] K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor learning using convex optimisation. In *Proc. ECCV*, 2012. 3

[25] B. Tversky and K. Hemenway. Objects, parts, and categories. *J. Experimental Psychology: General*, 113(2), 1984. 1

[26] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *Proc. ICCV*, 2011. 2

[27] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 4, 5

[28] J. Wang, K. Markert, and M. Everingham. Learning models for object recognition from natural language descriptions. In *Proc. British Machine Vision Conf.*, 2009. 3

[29] 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. 5

[30] L. Wiskott, J.-M. Fellous, and N. K. andChristoph von der Malsburg. Face recognition by elastic bunch graph matching. *Trans. PAMI*, 19(7), jul 1997. 2

[31] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Proc. Asian Conf. Computer Vision*, 2009. 1, 2

[32] B. Yao, G. Bradski, and L. Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *Proc. CVPR*, 2012. 2, 5

[33] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *Proc. CVPR*, 2011. 2, 5

[34] Q. Yin, X. Tang, and J. Sun. An associate-Predict model for face recognition. In *Proc. CVPR*, 2011. 2, 6

[35] N. Zhang, R. Farrell, and T. Darrell. Pose pooling kernels for sub-category recognition. In *Proc. CVPR*, 2012. 2, 5

[36] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4), 2003. 1