A Codebook-Free and Annotation-Free Approach for Fine-Grained Image Categorization

Bangpeng Yao[1] Gary Bradski[2] Li Fei-Fei[1]
[1]Computer Science Department, Stanford University, Stanford, CA
[2]Industrial Perception Inc., Palo Alto, CA
{bangpeng,feifeili}@cs.stanford.edu gary@industrial-perception.com

Abstract

Fine-grained categorization refers to the task of classifying objects that belong to the same basic-level class (e.g. different bird species) and share similar shape or visual appearances. Most of the state-of-the-art basic-level object classification algorithms have difficulties in this challenging problem. One reason for this can be attributed to the popular codebook-based image representation, often resulting in loss of subtle image information that are critical for fine-grained classification. Another way to address this problem is to introduce human annotations of object attributes or key points, a tedious process that is also difficult to generalize to new tasks. In this work, we propose a codebook-free and annotation-free approach for fine-grained image categorization. Instead of using vector-quantized codewords, we obtain an image representation by running a high throughput template matching process using a large number of randomly generated image templates. We then propose a novel bagging-based algorithm to build a final classifier by aggregating a set of discriminative yet largely uncorrelated classifiers. Experimental results show that our method outperforms state-of-the-art classification approaches on the Caltech-UCSD Birds dataset.

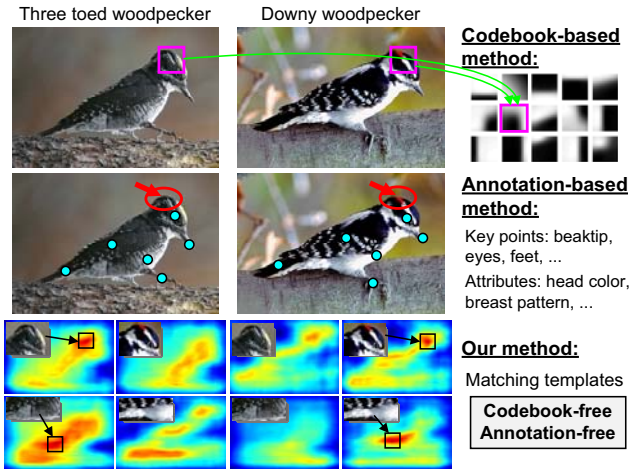


Figure 1. Our approach compared with the codebook and human-annotation based methods. Row 1: Codebook-based methods are likely to encode subtly discriminative features into one codework, resulting in loss of critical information; Row 2: Human-annotation based methods can be tedious and laborious, and might potentially miss key information (e.g. the red arrows point to a discriminative feature on these two different birds); Row 3: Our approach uses an efficient template matching method that is codebook-free and annotation-free. This figure is best viewed in color.

1. Introduction

Fine-grained image classification [1, 13, 3, 31, 24, 10, 27] refers to the task of classifying objects that belong to the same basic-level category. Unlike basic-level categorization, fine-grained classification often poses challenges to even highly knowledgeable humans (e.g. think of picking up edible mushrooms in forests), making it an important computer vision task that can potentially be useful in many real-world applications.

While it is related to generic object classification, fine-grained classification demands an algorithm to discriminate among highly similar object classes that are often differentiated by only subtle differences [23]. Traditional image classification approaches, however, often fail to perform this task satisfactorily [10]. We hypothesize that a key weakness might reside in the way features are encoded in most of today's state-of-the-art image classification systems [7, 17, 28, 31]. Specifically, image patches are often encoded by a universal dictionary of visual codewords built by clustering a large number of image patches. Such procedure, while computationally efficient and effective for generic object categorization, results in a large loss of finer details that are important for differentiating fine-grained object classes (row 1 of Fig.1).

One way of remedying the problem of automatically

constructed codebook is to introduce more human annotation in the learning stage. Instead of generating a universal codebook, several previous work has relied on detailed human labeling to indicate discriminative attributes [27] or key point locations [10] for fine-grained object classification. The results are promising, but the labor cost can be high. It is likely that different fine-grained classification tasks require different queries of attributes and key points, and in many cases domain experts are needed to design these queries. Moreover, while objects like birds and cars can be easily annotated with key point features, many objects in the visual world lack obviously visible key points (e.g. trees, food, etc.).

The above two observations inspire us to propose a new fine-grained object classification framework that is both *codebook free* (i.e. no need of a visual codebook to encode image regions) and *annotation free* (i.e. no need for tedious human annotations of key points) - see row 3 of Fig.1. Our algorithm captures the subtle differences of object classes by directly matching image regions through a highly efficient template matching algorithm inspired by [15, 14]. Images are then represented as feature response maps of these highly throughput feature templates. Finally, using a novel bagging-based algorithm, the algorithm builds a final classifier by aggregating a set of classifiers that all have large discriminative abilities while maintaining low correlations.

The rest of the paper is organized as follows. Sec.2 describes related work. Details of our image representation and classification algorithm are elaborated in Sec.3. Experimental results that show superior performance of our algorithm over existing state-of-the-art methods are given in Sec.4. Sec.5 concludes the paper.

2. Related Work

Fine-grained classification. There is a number of recent work in fine-grained image classification using various datasets such as flowers [22], stonefly larvae [20], bird species [29], and related leaf nodes of ImageNet [9] such as all fungus [8].

Codebook-based approaches. Most of today's state-of-the-art generic image classification systems [7, 17, 28, 31] are based on encoding local image patches to visual codewords, often resulting in coarse encoding of image patches that are lossy in detailed information. Some recent work [28, 33] uses sparse coding to obtain more accurate encodings of image patches, and have demonstrated performance improvement on many image classification tasks. Our algorithm is inspired by these approaches and goes a step further by matching image patches directly, instead of relying on any coding schemes where information loss is difficult to avoid.

Annotation-based approaches. To account for the significant object articulations and subtle distinctions in fine-grained classification problems, a growing body of work seeks to incorporate more inputs from humans, including the humans-in-the-loop approaches [3, 27] that ask humans to click on object parts and answer questions regarding object attributes, and a modified postlet-like [2] algorithm for bird recognition [10] that achieved good performances on the Caltech-UCSD Birds Dataset [29]. But as discussed in Sec.1, these algorithms require tedious human annotations of object attributes or key point locations, posing serious challenges to the situations where fully automatic learning is desired. Furthermore, it is costly to adapt these approaches to new fine-grained object classes, because the attribute queries and object key points usually need to be carefully designed by domain experts, especially for the objects without obvious key parts or distinct attributes (e.g. trees, food, etc.).

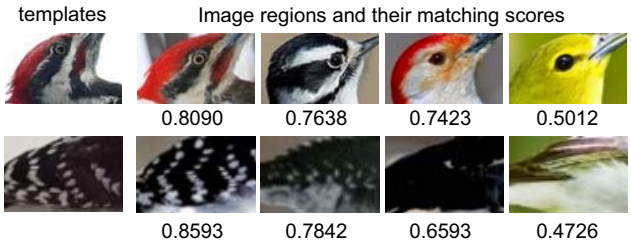


Figure 2. Template matching captures the subtle distinctions between image patches. The figure shows two templates and the scores of matching each of them to four different regions. The more similar the template and the image region are (in terms of color, texture, etc.), the larger their matching score.

Template-based approaches. Our algorithm obtains a feature response-map of matching an image with a large number of randomly generated image templates. This method is in spirit similar to a number of recent work that uses various pre-defined filters to generate image response features, such as object detectors [18, 25], human body part detectors [19], and cluster centers of image regions [6]. While these approaches have shown promising results in basic-level object and scene classification tasks, they still suffer from the issues of coarse codebook encoding ([6] where the best results are obtained by clustering image regions) or tedious human annotation ([18, 25, 19]). Furthermore, none of these representations has been tested on fine-grained classification.

Other object classification work. It is beyond the scope of this paper to discuss the large body of object classification work, mostly centered around basic-object categorization. Some seminal work such as part-based models (e.g. [12, 30, 11]) perform well in classifying objects of minimal articulation or localizing objects from the background. However, it is unclear how to use these methods for distinguishing fine-grained objects that share similar visual appearance and exhibit significant articulations.

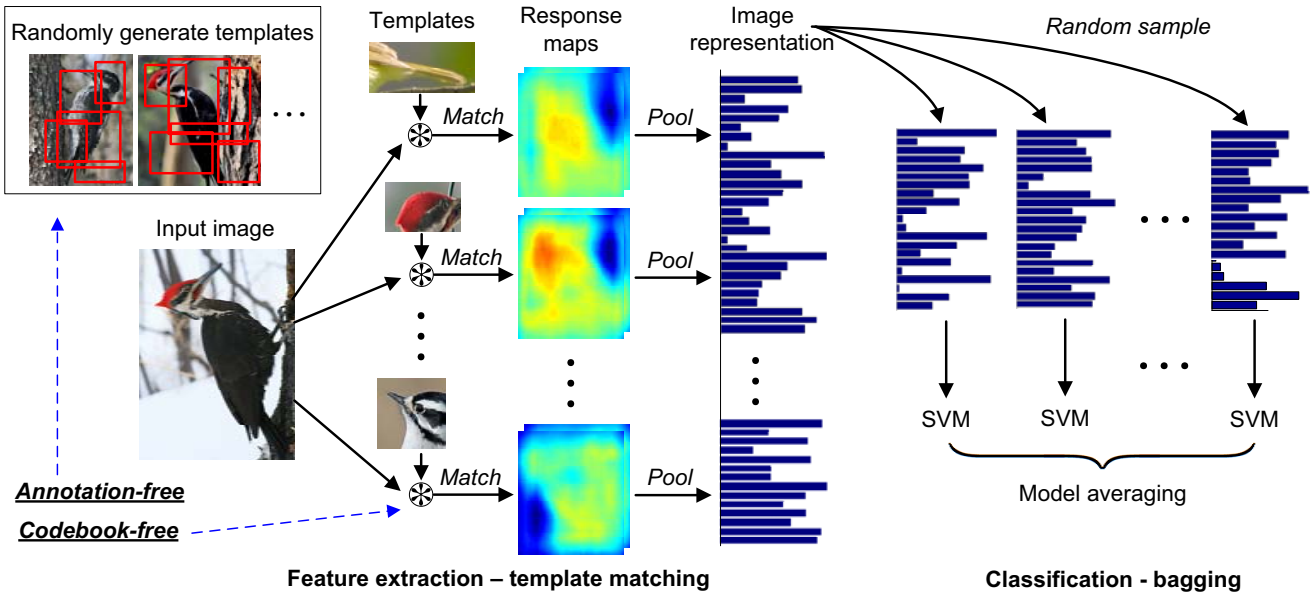


Figure 3. An overview of our fine-grained object classification algorithm. From the set of training images, we randomly generate a large number of image templates (denoted by red rectangles in the top-left part of this figure). An image is represented by pooling the response maps that are obtained from matching the image with a large set of randomly generated image templates. A bagging algorithm is then used to aggregate a set of classifiers that are trained on this representation.

3. Our Approach

An overview of our algorithm is shown in Fig.3. We match an image at various locations with a large number of randomly generated image templates, and use the pooling result on the response maps for image representation (Sec.3.1). To use this feature for classification, we propose a bagging-based algorithm that aggregates a set of discriminative classifiers that are guaranteed to have small correlations (Sec.3.2).

3.1. Template Matching Based Representation

Intuition. Here we describe our template matching based image representation. We generate a large number of templates by randomly sampling rectangular regions from all the training images, as shown in Fig.3. Then an image is represented by the response scores of matching itself with each of the templates. Using continuous template matching scores instead of discrete visual codewords for image representation (*codebook-free*), our method is able to capture the subtle distinction between similar image patches (as shown in Fig.2), which is important for fine-grained classification. Furthermore, our approach does not require tedious human annotations (*annotation-free*), since all the templates are directly generated from training images.

Templates and the Matching Approach. We use a highly optimized template matching algorithm [15] that is also resistant to small image deformations. For each training image $\mathcal{O}$, we denote its appearance represented by S different feature types (e.g. color, gradient, etc.) as $\{\mathcal{O}^s\}_{s=1}^S$. An image template is then denoted as $\mathcal{T} = \left(\{\mathcal{O}^s\}_{s=1}^S, \{r, s\}\right)$, where each pair of r and s indicates the image feature in location r of $\mathcal{O}^s$.

To obtain the feature representation for an image, we first resize it with different scale factors. We then match each rescaled image $\mathcal{I}$ with each template. The similarity between template $\mathcal{T}$ and the image region centered at location c in image $\mathcal{I}$ is computed by

$$\sum_{(r,s)} \left(\max_{c' \in \mathcal{R}(c+r)} f_s(\mathcal{O}^s(r), \mathcal{I}^s(c')) \right) \quad (1)$$

where $\mathcal{R}(c+r)$ is a small image neighborhood centered on location $c+r$ of image $\mathcal{I}$, designed to accommodate small image deformation and noise distortions. $\mathcal{O}^s(r)$ is the type-s feature value in location r of image $\mathcal{O}$, and $\mathcal{I}^s(c')$ is defined in the same way. $f_s(\mathcal{O}^s(r), \mathcal{I}^s(c'))$ measures the similarity between $\mathcal{O}^s(r)$ and $\mathcal{I}^s(c')$. Please refer to [15] and [14] for details of how to compute f_s.

To deal with object scale variations, we consider different scales of each template. The template matching step produces a map of response scores for each template on each scale. We use max-pooling on a two-level spatial pyramid to transform this response map to a feature vector, as shown in Fig.4. On level 1, we take the three-largest response scores with a constraint that their mutual distance should be no less than $0.1\times$ the map's width and height.

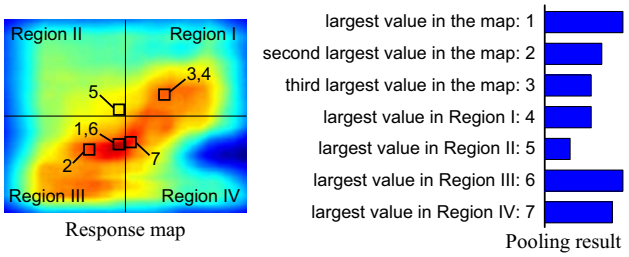


Figure 4. An illustrating our feature pooling approach. The image on the left represent a response map and the bar figure on the right is its pooling result. When we compute the three largest response scores, we make sure that their distance should be no less than $0.1\times$ the map's width and height.

On level 2, we take the largest score on each image region. This gives us a 7-dimensional vector on each scale of each template.

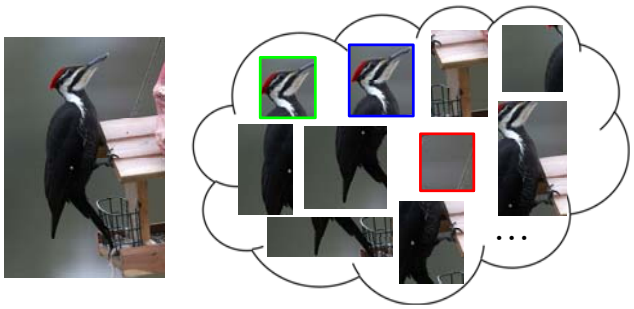
The final image representation is formed by concatenating the pooling results of all the templates on all the image scales, resulting in a `No.Templates` $\times$ `No.Scales` $\times$ 7 dimensional feature value.

Implementation Details. In our current framework, we consider two different image features, color and gradients. The color descriptor for each pixel is a $2^3 = 8$ bit binary value obtained by binning the R, G, and B channels, respectively. The gradient descriptor for each pixel is obtained by quantizing the gradient angle to 8 bins. To be more robust to noise, for each pixel location, we choose the most frequent value of color or gradients within a small neighborhood of 3×3 pixels. Using the highly optimized matching technique proposed in [15, 14], we can match each 200×300 image with tens of thousands of templates within one second on a standard CPU. Please refer to [15, 14] for more details of the matching approach.

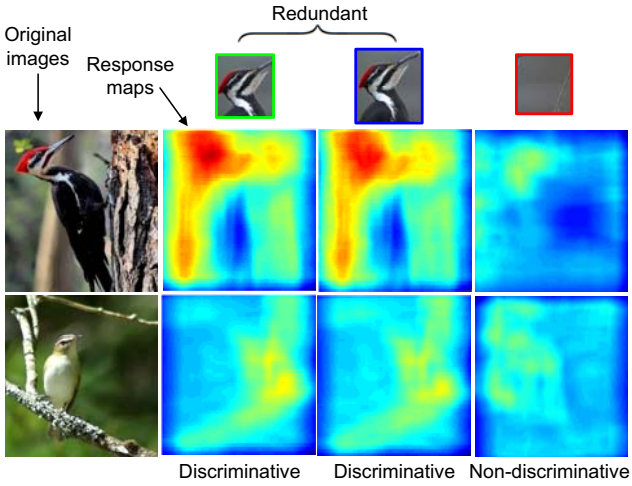
3.2. Bagging Based Classification Algorithm

Motivation. In the image representation discussed in Sec.3.1, our algorithm uses a large number of image templates, resulting in a very high-dimensional feature vector for each image. Using a larger number of templates is advantageous because they can provide a richer image representation and capture more subtle visual distinctions between different fine-grained object classes. Examples of possible templates that are sampled from an image are shown in Fig.5(a).

However, this feature vector is over-complete (many of the templates overlap in locations) and contain non-discriminative elements (some templates might be sampled from uninformative regions). Examples of redundant and non-discriminative templates are shown in Fig.5(b). Therefore, conventional classification algorithms such as training a single SVM are likely to suffer from the overfitting problem on this feature representation.



(a) Examples of templates sampled from a single image. One template can correspond to any location of the image and can be of any size.



(b) Response maps from different templates.

Figure 5. **(a)** From an image, we can generate a large number of templates by randomly sampling image regions. **(b)** Illustration of discriminative templates, redundant templates, and non-discriminative templates. Red color in the response maps indicates large matching score, while blue indicates small matching score.

Formulation of the Algorithm. We propose to train a set of SVM classifiers on this feature representation, and aggregate the results of all the classifiers, as shown in Fig.3. Our method guarantees that the correlation between the classifiers are small, so that aggregating them together will achieve better performance [4, 5]. By combining multiple classifiers, our method tries to make a full usage of all available template matching results, which is more desirable than using feature reduction algorithms such as PCA, as shown in [21]. Furthermore, instead of using the traditional bagging method which randomly selects elements from image feature vectors, we directly minimize the classification performance of bagging with respect to a set of constraints that guarantee the weak correlation between all the classifiers.

Our novel classification algorithm is formulated as follows. The set of training images are represented by $\{\mathbf{x}_i, y_i\}_{i=1}^N$ where $\mathbf{x}_i$ is the feature representation for the

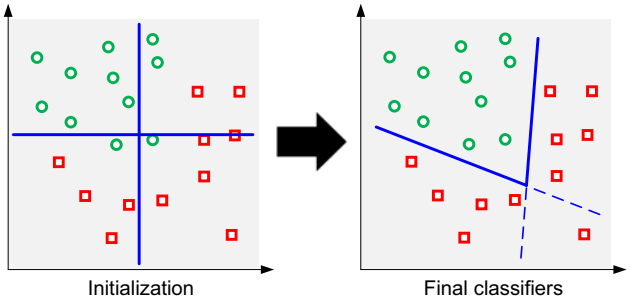


Figure 6. A 2D illustration of our optimization approach. In the initialization stage, we learn classifiers on disjoint sets of features (in the 2D case, one classifier per feature element). We then alternatively update all the classifiers to maximize their joint discriminative ability while maintaining low correlations. Note that in our problem, the feature dimension is very high.

i-th image and y_i is the class label. Denoting that our bagging algorithm contains P classifiers, each corresponds to a set of feature weights $\mathbf{w}_p$, our learning objective is

$$\min_{\mathbf{w}_1,\cdots,\mathbf{w}_P} \sum_{p=1}^{P} \left(\sum_{i=1}^{N} \ell\left(\mathbf{w}_p \cdot \mathbf{x}_i, y_i\right) + C\|\mathbf{w}_p\|_1 \right) \qquad (2)$$
$$s.t.\ \ \forall p \neq q,\ |\mathbf{w}_p \cdot \mathbf{w}_q| < T$$

where $\ell(\cdot,\cdot)$ is a loss function, C and T are the parameters that control the regularization of $\mathbf{w}$ and the correlation tolerance between all the classifiers. From Eq.2, we can see that our method maximizes the discriminative ability of all the classifiers (in the objective function) while maintaining low correlation among them (due to the constraint).

During the testing stage, the classification result of an image represented by $\mathbf{x}$ is the average of the confidence scores obtained by all the P classifiers.

Optimization. Eq.2 is not a globally convex problem. We use the Convex-Concave Produce (CCCP) [32] for optimization, alternating among different $\mathbf{w}_p$'s. In the initialization stage, we randomly partition the whole image feature vector to P disjoint blocks of equal length, each corresponds to the matching results obtained from $\frac{M}{P}$ templates where M is the number of templates. For each block, we train an SVM classifier which results to a weight vector that has non-zero values on the features of the corresponding blocks. Therefore after initialization, we have P sets of feature weights $\mathbf{w}_p^{(0)}$ that are mutually orthogonal.

We then alternatively optimize each $\mathbf{w}_p$. In step t when we optimize $\mathbf{w}_p$, the objective function becomes

$$\mathbf{w}_p^{(t)} = \arg\min_{\mathbf{w}_p} \left(\sum_{i=1}^{N} \ell(\mathbf{w}_p \cdot \mathbf{x}_i, y_i) + C\|\mathbf{w}_p\|_1 + \text{const} \right)$$
$$s.t. \quad |\mathbf{w}_p \cdot \mathbf{w}_q^{(t)}| < T, \quad q = 1, \cdots, p-1$$
$$|\mathbf{w}_p \cdot \mathbf{w}_r^{(t-1)}| < T, \quad r = p+1, \cdots, P \qquad (3)$$

Eq.3 is convex, which is solved by an interior point algorithm.

4. Experiments and Analysis

4.1. Dataset and Experiment Setup

We test the performance of our algorithm on the Caltech-UCSD Birds dataset (CUB-200) [29] which is widely used for evaluating fine-grained image classification tasks [3, 31, 27, 10]. The CUB-200 dataset contains photos of 200 bird species. In each class, there are 15 training images and 10 to 25 testing images. Using the same setting as in [10], we consider a fine-grained classification task on 14 bird species from the vireos and woodpeckers families. We use the training split provided by [29], where there are a total of 420 training images (original plus their left-right mirrored images) and 492 testing images. Following a standard image normalization procedure on this dataset similar to [31], all the images are cropped to be centered on the locations of the birds and contain $1.5\times$ the size of the provided bounding boxes, and re-sized such that the smaller dimension is 150 pixels.

In our implementation, we generate 100 templates from each training image. The location, width, and height of all the templates are randomly sampled from uniform distributions, with the only constraint that the width (or height) of the template should not be smaller than $0.1\times$ or larger than $0.4\times$ the width (or height) of the image. In the template matching stage, for each image we not only consider its original size, but also resize it with the scaling factors of 0.75 and 1.33. Therefore the overall length of the feature vector for each image is $420 \times 100 \times 3 \times 7 = 882000$. We then use the method described in Sec.3 for classification, where the bagging repetition number (P value in Eq.2) is set to 80.

4.2. Results and Analysis

The performance of our method, denoted as Ours-full in Tbl.1, is 44.73% mean-average precision, whereas the performance of training a single SVM classifier on the same

Classification method		mAP
Codebook-based	cSIFT [26]+SPM [17]	37.12%
	MKL [3]	37.02%
Annotation-based	Birdlet [10]	40.25%
Our method	Ours-SVM	39.76%
	Ours-Full	**44.73%**

Table 1. Classification results on the CUB Birds dataset. The performance is measured by mean average precision (mAP). The best result is marked by bold font. We compare our method with both codebook-based and annotation-based approaches. Our full method (Ours-Full) achieves the best performance.

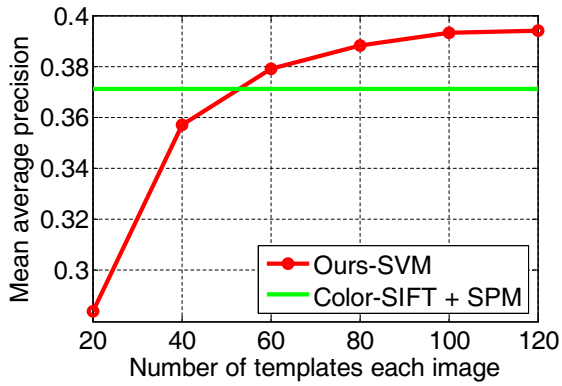


Figure 7. Ours-SVM classification performance with respect to the number of templates generated from each training image. Color-SIFT+SPM is used as a baseline method.

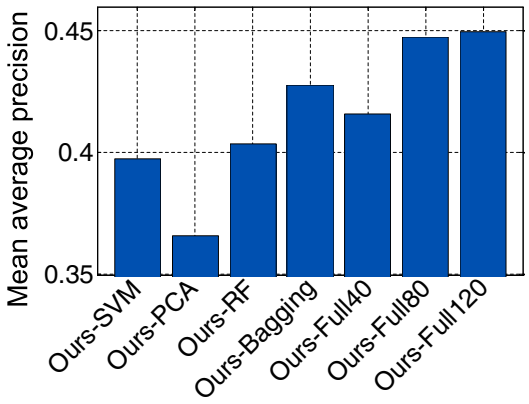


Figure 8. Evaluation of our full method as compared to various baselines using the same image representation, as well as different parameters in our method.

feature (Ours-SVM in Tbl.1) is 39.76%.

Analysis of "codebook-free". We first compare our method with the approaches that are based on encoding local image features to codewords. We consider two such algorithms that have been applied to this dataset: color-SIFT (cSIFT) [26] for image representation and spatial pyramid matching (SPM) [17] for classification, and multiple-kernel learning [3]. Results of the two methods reported in [10] are shown in Tbl.1. We observe that even Ours-SVM outperforms both colorSIFT+SPM and MKL, demonstrating the advantage of template matching over codeword encoding on this task.

Analysis of "annotation-free". Tbl.1 also compares our method with the "Birdlet" approach [10], which normalizes the visual features by localizing different parts of birds. Based on the key points (such as beaktip, eyes, etc.) annotated by humans, the parts are obtained by training a set of part detectors. Tbl.1 shows that our bagging-based method outperforms Birdlet (44.73% vs. 40.25%). We achieve this performance gain without using any additional key points annotation.

Analysis of the bagging-based classification algorithm. To evaluate the effectiveness of our bagging-based algorithm, we compare it with some baseline approaches using the same image representation.

- Ours-SVM: We train a single SVM classifier on the template matching response scores (as in Tbl.1). Here we further generate different number of templates, and evaluate how classification performance changes with respect to the number of templates.
- Ours-PCA: Because the image representation is over-complete, it might intrinsically lie in a lower dimensional feature space. Therefore we use principle component analysis (PCA) [16] for feature dimension reduction. The feature dimension is selected to keep 98% of the total energy.
- Ours-RF: One of the most prominent bagging approaches is to aggregate a set of randomized decision trees, i.e. random forest (RF) [5]. RF has been applied to fine-grained image classification problems before [31]. Similar to [31], on the interior tree nodes we train SVM classifiers based on a set of 500 randomly sampled features rather than randomly generating feature weights. We repeat this procedure for 50 times and select the best classifier for each node.
- Ours-Bagging: In this experiment, we train SVM classifiers based on completely randomly generated feature elements, without considering the correlation between these classifiers as in Eq.2.
- Ours-Full: The full algorithm described in Sec.6. Here we use three different number of bagging repetitions (40, 80, and 120). The corresponding algorithms are denoted as Ours-Full40, Ours-Full80, and Ours-Full120, respectively.

Fig.7 shows that the classification performance increases when we generate more templates. But the performance gain is very small after having 80 templates for each image. The reason is that the generated templates will largely overlap with each other if the number of templates from each image is too large.

From Fig.8, we observe that the bagging based approaches (Ours-RF, Ours-Bagging, and Ours-Full) generally perform better than the others (Ours-SVM and Ours-PCA) on the over-complete and noisy image representation. Furthermore, in Ours-Full, generating 80 templates from each image is better than using 40 templates, while the difference between using 80 and 120 templates is small. This observation is similar to that on the Ours-SVM method

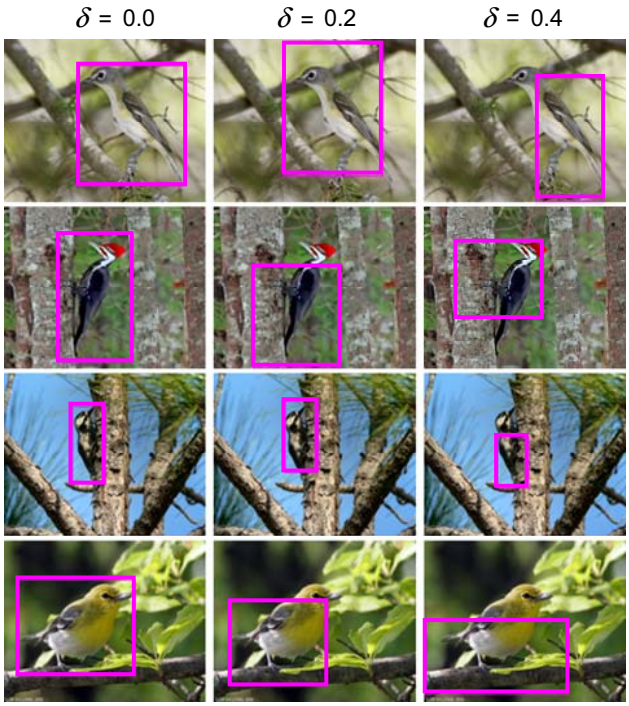


Figure 9. In each row, the bounding boxes are generated based on ground-truth boxes (the first column) and a parameter δ. For each bounding box whose size and image location are (w, h) and (x, y), we generate four parameters $\Delta s_w, \Delta s_h, \Delta l_x, \Delta l_y$ from a uniform distribution $\mathcal{U}[0, \delta]$. We reset the bounding box size and location to $(w + \Delta s_w w, h + \Delta s_h h)$ and $(x + \Delta l_x x, y + \Delta l_y y)$, respectively. We can see that when δ is large, the bounding box can be very different from the ground truth.

shown in Fig.7.

Robustness to non-accurate or missing object locations. In the above experiments, we have made use of the bounding boxes of birds provided in [29] to normalize the images, as in most previous works on this dataset [3, 31, 10]. However, these accurately annotated bounding boxes might not be available in real applications. (Imagine we run a bird detector such as [11] on the dataset.) Here we investigate the performance of our method when the bounding boxes of birds are inaccurate or missing.

We randomly shift and re-scale the bounding boxes. For each image, we randomly generate four values between 0 and a constant $\delta \in [0, 1]$ which determines to what degree we want to shift and re-scale the bounding box along the horizonal and vertical directions, respectively. The larger δ is, the more likely that the bounding box will be different from the ground truth. Fig.9 shows the bounding boxes in some example images where δ is 0, 0.2, and 0.4.

We also consider the situation where the bounding boxes are unavailable. In this case, we need to perform classification based on the whole image without knowing the locations of the birds. To compare our method with existing algorithms, we implement a baseline by using a pyramid histogram on Locality-constrained Linear Coding (LLC) [28] of SIFT and color features. Our baseline achieves similar performance as the color-SIFT + SPM implemented in [10].

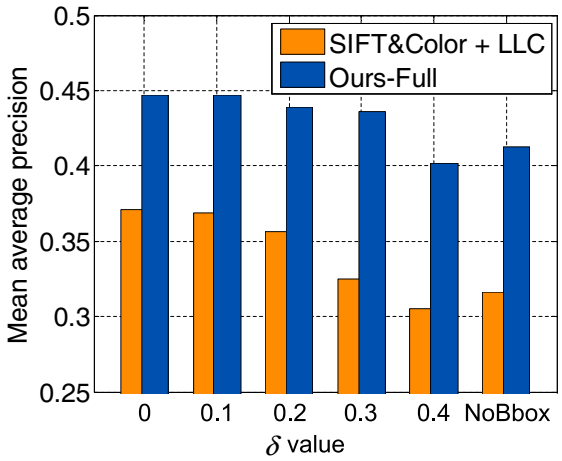


Figure 10. Results of using different bounding box settings (larger δ values indicate that the bounding boxes are more different from their ground truth) or no bounding box.

Fig.10 shows the results of comparing our algorithm and the baseline on different bounding box settings. The results show that the classification performance decreases as the bounding box locations become more and more difference from the ground truth. However we observe that the performance of our method drops by only 1% when $\delta = 0.3$. Even when $\delta = 0.4$ where the bounding boxes can be very different from the ground truth as shown in Fig.9, our method still achieves a 40.27% mean-average precision, which is comparable with the Birdlet method and higher than the codebook based algorithms in Tbl.1. Furthermore, Fig.10 shows that the performance with no bounding box is better than the situation where the bounding box contains many errors.

Speed of our algorithm. Feature extraction is usually the most time consuming part in most image classification systems. In our approach, with the highly optimized template matching approach, we can match tens of thousands of templates to an image, and the class label of an image can be predicted within three seconds on a single standard CPU.

5. Conclusion

In this paper, we propose a codebook free and annotation free approach for fine-grained image classification. We propose to match an image with a large set of templates for image representation, and develop a bagging based algorithm for classification. One direction of future work is to delve deeper into the templates to perform joint image classification and pose normalization.

Acknowledgement

This research is partially supported by an ONR MURI grant, the DARPA CSSG program, an NSF CAREER grant (IIS-0845230), a research sponsorship from Intel to L.F-F., and the SAP Stanford Graduate Fellowship to B.Y. We also would like to thank Industrial Perception Inc and Willow Garage for supporting this work.

References

[1] I. Biederman, S. Subramaniam, M. Bar, P. Kalocsai, and J. Fiser. Subordinate-level object classification reexamined. *Psychol. Res.*, 62(2-3), 1999. 1

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

[3] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010. 1, 2, 5, 6, 7

[4] L. Breiman. Bagging predictors. *Mach. Learn.*, 24:123–140, 1996. 4

[5] L. Breiman. Random forests. *Mach. Learn.*, 45:5–32, 2001. 4, 6

[6] A. Coates, H. Lee, and A. Ng. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011. 2

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on SLCV*, 2004. 1, 2

[8] J. Deng, A. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, 2010. 2

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[10] R. Farrell, O. Oza, N. Zhang, V. Morariu, T. Darrell, and L. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011. 1, 2, 5, 6, 7

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010. 2, 7

[12] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *ICCV*, 2003. 2

[13] A. Hillel and D. Weinshall. Subordinate class recognition using relational object models. In *NIPS*, 2007. 1

[14] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of texture-less objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012. 2, 3, 4

[15] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*, 2011. 2, 3, 4

[16] I. Jolliffe. *Principle Component Analysis*. Springer-Verlag, 1986. 6

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 2, 5, 6

[18] L.-J. Li, H. Su, E. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010. 2

[19] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR*, 2011. 2

[20] G. Martinez-Munoz, W. Zhang, N. Payet, S. Todorovic, N. Larios, A. Yamamuro, D. Lytle, A. Moldenke, E. Mortensen, R. Paasch, L. Shapiro, and T. Dietterich. Dictionary-free categorization of very similar objects via stacked evidence trees. In *CVPR*, 2009. 2

[21] M. Munson and R. Caruana. On feature selection, bias-variance, and bagging. In *ECML PKDD*, 2009. 4

[22] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *CVPR*, 2006. 2

[23] E. Rosch, C. Mervis, W. Gray, D. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Sci.*, 8(3):382–439, 1976. 1

[24] J. Sanchez, F. Perronnin, and Z. Akata. Fisher vectors for fine-grained visual categorization. In *CVPR Workshop on FGVC*, 2011. 1

[25] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010. 2

[26] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1582–1596, 2010. 5, 6

[27] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011. 1, 2, 5

[28] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Learning locality-constrained linear coding for image classification. In *CVPR*, 2010. 1, 2, 7

[29] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD birds 200. Technical report, Caltech, 2010. 2, 5, 7

[30] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR*, 2010. 2

[31] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011. 1, 2, 5, 6, 7

[32] A. Yuille and A. Rangarajan. The concave-convex procedure (CCCP). *Neural Comput.*, 15(4):915–936, 2003. 5

[33] X. Zhou, K. Yu, T. Zhang, and T. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV*, 2010. 2