Combining Randomization and Discrimination for Fine-Grained Image Categorization

Bangpeng Yao* Aditya Khosla* Li Fei-Fei
Computer Science Department, Stanford University, Stanford, CA
{bangpeng,aditya86,feifeili}@cs.stanford.edu

Abstract

In this paper, we study the problem of fine-grained image categorization. The goal of our method is to explore fine image statistics and identify the discriminative image patches for recognition. We achieve this goal by combining two ideas, discriminative feature mining *and* randomization. *Discriminative feature mining allows us to model the detailed information that distinguishes different classes of images, while randomization allows us to handle the huge feature space and prevents over-fitting. We propose a* random forest with discriminative decision trees algorithm*, where every tree node is a discriminative classifier that is trained by combining the information in this node as well as all upstream nodes. Our method is tested on both subordinate categorization and activity recognition datasets. Experimental results show that our method identifies semantically meaningful visual information and outperforms state-of-the-art algorithms on various datasets.*

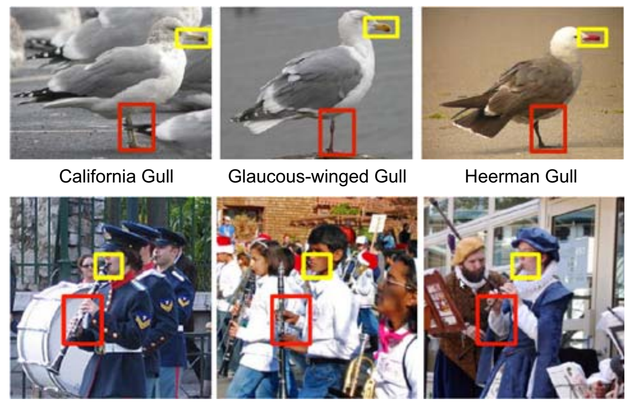


Figure 1. We consider two fine-grained image classification problems: subordinate categorization where fine image parts distinguishes different classes (top row) and human activity recognition where the human body dominates the image region (bottom row). Bounding boxes indicate discriminative image patches.

1. Introduction

Psychologists have shown that the ability of humans to perform basic-level categorization (e.g. cars vs. dogs; kitchen vs. highway) develops well before their ability to perform subordinate-level categorization, or fine-grained visual categorization (e.g. Golden retrievers vs. Labrador) [12]. It is interesting to observe that computer vision research has followed a similar trajectory. Basic-level object and scene recognition has seen great progress [10, 13, 16, 20] while fine-grained categorization has received little attention. Unlike basic-level recognition, even humans might have difficulty with some of the fine-grained categorization [21]. Thus, an automated visual system for this task could be valuable in many applications.

Fig.1 captures the difficulty of dealing with such tasks. The bounding boxes demarcate the distinguishing characteristics between closely related bird species, or different musical instruments or human poses that differentiate the different playing activities. Models and algorithms designed for basic-level object or image categorization tasks are often unprepared to capture such subtle differences among the fine-grained visual classes. In this paper, we approach this problem from the perspective of finding a large number of image patches with arbitrary shapes, sizes, or locations, as well as interactions between pairs of patches that carry discriminative image statistics [6, 22] (Sec.3). However, this approach poses a fundamental challenge: without any feature selection, even a modestly sized image will yield millions or billions of image patches. Furthermore, these patches are highly correlated because many of them overlap significantly. To address these issues, we propose the use of *randomization* that considers a random subset of features at a time.

In this paper, we propose a *random forest with discriminative decision trees* algorithm to discover image patches and pairs of patches that are highly discriminative for fine-grained categorization tasks. Unlike conventional decision trees [3, 1], our algorithm uses strong classifiers at each node and combines information at different depths of the tree to effectively mine a very dense sampling space. Our

*Bangpeng Yao and Aditya Khosla contributed equally to this paper.

method significantly improves the strength of the decision trees in the random forest while still maintaining low correlation between the trees. This allows our method to achieve low generalization error according to the theory of random forest [3].

We evaluate our method on two fine-grained categorization tasks: human activity recognition in still images [22, 7] and subordinate categorization of closely related animal species [21], outperforming state-of-the-art results. Furthermore, our method identifies semantically meaningful image patches that closely match human intuition. Additionally, our method tends to automatically generate a coarse-to-fine structure of discriminative image regions, which parallels the human visual system [4].

The remaining part of this paper is organized as follows: Sec.2 discusses related work. Sec.3 describes our dense feature space and Sec.4 describes our algorithm for mining this space. Experimental results are discussed in Sec.5, and Sec.6 concludes the paper.

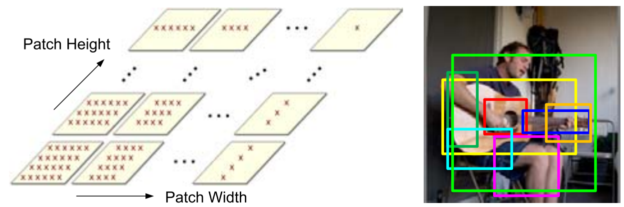
2. Related Work

Image classification has been studied for many years. Most of the existing work focuses on basic-level categorization such as objects [8, 1, 10] or scenes [16, 9, 13]. In this paper we focus on fine-grained image categorization [11, 2], which requires an approach to capture the fine and detailed information in images.

In this work, we explore a dense feature representation to distinguish fine-grained image classes. Our previous work has shown the advantage of dense features ("Grouplet" features [22]) in classifying human activities. Instead of using the generative local features as in Grouplet, here we consider a richer feature space in a discriminative setting where both local and global visual information are fused together. Inspired by [6, 22], our approach also considers pairwise interactions between image regions.

We use a random forest framework to identify discriminative image regions. Random forests have been used successfully in many vision tasks such as object detection [1], segmentation [17] and codebook learning [15]. Inspired from [18], we combine discriminative training and randomization to obtain an effective classifier with good generalizability. Our method differs from [18] in that for each tree node, we train an SVM classifier from one of the randomly sampled image regions, instead of using AdaBoost to combine weak features from a fixed set of regions. This allows us to explore an extremely large feature set efficiently.

A classical image classification framework [20] is *Feature Extraction* $\rightarrow$ *Coding* $\rightarrow$ *Pooling* $\rightarrow$ *Concatenating*. *Feature extraction* [14] and better *coding* and *pooling* methods [20] have been extensively studied for object recognition. In this work, we use discriminative feature mining and randomization to propose a new feature *concatenating* approach, and demonstrate its effectiveness on fine-grained image categorization tasks.



(a) (b)

Figure 2. (a) Illustration of our dense sampling space. We densely sample rectangular image patches with varying widths and heights. The regions are closely located and have significant overlaps. The red $\times$ denote the centers of the patches, and the arrows indicate the increment of the patch width or height. (The actual density of regions considered in our algorithm is significantly higher. This figure has been simplified for visual clarity.) We note that the regions considered by Spatial Pyramid Matching [13] is a very small subset lying along the diagonal of the height-width plane that we consider. (b) Illustration of some image patches that may be discriminative for "playing-guitar". All those patches can be sampled from our dense sampling space.

3. Dense Sampling Space

Our algorithm aims to identify fine image statistics that are useful for fine-grained categorization. For example, in order to classify whether a human is playing a guitar or holding a guitar without playing it, we want to use the image patches below the human face that are closely related to the human-guitar interaction (Fig.2(b)). An algorithm that can reliably locate such regions is expected to achieve high classification accuracy. We achieve this goal by searching over rectangular image patches of arbitrary width, height, and image location. We refer to this extensive set of image regions as the *dense sampling space*, as shown in Fig.2(a). Furthermore, to capture more discriminative distinctions, we consider interactions between pairs of arbitrary patches. The pairwise interactions are modeled by applying concatenation, absolute of difference, or intersection between the feature representations of two image patches.

However, the dense sampling space is very huge. Sampling image patches of size 50×50 in a 400×400 image every four pixels leads to thousands of patches. This increases many-folds when considering regions with arbitrary widths and heights. Further considering pairwise interactions of image patches will effectively lead to trillions of features for each image. In addition, there is much noise and redundancy in this feature set. On the one hand, many image patches are not discriminative for distinguishing different image classes. On the other hand, the image patches are highly overlapped in the dense sampling space, which introduces significant redundancy among these features. Therefore, it is challenging to explore this high-dimensional, noisy, and redundant feature space. In this work, we address this issue using randomization.

4. Random Forest with Discriminative Decision Trees

In order to explore the dense sampling feature space for fine-grained visual categorization, we combine two concepts: (1) *Discriminative training* to extract the information in the image patches *effectively*; (2) *Randomization* to explore the dense feature space *efficiently*. Specifically, we adopt a random forest [3] framework where each tree node is a discriminative classifier that is trained on one or a pair of image patches. In our setting, the discriminative training and randomization can benefit from each other. We summarize the advantages of our method below:

- The random forest framework allows us to consider a subset of the image regions at a time, which allows us to explore the dense sampling space efficiently in a principled way.
- Random forest selects a best image patch in each node, and therefore it can remove the noise-prone image patches and reduce the redundancy in the feature set.
- By using discriminative classifiers to train the tree nodes, our random forest has much stronger decision trees with small correlation. This allows our method to have low generalization error (Sec.4.4) compared with the traditional random forest [3] which uses weak classifiers in the tree nodes.

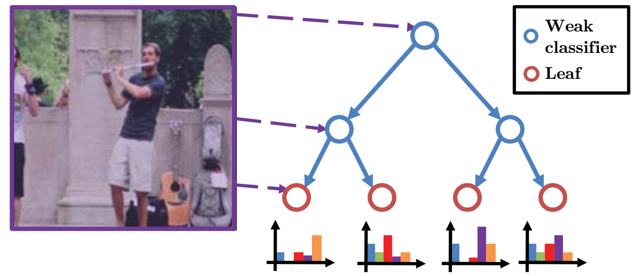
An overview of the random forest framework we use is shown in Algorithm 1. In the following sections, we first describe this framework (Sec.4.1). Then we elaborate on our feature sampling (Sec.4.2) and split learning (Sec.4.3) strategies in detail, and describe the generalization theory [3] of random forest which guarantees the effectiveness of our algorithm (Sec.4.4).

4.1. The Random Forest Framework

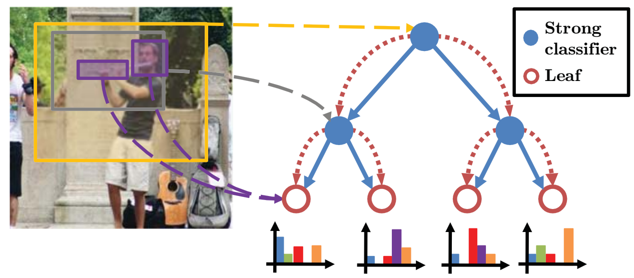
Random forest is a multi-class classifier consisting of an ensemble of decision trees where each tree is constructed via some randomization. As illustrated in Fig.3(a), the leaf nodes of each tree encode a distribution over the image classes. All internal nodes contain a binary test that splits the data and sends the splits to its children nodes. The splitting is stopped when a leaf node is encountered. An image is classified by descending each tree and combining the leaf distributions from all the trees. This method allows the flexibility to explore a large feature space effectively because it only considers a subset of features in every tree node.

Each tree returns the posterior probability of an example belonging to the given classes. The posterior probability of a particular class at each leaf node is learned as the proportion of the training images belonging to that class at the given leaf node. The posterior probability of class c at leaf l of tree t is denoted as $P_{t,l}(c)$. Thus, a test image can be classified by averaging the posterior probability from the leaf node of each tree: $c^* = \arg\max_c \frac{1}{T}\sum_{t=1}^{T} P_{t,l_t}(c)$, where c^* is the predicted class label, T is the total number of trees, and l_t is the leaf node that the image falls into.

In the following sections, we describe the process of obtaining $P_{t,l}(c)$ using our algorithm. Readers can refer to previous works [3, 1, 17] for more details of the conventional decision tree learning procedure.



(a) Conventional random decision tree.



(b) Discriminative decision tree.

Figure 3. Comparison of conventional random decision trees with our discriminative decision trees. Solid blue arrows show binary splits of the data. Dotted lines from the shaded image regions indicate the region used at each node. Conventional decision trees use information from the entire image at each node, which encodes no spatial or structural information, while our decision trees sample single or multiple image regions from the dense sampling space (Fig.2(a)). The histograms below the leaf nodes illustrate the posterior probability distribution $P_{t,l}(c)$ (Sec.4.1). In (b), dotted red arrows between nodes show our nested tree structure that allows information to flow in a top-down manner. Our approach uses strong classifiers in each node (Sec.4.3), while the conventional method uses weak classifiers.

4.2. Sampling the Dense Feature Space

As shown in Fig.3(b), each internal node in our decision tree corresponds to a single or a pair of rectangular image regions that are sampled from the dense sampling space (Sec.3), where the regions can have many possible widths, heights, and image locations. In order to sample a candidate image region, we first normalize all images to unit width

foreach *tree* t **do**
- Obtain a random set of training examples $\mathcal{D}$;
- `SplitNode(`$\mathcal{D}$`);`
if *needs to split* **then**
i. Randomly sample the candidate (pairs of) image regions (Sec.4.2);
ii. Select the best region to split $\mathcal{D}$ into two sets $\mathcal{D}_1$ and $\mathcal{D}_2$ (Sec.4.3);
iii. `SplitNode(`$\mathcal{D}_1$`)` and `SplitNode(`$\mathcal{D}_2$`)`.
else
Return $P_t(c)$ for the current leaf node.
end
end

Algorithm 1: Overview of the process of growing decision trees in the random forest framework.

and height, and then randomly sample (x_1, y_1) and (x_2, y_2) from a uniform distribution $U([0, 1])$. These coordinates specify two diagonally opposite vertices of a rectangular region. Such regions could correspond to small areas of the image (e.g. the purple bounding boxes in Fig.3(b)) or even the complete image. This allows our method to capture both global and local information in the image.

In our approach, each sampled image region is represented by a histogram of visual descriptors. For a pair of regions, the feature representation is formed by applying histogram operations (e.g. concatenation, intersection, etc.) to the histograms obtained from both regions. Furthermore, the features are augmented with the decision value $\mathbf{w}^T\mathbf{f}$ (described in Sec.4.3) of this image from its parent node (indicated by the dashed red lines in Fig.3(b)). Therefore, our feature representation combines the information of all upstream tree nodes that the corresponding image has descended from. We refer to this idea as "nesting". Using feature sampling and nesting, we obtain a candidate set of features, $\mathbf{f} \in \mathbb{R}^n$, corresponding to a candidate image region of the current node.

Implementation details. Our method is flexible to use many different visual descriptors. In this work, we densely extract SIFT [14] descriptors on each image with a spacing of four pixels. The scales of the grids to extract descriptors are 8, 12, 16, 24, and 30. Using k-means clustering, we construct a vocabulary of codewords[1]. Then, we use Locality-constrained Linear Coding [20] to assign the descriptors to codewords. A bag-of-words histogram representation is used if the area of the patch is smaller than 0.2, while a 2-level or 3-level spatial pyramid is used if the area is between 0.2 and 0.8 or larger than 0.8 respectively.

During sampling (step i of Algorithm 1), we consider four settings of image patches: a single image patch and three types of pairwise interactions (concatenation, intersection, and absolute of difference of the two histograms). We sample 25 and 50 image regions (or pairs of regions) in the root node and the first level nodes respectively, and sample 100 regions (or pairs of regions) in all other nodes. Sampling a smaller number of image patches in the root can reduce the correlation between the resulting trees.

[1]A dictionary size of 1024, 256, 256 is used for PASCAL action [7], PPMI [22], and Caltech-UCSD Birds [21] datasets respectively.

4.3. Learning the Splits

In this section, we describe the process of learning the binary splits of the data using SVM (step ii in Algorithm 1). This is achieved in two steps: (1) Randomly assigning all examples from each class to a binary label; (2) Using SVM to learn a binary split of the data.

Assume that we have C classes of images at a given node. We uniformly sample C binary variables, $\mathbf{b}$, and assign all examples of a particular class c_i a class label of b_i. As each node performs a binary split of the data, this allows us to learn a simple binary SVM at each node. This improves the scalability of our method to a large number of classes and results in well-balanced trees. Using the feature representation $\mathbf{f}$ of an image region (or pairs of regions) as described in Sec.4.2, we find a binary split of the data:

$$\begin{cases} \mathbf{w}^T\mathbf{f} \leq 0, \text{go to left child} \\ \text{otherwise, go to right child} \end{cases}$$

where $\mathbf{w}$ is the set of weights learned from a linear SVM.

We evaluate each binary split that corresponds to an image region or pairs of regions with the information gain criteria [1], which is computed from the complete training images that fall at the current tree node. The splits that maximize the information gain are selected and the splitting process (step iii in Algorithm 1) is repeated with the new splits of the data. The tree splitting stops if a pre-specified maximum tree depth has been reached, or the information gain of the current node is larger than a threshold, or the number of samples in the current node is small.

4.4. Generalization Error of Random Forests

In [3], it has been shown that an upper bound for the generalization error of a random forest is given by $\rho(1-s^2)/s^2$, where s is the strength of the decision trees in the forest, and ρ is the correlation between the trees. Therefore, the generalization error of a random forest can be reduced by making the decision trees stronger or reducing the correlation between the trees.

In our approach, we learn discriminative SVM classifiers for the tree nodes. Therefore, compared to the traditional random forests where the tree nodes are weak classifiers of randomly generated feature weights [1], our decision trees are much stronger. Furthermore, since we are considering an extremely dense feature space, each decision tree only considers a relatively small subset of image

Method	Phoning	Playing instrument	Reading	Riding bike	Riding horse	Running	Taking photo	Using computer	Walking	Overall
CVC-BASE	**56.2**	56.5	34.7	75.1	83.6	86.5	25.4	60.0	69.2	60.8
CVC-SEL	49.8	52.8	34.3	74.2	85.5	85.1	24.9	64.1	72.5	60.4
SURREY-KDA	52.6	53.5	35.9	81.0	89.3	86.5	32.8	59.2	68.6	62.2
UCLEAR-DOSP	47.0	**57.8**	26.9	78.8	89.7	87.3	32.5	60.0	70.1	61.1
UMCO-KSVM	53.5	43.0	32.0	67.9	68.8	83.0	34.1	45.9	60.4	54.3
Our Method	45.0	57.4	**41.5**	**81.8**	**90.5**	**89.5**	**37.9**	**65.0**	**72.7**	**64.6**

Table 1. Comparison of the average precision (%) of our method with the winners of PASCAL VOC2010 action classification challenge [7]. Each row shows the results obtained from one method. The best results are highlighted with bold fonts.

patches. This means there is little correlation between the trees. Therefore, our random forest with discriminative decision trees algorithm can achieve very good performance on fine-grained image classification, where exploring fine image statistics discriminatively is important. In Sec.5.4, we show the strength and correlation of different settings of random forests with respect to the number of decision trees, which justifies the above arguments. Please refer to [3] for details about how to compute the strength and correlation values for a random forest.

5. Experiments

In this section, we evaluate our algorithm on three fine-grained image datasets: the action classification dataset in PASCAL VOC2010 [7] (Sec.5.1), actions of people-playing-musical-instrument (PPMI) [22] (Sec.5.2), and a subordinate object categorization dataset of 200 bird species [21] (Sec.5.3). Experimental results show that our algorithm outperforms state-of-the-art methods on these datasets. We also evaluate the strength and correlation of the decision trees in our method, and compare the result with the other settings of random forests to show why our method can lead to better classification performance (Sec.5.4).

5.1. PASCAL Action Classification

The most recent PASCAL VOC challenge [7] incorporated the task of recognizing actions in still images. The images describe nine common human activities: “Phoning”, “Playing a musical instrument”, “Reading”, “Riding a bicycle or motorcycle”, “Riding a horse”, “Running”, “Taking a photograph”, “Using a computer”, and “Walking”. Each person that we need to classify is indicated by a bounding box and is annotated with one of the nine actions they are performing. There are 40$\sim$90 training/validation images and a similar number of testing images for each class.

As in [5], we obtain a foreground image for each person by extending the bounding box of the person to contain $1.5\times$ the original size of the bounding box, and resizing it such that the larger dimension is 300 pixels. We also resize the original image accordingly. Therefore for each person, we have a “person image” as well as a “background image”.

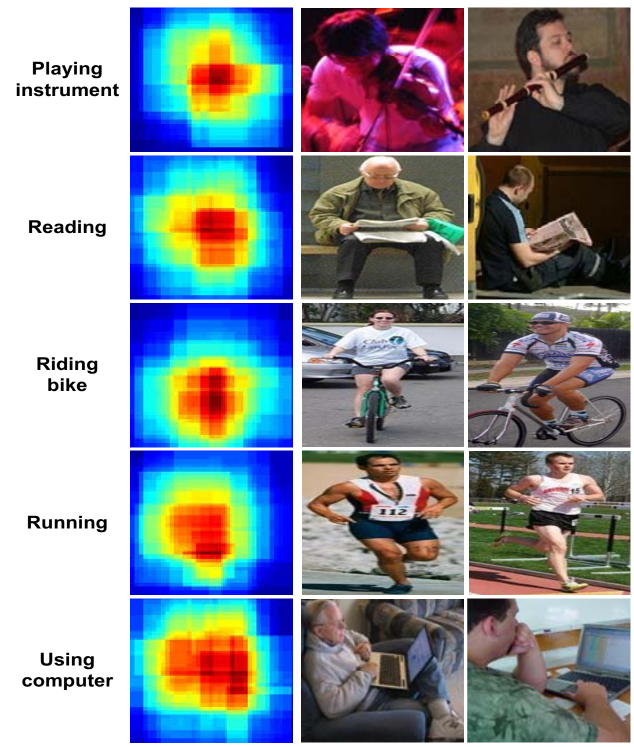


Figure 4. Heat maps that show distributions of frequency that an image patch is selected in our method. The heat maps are obtained by aggregating image regions of all the tree nodes in the random forest weighted by the probability of the corresponding class. Red indicates high frequency and blue indicates low frequency.

We only sample regions from the foreground and concatenate the features with a 2-level spatial pyramid of the background. We use 100 decision trees in our random forest.

We compare our algorithm with the methods on PASCAL challenge [7] that achieve the best average precision. The results are shown in Tbl.1. Our method outperforms the others in terms of mean average precision, and achieves the best result on seven of the nine actions. Note that we achieved this accuracy based on only grayscale SIFT descriptors, without using any other features or contextual information like object detectors.

Fig.4 shows the frequency of an image patch being se-

Method	BoW	Grouplet [22]	SPM [13]	LLC [20]	Ours
mAP (%)	22.7	36.7	39.1	41.8	**47.0**

Table 2. Mean Average Precision (% mAP) on the 24-class classification problem of the PPMI dataset. The best result is highlighted with bold fonts. The grouplet uses one SIFT scale, while all the other methods use multiple SIFT scales described in Sec.4.2.

Instrument	BoW	Grouplet [22]	SPM [13]	LLC [20]	Ours
Bassoon	73.6	78.5	84.6	85.0	**86.2**
Erhu	82.2	87.6	88.0	89.5	**89.8**
Flute	86.3	95.7	95.3	97.3	**98.6**
FrenchHorn	79.0	84.0	93.2	93.6	**97.3**
Guitar	85.1	87.7	**93.7**	92.4	93.0
Saxophone	84.4	87.7	89.5	88.2	**92.4**
Violin	80.6	93.0	93.4	**96.3**	95.7
Trumpet	69.3	76.3	82.5	86.7	**90.0**
Cello	77.3	84.6	85.7	82.3	**86.7**
Clarinet	70.5	82.3	82.7	84.8	**90.4**
Harp	75.0	87.1	92.1	**93.9**	92.8
Recorder	73.0	76.5	78.0	79.1	**92.8**
Average	78.0	85.1	88.2	89.2	**92.1**

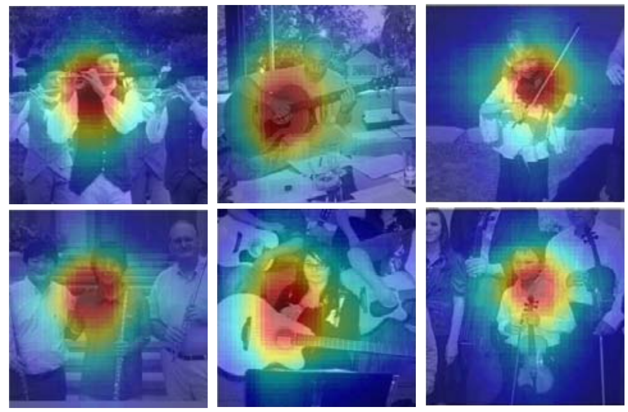
Table 3. Comparison of mean Average Precision (% mAP) of our method and the baseline results on the PPMI binary classification tasks of people playing and holding different musical instruments. Each column shows the results obtained from one method. The best results are highlighted with bold fonts.

lected by our method. For each activity, the figure is obtained by considering the features selected in the tree nodes weighted by the proportion of samples of this activity in this node. From the results, we can clearly see the difference of distributions for different activities. For example, the image patches corresponding to human-object interactions are usually highlighted, such as the patches of bikes and books. We can also see that the image patches corresponding to background are not frequently selected. This demonstrates our algorithm's ability to deal with background clutter.

5.2. People-Playing-Musical-Instrument (PPMI)

The people-playing-musical-instrument (PPMI) data set is introduced in [22]. This data set puts emphasis on understanding subtle interactions between humans and objects. There are twelve musical instruments; for each instrument there are images of people playing the instrument and holding the instrument but not playing it. We evaluate the performance of our method with 100 decision trees on the 24-class classification problem. We compare our method with many baseline results[2]. Tbl.2 shows that we significantly outperform the baseline results.

[2] The baseline results are available from the dataset website http://ai.stanford.edu/∼bangpeng/ppmi.



(a) flute (b) guitar (c) violin

Figure 5. (a) Heat map of the dominant regions of interest selected by our method for playing flute on images of playing flute (top row) and holding a flute without playing it (bottom row). (b,c) shows similar images for guitar and violin, respectively. Refer to Fig.4 for how the heat maps are obtained.

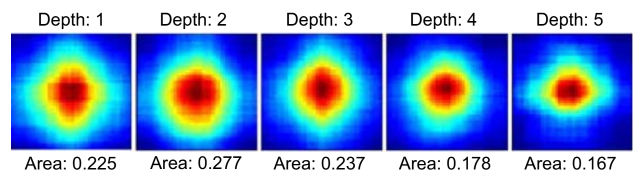


Figure 6. Heat map for "playing trumpet" class with the weighted average area of selected image regions for each tree depth.

Tbl.3 shows the result of our method on the 12 binary classification tasks where each task involves distinguishing the activities of playing and not playing for the same instrument. Despite a high baseline of 89.2% mAP, our method outperforms by 2.9% to achieve a result of 92.1% overall. Furthermore, we outperform the baseline methods on nine of the twelve binary classification tasks. In Fig.5, we visualize the heat map of the features learned for this task. We observe that they show semantically meaningful locations of where we would expect the discriminative regions of people playing different instruments to occur. For example, for flute, the region around the face provides important information while for guitar, the region to the left of the torso provides more discriminative information. It is interesting to note that despite the randomization and the algorithm having no prior information, it is able to locate the region of interest reliably.

Furthermore, we also demonstrate that the method learns a coarse-to-fine region of interest for identification. This is similar to the human visual system which is believed to analyze raw input in order from low to high spatial frequencies or from large global shapes to smaller local ones [4]. Fig.6 shows the heat map of the area selected by our classifier as we consider different depths of the decision tree. We ob-

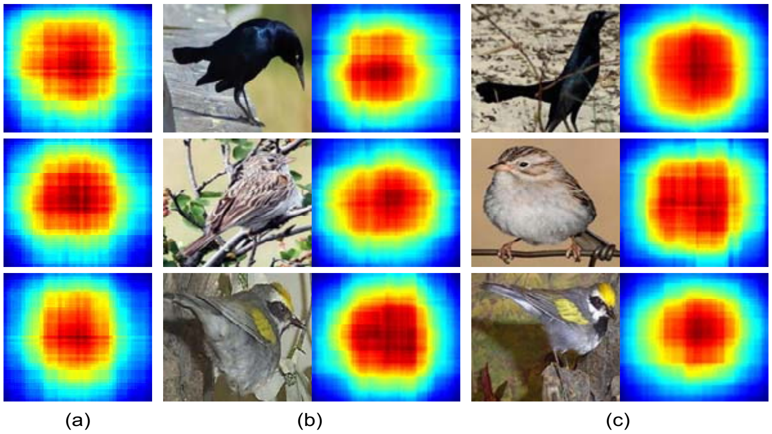


Figure 7. Each row represents visualizations for a single class of birds (from top to bottom): boat tailed grackle, brewer sparrow, and golden winged warbler. For each class, we visualize: (a) Heat map for the given class as described in Fig.4; (b,c) Two example images of the corresponding class and the distribution of image patches selected for the specific image. The heat maps are obtained by descending each tree for the corresponding image and only considering the image regions of the nodes that this image falls in.

Method	MKL [2]	LLC [20]	Ours
Accuracy	19.0%	18.0%	**19.2%**

Table 4. Comparison of the mean classification accuracy of our method and the baseline results on the Caltech-UCSD Birds 200 dataset. The best performance is indicated with bold fonts.

serve that our random forest follows a similar coarse-to-fine structure. The average area of the patches selected reduces as the tree depth increases. This shows that the classifier first starts with more global features or high frequency features to discriminate between multiple classes, and finally zeros in on the specific discriminative regions for some particular classes.

5.3. Caltech-UCSD Birds 200 (CUB-200)

The Caltech-UCSD Birds (CUB-200) dataset contains 6,033 annotated images of 200 different bird species [21]. This dataset has been designed for subordinate image categorization. It is a very challenging dataset as the different species are very closely related and have similar shape/color. There are around 30 images per class with 15 for training and the remaining for testing. The test-train splits are fixed (provided on the website).

The images are cropped to the provided bounding box annotations. These regions are resized such that the smaller image dimension is 150 pixels. As color provides important discriminative information, we extract C-SIFT descriptors [19] in the same way described in Sec.4.2. We use 300 decision trees in our random forest. Tbl.4 compares the performance of our algorithm against the LLC baseline and the state-of-the-art result (multiple kernel learning (MKL) [2]) on this dataset. Our method outperforms LLC and achieves comparable performance with the MKL approach. We note that [2] uses multiple features e.g. geometric blur, gray/color SIFT, full image color histograms etc. It is expected that including these features can further improve the performance of our method. Furthermore, we show in Fig.7 that our method is able to capture the intra-class pose variations by focusing on different image regions for different images.

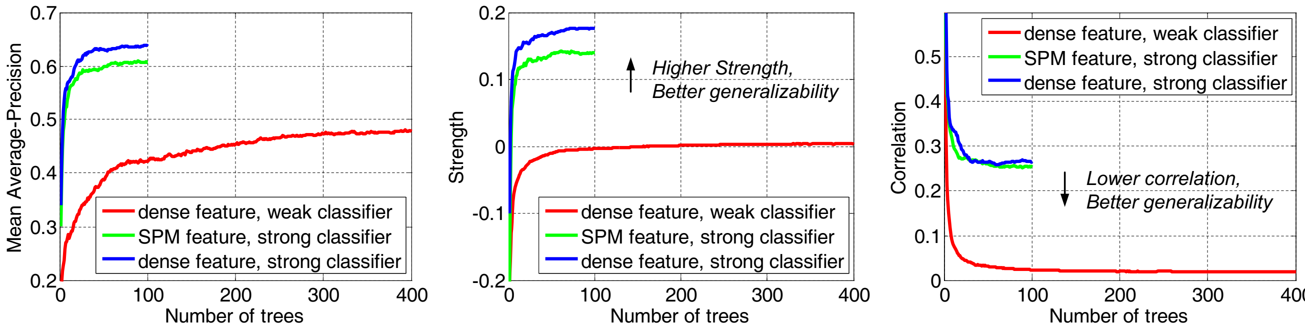
5.4. Strength and Correlation of Decision Trees

We compare our method against two control settings of random forests on the PASCAL action dataset [7].

- *Dense feature, weak classifier*: For each image region or pairs of regions sampled from our dense sampling space, replace the SVM classifier in our method with a weak classifier as in the conventional decision tree learning approach [1], i.e. randomly generating 100 sets of feature weights and select the best one.
- *SPM feature, strong classifier*: Use SVM classifiers to split the tree nodes as in our method, but the image regions are limited to that from a 4-level spatial pyramid.

Note that all other settings of the above two approaches remain unchanged as compared to our method (as described in Sec.4). Fig.8 shows that on this dataset, a set of strong classifiers with relatively high correlation can lead to better performance than a set of weak classifiers with low correlation. We can see that the performance of random forests can be significantly improved by using strong classifiers in the nodes of decision trees. Compared to the random forests that only sample spatial pyramid regions, using the dense sampling space obtains stronger trees without significantly increasing the correlation between different trees, thereby improving the classification performance. Furthermore, the performance of the random forests using discriminative node classifiers converges with a small number of decision trees, indicating that our method is more efficient than the conventional random forest approach. In our experiment, the two settings and our method need a similar amount of time to train a single decision tree.

Additionally, we show the effectiveness of random binary assignment of class labels (Sec.4.3) when we train classifiers for each tree node. Here we ignore this step and



(a) Mean average precision (mAP). (b) Strength of the decision trees. (c) Correlation between the decision trees.

Figure 8. (a) We compare the classification performance (mAP) obtained by our method "dense feature, strong classifier" with two control settings. Please refer to Sec.5.4 for details of these settings. (b,c) We also compare the strength of the decision trees learned by these approaches and correlation between these trees (Sec. 4.4), which are highly related to the generalization error of random forests.

train a one-vs-all multi-class SVM for each sampled image region or pairs of regions. In this case C sets of weights are obtained when there are C classes of images at the current node. The best set of weights is selected using information gain as before. This setting leads to deeper and significantly unbalanced trees, and the performance decreases to 58.1% with 100 trees. Furthermore, it is highly inefficient as it does not scale well with increasing number of classes.

6. Conclusion

In this work, we propose a random forest with discriminative decision trees algorithm to explore a dense sampling space for fine-grained image categorization. Experimental results on subordinate classification and activity classification show that our method achieves state-of-the-art performance and discovers much semantically meaningful information. The future work is to jointly train all the information that is obtained from the decision trees.

Acknowledgement. L.F-F. is partially supported by an NSF CAREER grant (IIS-0845230), an ONR MURI grant, the DARPA VIRAT program and the DARPA Mind's Eye program. B.Y. is partially supported by the SAP Stanford Graduate Fellowship. We would like to thank Carsten Rother and his colleagues in Microsoft Research Cambridge for helpful discussions about random forest. We also would like to thank Hao Su and Olga Russakovsky for their comments on the paper.

References

[1] A. Bosch, A. Zisserman, and X. Muñoz. Image classification using random forests and ferns. In *ICCV*, 2007.

[2] 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.

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

[4] C. A. Collin and P. A. McMullen. Subordinate-level categorization relies on high spatial frequencies to a greater degree than basic-level categorization. *Percept. Psychophys.*, 67(2):354–364, 2005.

[5] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: A study of bag-of-features and part-based representations. In *BMVC*, 2010.

[6] G. Duan, C. Huang, H. Ai, and S. Lao. Boosting associated pairing comparison features for pedestrian detection. In *ICCV Workshop on Visual Surveillance*, 2009.

[7] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results.

[8] L. Fei-Fei, R. Fergus, and A. Torralba. Recognizing and learning object categories. In *ICCV Tutorial*, 2005.

[9] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE T. Pattern Anal.*, 32(9):1627–1645, 2010.

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

[12] K. E. Johnson and A. T. Eilers. Effects of knowledge and development on subordinate level categorization. *Cognitive Dev.*, 13(4):515–545, 1998.

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

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[15] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, 2007.

[16] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001.

[17] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[18] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *ICCV*, 2005.

[19] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE T. Pattern Anal.*, 32(9):1582–1596, 2010.

[20] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[21] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD birds 200. Technical Report CNS-TR-201, Caltech, 2010.

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