# Fused One-vs-All Mid-Level Features for Fine-Grained Visual Categorization

Xiaopeng Zhang[1], Hongkai Xiong[1], Wengang Zhou[2], Qi Tian[3]
[1] Dept. of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
[2] Dept. of EEIS, University of Science and Technology of China, Hefei, 230027, China
[3] Dept. of Computer Science, University of Texas at San Antonio, Texas, TX 78249
{zxphistory, xionghongkai}@sjtu.edu.cn[1], zhwg@ustc.edu.cn[2], qitian@cs.utsa.edu[3]

## ABSTRACT

As an emerging research topic, fine-grained visual categorization has been attracting growing attentions in recent years. Due to the large inter-class similarity and intra-class variance, recognizing objects in fine-grained domains is extremely challenging, and sometimes even humans can not recognize them accurately. Traditional bag-of-words model could obtain desirable results for basic-level category classification by weak alignment using spatial pyramid matching model, but may easily fail in fine-grained domains since the discriminative features are not only subtle but also extremely localized. The fine differences often get swamped by those irrelevant features, and it is virtually impossible to distinguish them.

To address the problems above, we propose a new framework for fine-grained visual categorization. We strengthen the spatial correspondence among parts by including foreground segmentation and part localization. Based on the part representations of the images, we learn a large set of mid-level features which are more suitable for fine-grained tasks. Comparing with the low level features directly extracted from the images, the learned one-vs-all mid-level features enjoy the following advantages. First, the dimension of the mid-level features is relatively small. In order to obtain high classification accuracy, the dimension of the low level features usually reaches several thousand to tens of thousand, and becomes even larger when introducing spatial pyramid model. However, the dimension of our mid-level features is related to the number of classes, which is far less. Second, each entry of the proposed mid-level features is meaningful, which forms a more compact representation of the image. Third, the mid-level features are more robust than the low level ones, which is helpful for classification. Fourth, the learning process of the mid-level features is independent and can be easily combined with other techniques to boost the performance. We evaluate the proposed approach on the extensive fine-grained dataset CUB 200-2011 and Stanford Dogs, by learning the mid-level features based on the popular Fisher vectors and convolutional neural network, we boost the classification accuracy by a considerable margin and advance the state-of-the-art performance in fine-grained visual categorization.


*MM'14,* November 3–7, 2014, Orlando, Florida, USA.

http://dx.doi.org/10.1145/2647868.2654937.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

## General Terms

Algorithms, Experimentation

## Keywords

Mid-level Features; Fine-Grained Visual Categorization; Convolutional Neural Networks; Image Similarity

## 1. INTRODUCTION

As an emerging research topic, fine-grained visual categorization discriminates typically hundreds of sub-categories belonging to the same basic-level category. It lies between the basic-level category classification (*e.g.* The PASCAL VOC dataset including bikes, boats, cars and so on) and the identification of individual instances (*e.g.* face recognition). The major challenge lies in great intra class variation and sometimes small inter-class variation. Taking the widely used fine-grained dataset CUB 200-2011 as an example, as shown in Fig. 1, it is even difficult for the humans to recognize them accurately. The basic-level categorization such as distinguishing a car from a dog is easy because there are plenty of helpful visual cues to tell them apart, while for fine-grained visual categorization, there are much fewer discriminative features compared with that at the basic-level. The differences between fine-grained classes are very subtle and extremely localized, *e.g.* only the shape of the beak or the color of the crown matter when recognizing similar birds species.

Traditional Bag-of-Features [7] framework has been widely used in various image classification applications due to its simpleness and effectiveness. However, with the ignorance of spatial layout information of the features, it suffers severely limited descriptive capability. A standard way to introduce weak geometry in Bag-of-Features representation is the use of spatial histogram [17], which defines pooling regions based on a uniform grid at predefined scales (typically the whole image, then quadrants, sixteenths, *etc.*). The spatial pyramid matching method is effective for basic-level category classification, but suffers in fine-grained domains

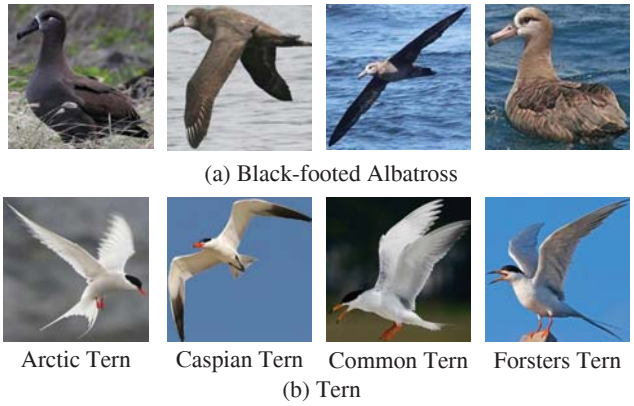


**Figure 1: Sample images from the CUB-200-2011 dataset, which shows (a) great intra class variance and (b) small inter-class variance. It's hard even for humans to recognize them accurately.**

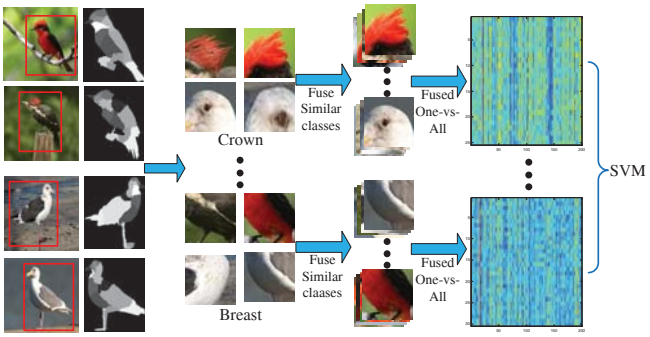


**Figure 2: An overview framework of our proposed algorithm. Foreground segmentation and part localization are performed before feature extraction. The fused one-vs-all mid-level features are defined by specifying any part, fusing classes with similar parts into one super-class, and supervised learning one-vs-all mid-level features with SVM, then another SVM classifier is learned based on the mid-level features to get the final classification results.**

due to the highly localized nature of distinguished features. The fact implies that a stronger correspondence should be considered to achieve higher classification accuracy. Hence, all of current works on fine-grained visual categorization use part-based features in one way or another. As a matter of fact, fine-grained visual categorization conveniently enables part-based models, as all the sub-categories share the same type of parts and attributes. For example, all birds should have beaks, wings and legs. Once the discriminative parts are determined, they are encoded into different parts of visual words, enabling the classifier to pick up the differences based on parts. On the other hand, it is well known that the background information often offers useful clues in basic-level categorization. For example, the background road is helpful for classifying cars from other categories. However, backgrounds are seldom discriminative for fine-grained visual categorization because all of them are in the similar scene. For example, all birds are usually either on trees or flying in the sky. Hence, ignoring the background information is a reasonable operation. In the light of these observations, it is necessary to incorporate foreground segmentation and part localization into fine-grained categorization to improve the classification accuracy.

Most of current classification tasks follow the pipeline of extracting basic descriptors like SIFT and HOG, quantizing the descriptors into compact visual words and pooling the visual word histograms for image representation. However, these low level features may not be optimal for specific classification tasks due to the well-known semantic gap [18] between low level features and high-level image semantics. Furthermore, extremely large dimension of the low level features, especially for Fish vectors, leads to overfitting in a discriminative hyperplane spanned by the linear classifiers, which degrades the classification accuracy and increases the computational complexity. Various methods are proposed to address these problems. Since single descriptor might fail to capture the rich information within local pathes, it is reasonable to extract multiple descriptors for compensation. They are more discriminative for classification as they describe the image from multiple aspects. By simply concatenating different descriptors together, [3] has outperformed the state-of-the-art result using single descriptor. The others [25], [4], [29] learn semantic representations of the images by aggregating neighboring descriptors to form micro-features or visual phrases. In comparison to the low level features, they are more meaningful. However, they are de facto low level features suffering from the dimension dilemma.

This paper proposes a new framework for fine-grained visual categorization based on the above observations. As shown in Fig. 2, first, we infer the foreground objects and localize the parts to enable part-based feature extractions. Coarse-to-fine part localization is obtained according to the accessibility of the ground truth part annotations. Second, we learn one-vs-all mid-level features part by part based on the extracted low level features. Compared with the high-dimensional low level features directly extracted from the images, the proposed mid-level features are dimension friendly, entry meaningful and robust for classification. The dimension of the mid-level features is related to the number of classes, which is far less than that of the low level ones and each entry of the mid-level features has specific meaning. Furthermore, our proposed learning algorithm can be easily combined with other techniques to boost the performance. Integrating all these techniques produces a more powerful framework for fine-grained visual categorization, and outperforms the state-of-the-art classification accuracy by a noticeable margin.

The rest of this paper is organized as follows. In Section 2, we review related works on fine-grained visual categorization. In Section 3, foreground segmentation and part localization are performed before feature extraction. Depending on the accessibility of the ground truth part annotations, alternative methods are selected to obtain fine or coarse segmentation and localization results. Section 4 presents the detailed algorithm of learning one-vs-all mid-level features with elaborated analysis of their principles. Furthermore, the fused one-vs-all mid-level features are proposed to enhance the performance. Experimental results are shown in Section 5. Finally, we draw our conclusion in Section 6.

## 2. RELATED WORK

Fine-grained categorization has been studied only recently and quickly becomes a popular topic in recognition. Previous works have aimed at various aspects of fine-grained categorization, such as the localization of parts, the description of the part objects and the human in the loop techniques to boost recognition accuracy.

### 2.1 Fine-Grained Part Localization

Part-based methods have recently experienced renewed interest and success [12]. Fine-grained visual categorization also conveniently enables the part-based approaches because objects belong to the same basic level often share the same parts. The detection of fine-grained categories includes segmenting the objects of interest from the background and localizing the individual parts of the objects. In [6] Chai *et al.* introduce techniques that can improve both segmentation and part localization accuracy by co-segmentation. Considering the varied poses and appearances of bird species, Zhang *et al.* [27] propose pose pooling kernel to define semantic pooling regions and aggregate features from each part regions. In [19] Parkhi *et al.* propose to use deformable part models to detect the head of the cats and the dogs. Berg *et al.* [2] propose to automatically detect part locations using the face localization methods. The work in [26] accomplishes unsupervised learning of a deformable part model to find discriminative parts for fine-grained categorization. These methods attempt to find parts of the images that are discriminative in different ways without the explicit part labels, but can't achieve the accuracy performance of a supervised part-based approach. In [24], Xie *et al.* use the Ultrametric Contour Map as an unsupervised algorithms to calculate the closed boundaries with the guidance of the part annotations. Although its accuracy performance is promising, it suffers the debate of using too many human annotations, which is unrealistic in real-word applications.

### 2.2 Fine-Grained Descriptors

For the description of fine-grained objects, different proposals have been made in the literature. The most widely used descriptors are color SIFT, gray SIFT descriptors plus color histogram [6], [13]. Zhang *et al.* [28] use gradient, local binary pattern, RGB color and normalized RGB color based kernel descriptors. A common characteristic of these descriptors is that they are largely handcrafted. They are all comprising dense sampling of local image patches, describing them by means of low level visual descriptors, encoding them into a high-dimensional representation, and pooling over the images. There are also works trying to learn features from the dataset. Berg *et al.* [2] build part-based one-vs-one features library which are discriminative mid-level features, and use these features for classification.

Recently, these handcrafted descriptors have been substantially outperformed by the introduction of convolutional neural networks [15], which have a more complicated structure than traditional representations. They contain several layers of non-linear feature extractors, and are said to be deep representation of images (in contrast, traditional descriptors such as SIFT would be referred as shallow representation). These networks have achieved competition-winning numbers on large benchmark dataset. It has been demonstrated that training a convolutional network to simultaneously classify, locate and detect objects in images can boost both the classification accuracy and localization accuracy [20]. The advantage of convolutional neural networks is that the system is end-to-end, and alleviates the requirement to manually design a suitable feature extractor. Though not specifically designed to model subcategory level differences, it has been demonstrated that [10] the convolutional neural network features capture such information well and obtain the state-of-the-art results for fine-grained visual categorization so far.

### 2.3 Human Interaction

Since fine-grained visual categorization is difficult for both humans and computers, an interactive method that assists a human in discovering the true class is useful and preferable. Human interaction methods have recently experienced a strong resurgence in popularity. Deng *et al.* [9] propose a "human-in-the-loop" approach to find accurate distinctive regions for recognition. It resorts to an online game Bubble to help find which part/parts are more helpful for categorization, then SIFT descriptors are extracted from these labeled regions for training classifiers. Duan *et al.* [11] propose to use a latent conditional random field to generate localized attributes that are both machine and human friendly, and employ a recommender system that selects attributes likely to be semantically meaningful, then human interaction is used to provide semantic names for the discovered attributes.

## 3. SEGMENTATION AND PART LOCALIZATION

For the fine-grained visual categorization, almost all the categories share similar background clutters. Ignoring the background information is a reasonable preprocessing step since background seldom offers useful clues for recognition of sub-categories. On the other hand, the regular spatial pyramid model often fails to align the corresponding parts in the fine-grained tasks, we need a more accurate part alignment method to capture the semantic parts of the objects. We chose two different kinds of foreground segmentation and part localization methods, namely fine part localization and coarse part localization, according to the accessibility of the ground truth part annotations.

### 3.1 Segmentation and Part Localization with Ground Truth Part Annotations

The ground truth part annotations provided in the CUB 200-2011 [22] dataset include the locations of fifteen parts, *e.g* back, beak, belly and so on. With the help of the provided bounding boxes and part annotations, we could obtain accurate foreground masks and fine part localization results. We use the Grab-Cut algorithms [5] for foreground segmentation as in [24]. The initial mask is constructed using ground truth part annotations, the pixels outside the bounding boxes are regarded as definite background, the annotations are regarded as definite foreground and the others are regarded as possible foreground. In the vast majority of cases the algorithms are able to return a rather precise contour of the object.

With the guidance of part annotations, we calculate the Ultrametric Contour Map [1] of the foreground object, which generates closed contours with decreasing boundary intensities to cut the image into smaller and smaller regions. De-

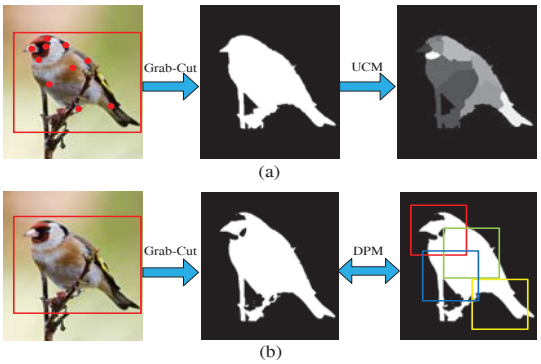


**Figure 3: The foreground segmentation and part localization in different situations. (a) With ground truth part annotations, fine part localization can be obtained via Grab-Cut and Ultrametric Contour Map segmentation. (b) Using the Grab-Cut and Deformable Part Model interactively to obtain coarse part localization with only provided bounding boxes, the color boxes frame the detected part regions.**

note the image as $U = (u_{i,j})_{W \times H}$, where $u_{i,j}$ is the pixel values at position $(i, j)$, we construct a directed graph $G = (V, \varepsilon)$, where $V = \{v_{ij}\}$ consists of all pixels, and $\varepsilon$ represents edges connecting adjacent pixels:

$$\varepsilon = \{(v_{ij} \rightarrow v_{i'j'}) | \mid i - i' \mid \ + \mid j - j' \mid = 1\} \quad (1)$$

The weight of an edge is related to the boundary intensity at the tail node, which is denoted as:

$$\omega = (v_{ij} \rightarrow v_{i'j'}) = u_{i'j'} + \lambda \quad (2)$$

here, $\lambda$ is called step penalty, which takes the geometric distance into consideration. After the graph is complete, we take the part annotation points as seed nodes, and calculate their shortest paths to the other nodes. The segmentation process is to assign each node to its nearest part annotation nodes, taking both the intensity values and coordinate distance into consideration. The segmentation and part localization process is illustrated in Fig. 3(a), the foreground in different colors represents different part regions.

## 3.2 Segmentation and Part Localization without Ground Truth Part Annotations

The ground truth part annotations need trivial manual calibration and do not give automatically classification performance. To accomplish automatical classification, without loss of generality, we also include segmentation and part localization results without the ground truth part annotations. The deformable part model [12] is the most widely used part-based object detection model for automatic part localization. Chai *et al.* [6] have demonstrated that the foreground segmentation and part localization can boost each other if they are performed interactively, better segmentation results can be obtained by taking part localization into account when performing segmentation and likewise, more accurate parts can be detected if the foreground masks are considered when performing part localization. Based on the deformable part model $W$ and saliency model $S$ trained using a set of training images, the model is fitted to an image $I$ via jointly optimizing the Grab-Cut and deformable part model objective functions:

$$E(p, f, c|W, S, I) = \alpha E^{dpm}(p|W, I) + \beta E^{gc}(f, c|I) + E^{c}(p, f|S) \quad (3)$$

Here, $p, f, c$ represent the part localization, the foreground mask and the color distributions of the foreground and background, respectively. $\alpha$ and $\beta$ are weights controlling the balance between the energy terms. $E^{dpm}$ and $E^{gc}$ are the deformable part model and grab cut energy function, $E^{c}$ is a penalty function which penalizes the case where the foreground segmentation $f$ and part localization $p$ do not agree. By imposing the consistency between foreground segmentation and part localization, better segmentation and part localization results can be obtained comparing to simply concatenation of the two operations. Sample segmentation and part localization results are shown in Fig. 3(b).

# 4. LEARNING MID-LEVEL FEATURES

Based on the foreground segmentation and part localizations, different features can be extracted from each part. The image $I$ can be represented as a set of local features:

$$D = \{(f_1, R_1), (f_2, R_2), ..., (f_M, R_M)\} \quad (4)$$

where $f_i$, $R_i$, $i \in \{1, 2, ..., M\}$ denote the $ith$ feature vector and the occupied region respectively, and $M$ is the total number of regions. The features can be obtained via traditional methods such as Fisher vector aggregating SIFT-like low level descriptors or convolutional neural networks.

We could get the representation of the image by simply concatenating different part features into a long feature vector, then traditional SVM classifier is learned and evaluated based on the long feature vectors. It's the pipeline that most of the classification tasks follow. However, this kind of image representation is not only high dimensional (to achieve high classification accuracy, the dimension of the low level features could be as large as tens of thousand, and even larger when concatenating different parts into a long one) but also entry meaningless, from which we could not interpret what does each entry of the features mean. Furthermore, with these high-dimensional feature vectors, a discriminative hyperplane can be easily obtained even using linear classifier, but it may also introduce overfitting and perform badly on the test features. Based on these observations, we try to learn mid-level features which are dimensional friendly and semantic meaningful, and robust to some extent when performing classification.

## 4.1 One-vs-All Mid-Level Features

Our method requires as input part-based features extracted from the same part of different objects, and annotated with class labels. Attempting to learn mid-level features is intuitive since they share more semantic meanings than low level ones. Given the reference dataset, let the training set consists of images belong to $N$ classes $\{1, ..., N\}$ and includes $P$ parts $\{1, ...P\}$. The one-vs-all mid-level features are learned as follows:

1. Select any part $p \in \{1, ..., P\}$ from the objects.

2. Learning one-vs-all mid-level features based on the part $p$ features. Choosing the part features from all the classes $\{1, ..., N\}$ which describe the part $p$, except those zero vectors, we denote these features as $f^p$ . Based on the part features $f^p$, we learn a one-vs-all SVM classifier, and project

the part features $f^p$ to get one-vs-all scores based on the learned SVM weights. The dimension of the projected scores is equal to the number of classes, which is $N$ dimension in our definitions.

3. According to the projection transformation, all the low level training features are mapped into the new mid-level feature space. After an $L_2$ normalization, we simply concatenate the mid-level features part by part, by this way we can obtain our mid-level feature representation for any training image.

4. For any image in the test set, we extract low level features based on part $p$, and project corresponding part features to $N$ dimensional vector according to the learned SVM weights. Concatenating the mid-level features the same as the training ones, from which we can get our one-vs-all mid-level features for test images.

After a simple one-vs-all transformation, all the low level features are projected to the mid-level feature space. The advantages of the learned mid-level features over the low level ones are obvious. Firstly, the dimension of the mid-level features is far less than that of the low level ones, since the transformation projects the low level features to $N$ dimensional feature vectors regardless of the dimension of the low level features, where $N$ is the number of classes, so the total dimension of the mid-level features is $NP$, which are only several thousand in most situations. We would demonstrate the super-performance of the learned mid-level features in the following sections. Secondly, comparing with the low level features which are entry meaningless, every entry of the mid-level features has specific semantic meaning. For example, given a reference image $x$, denote the mid-level features for part $p$ as $X^p = \{X_1^p, X_2^p, ...X_N^p\}$, The score $X_i^p (i \in \{1, ...N\})$ represents the signed distance between current image to category $i$ for part $p$. The larger the value $X_i^p$ is, the more similar of image $x$ with category $i$ for part $p$. This kind of formulation is very helpful and understandable for classification, if $X_i^p$ is larger than any other scores in $X_p$, we can say that part $p$ of image $x$ is most similar with that of class $i$.

## 4.2 Deep Insights for One-vs-All Mid-Level Features

Now we have a deep insight into the one-vs-all mid-level features. The learned method is easy to understand, however, the intuition behind the simple algorithm is not that straightforward. Here, we clarify the advantages of our proposed one-vs-all mid-level features to give a better understanding of it. For a given part $p \in \{1, ..., P\}$, given a set of part-level features together with their corresponding labels $(x_i^p, y_i^p)$, $i = 1, ..., l$, $x_i^p \in R^n$ and $y_i^p \in \{1, ..., N\}$, where $l, n, N$ denote the number of training instances, the dimension of low level features and the number of classes, respectively. The one-vs-all SVM classifier based on this part tries to solve the following optimization problem:

$$\begin{aligned} \min_{w_m, \xi_i^p} \quad & \frac{1}{2}\sum_{m=1}^{N} w_m^T w_m + C\sum_{i=1}^{l}\xi_i^p \\ s.t. \quad & w_{y_i^p}^T x_i^p - w_m^T x_i^p \geq e_i^{pm} - \xi_i^p, i = 1, ..., l \end{aligned} \tag{5}$$

where

$$e_i^{pm} = \begin{cases} 0, & \text{if } y_i^p = m \\ 1, & \text{if } y_i^p \neq m \end{cases}$$

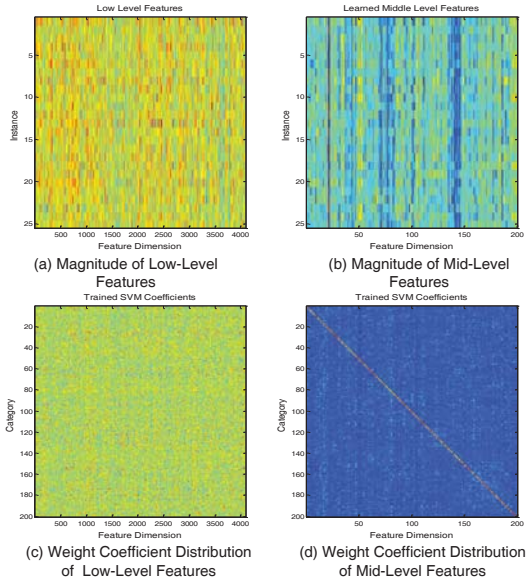


**Figure 4: The upper row shows the magnitude distribution of (a) the low level features and (b) the learned mid-level features for a given part, both from the same class. The lower row shows the corresponding SVM weights learned from the two kinds of features. Different colors represent different magnitudes, which are sorted in descending order and correspond to colors of red, yellow and blue. Note that in (a), the magnitude distribution is irregular while in (b) it exhibits regular stripes, especially the red line around dimension 25. The weights lie on the diagonal from upper left to lower right are dominant in (d), while the weights are irregular in (c).**

The decision function is

$$\arg \max_{m=1,...,N} \quad w_m^T x_i^p \tag{6}$$

The mid-level features are the projection of the low level ones based on the learned weights, which are the signed distance to the decision hyperplane. Denote the mid-level features of $x_i^p$ as $X_i^p = \{X_{i1}^p, X_{i2}^p, ..., X_{iN}^p\}$, each entry of $X_i^p$ are obtained according to the following equation:

$$X_{im}^p = \frac{w_m^T x_i^p}{\|w\|_2}, \quad m = 1, 2, ..., N \tag{7}$$

From the constrained conditions in Eq. (5), we can see that SVM classifier tends to fit the positive samples and makes the positive sample scores larger than that of the negative ones during every one-vs-all comparison in Eq. (6). Hence, for the mid-level feature $X_i^p$ (instance $i$ and part $p$) learned from the SVM classifier, $X_{iy_i^p}^p$ is usually larger than all the other entries in $X_i^p = \{X_{i1}^p, X_{i2}^p, ..., X_{iN}^p\}$. As shown in Fig. 4, the upper row of this figure shows an example of the magnitude distribution of the low level features and learned mid-level features for a given part, all belongs to the same sub-category. The magnitude of the low level features are orderless, while the learned mid-level features exhibit regular stripes, especially the one around feature dimension 25, the dark red line indicates that the magnitudes are large,

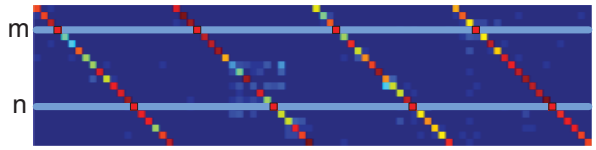


**Figure 5: Illustration of robustness of the one-vs-all mid-level features. The dominant weight coefficients lie in the diagonal locations for each part. The response is robust to the disturbance in the non-dominant locations.**

which means that this part is more likely belonging to the corresponding category. We simply concatenate the mid-level features part by part to obtain the representation of an image.

Next, another one-vs-all SVM classifier is trained based on these mid-level features. As Fig. 4(b) shows, for a given part $p$ the magnitudes at the specific positions are large and stable , which makes these mid-level features robust, so the model coefficients trained on the mid-level features are large at these positions. More specifically, for a given part $p$, the largest coefficients in the learned model $W^p_{(N\times N)}$ are focused on $\{w_{(1,1)}, w_{(2,2)}, ..., w_{(N,N)}\}$ if the training mid-level features are ordered. As shown in Fig. 4(d), The weight coefficients lie on the diagonal from upper left to lower right are large and we call these dominate weight coefficients. As a comparison, we also show the weight coefficients learned directly based on the low level features, which are shown in Fig. 4 (c), the coefficients are irregular and we can't find any dominant weight coefficients.

Finally, based on the learned mid-level features and the weight coefficients, the ultimate classifier assigns result which has the max response in decision function just like (6). Now we elaborate the robust advantage of this kind of representation and analyze in what situations misclassification would happen. We denote the weight coefficients learned from the mid-level features as $W_{N\times NP}$. Suppose a test instance belong to class $m$ with mid-level features $X \in R^{NP}$. Considering the following objective function:

$$\min (W_m^T - W_n^T)X \quad n \in \{1, 2, ..., N\}, n \neq m \tag{8}$$

If the objective value of Eq. (8) is positive, then we success this classification. Note that for weight vector $W_m$, $m \in \{1, 2, ..., N\}$, the most largest weight coefficients are focused on $\{W_{m,m}, W_{m,m+N}, ..., W_{m,m+(P-1)N}\}$, we call these dominant weight coefficients which contribute most to the final decision scores, as shown in Fig. 5. So the disturbance in the non-dominant entries would not influence the score $W_m^T X$ too much, hence manifest the robustness of this representation. On the other hand, the dominant weight coefficients for class $n$ lie in $\{W_{n,n}, W_{n,n+N}, ..., W_{n,n+(P-1)N}\}$, the equation would become negative when and only when disturbance happened in both of these two locations. At this situation, classes $m$ and $n$ are very similar, and most of their corresponding parts are similar and couldn't tell them apart.

## 4.3 Fused One-vs-All Mid-Level Features

The analysis above has demonstrated the fact that in some situations the discriminative features are subtle and extremely localized. These subtle features may be easily

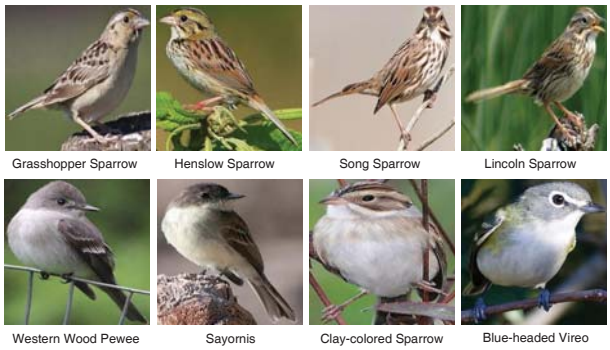


**Figure 6: The similar parts found by linear discriminant analysis. The upper row shares similar back while the lower shares similar breast. Note that the classes in the same row are all different. During the one-vs-all mid-level features learning, it is better to treat the classes with similar parts as one.**

swamped by plenty of irrelevant features. The misclassification would happen when the misclassified results are very similar with the correct ones. In the one-vs-all comparisons, we divide each part into $N$ different classes, where $N$ is the total number of classes. This kind of partition is not optimal since different classes may also share similar parts. It might bring into overfitting if we try to separate these very similar parts (actually they can't be distinguished only by the similar parts) and swamp discriminative features when recognizing classes sharing these similar parts. These observations suggest us to down weight the features which are not informative to discriminate similar classes. In this section, we propose a method which fuses the similar parts into a super-one to avoid dividing them into different classes, and learn one-vs-all mid-level features based on these fused classes. In order to fuse the similar classes into a bigger one, we first need to find the most similar classes for a given part. A standard $L_1$ or $L_2$ distance-based measurement is appealing for its simplicity, but considers all features to be equally important, which is unlikely to be a good idea. The low level features are high dimensional, some of them are not helpful to discriminate the classes. We wish to down weight the features that are not discriminative, and emphasize those that are. We apply a linear discriminate analysis to project the features into a low dimensional subspace and then measure the similarity between them. For consistency, we use the same notations as the last section. For a specified part $p$, given a set of part-level pairs $(x_i^p, y_i^p)$, $i = 1, ..., l$, $x_i^p \in R^n$ and $y_i^p \in \{1, ..., N\}$, denote the number of samples belonging to each class as $\{n_1, n_2, ..., n_N\}$. Linear discriminant analysis projects the $n$ dimensional features $x_i^p$ to $K (K < n)$ dimensional subspace via finding the optimal projection matrix $W = [w_1|w_2|...|w_K], w_i \in R^n, i = 1, 2, ..., K$ that minimize the ratio of within-class variance to between-class variance.

After projecting the features into $K$ dimensional subspace, for any given two classes, we measure the similarity of part features as

$$s_f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{9}$$

where $\mu_1$ and $\mu_2$ are the mean feature values for the two classes, and $\sigma_1$ and $\sigma_2$ are the corresponding standard de-

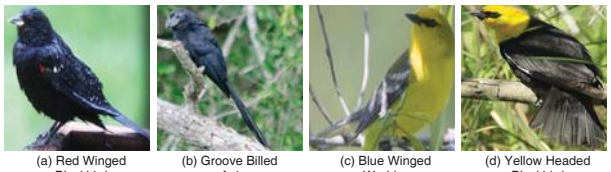


**Figure 7: Classification result comparisons between the one-vs-all and the fused one-vs-all mid-level features. The one-vs-all mid-level features correctly recognize (c) Blue Winged Warbler but misclassify (a) Red Winged Blackbird as (b) Groove Billed Ani. While the fused ones correctly recognize (a) Red Winged Blackbird but misclassify (c) Blue Winged Warbler as (d) Yellow Headed Blackbird.**

viations. We could get the similarity measure for any class pairs, and rank the similar classes for a given part. Fig. 6 shows an example of applying LDA similarity measurement to find the most similar classes for a given part. The upper are for part back and the lower are for part breast. All the classes in the same row are different, from which we can see that these parts are not discriminative to recognize the similar classes. As an alternative, we first fuse these similar classes into a bigger one, and learn the one-vs-all mid-level features based the fused classes.

# 5. EXPERIMENTAL RESULTS

## 5.1 Dataset

We test our method on CUB-200-2011 dataset, which is one of the most extensive datasets in fine-grained literature. It contains 11788 images spanning 200 sub-species, several of which share so tremendous similarities that even humans can't recognize them easily. Each image is labeled with its sub-species, a bounding box which encloses the object and at most 15 ground truth part annotations which indicate the locations of different parts of the images, *e.g.* beak, wings, throat and so on. Without loss of generality, we test our method in both situations: with and without part annotations. We make use of the provided bounding boxes as do most of the approaches we compared to. During Preprocessing, each image is resized to the same size, *i.e.* width and height do not exceed 500 pixels. We use the fixed train/test split provided in the dataset for all experiments. There are roughly 30 images per category to train and the rest for test. We mirror the train images to double the size of the training set, since it has been demonstrated an effective way to improve the classification accuracy to some extent. We also include results for Stanford Dogs [8], using similar experimental settings as for CUB-200-2011 Birds. For accuracy evaluation, we use the standard evaluation metric which is the average classification accuracy by category.

## 5.2 Implementation Details

We extract convolutional neural network features for its powerful classification capacity. The implementation [20] is used to extract CNN features. For a given part region (ground truth part or detected part), we crop the image to the minimal rectangle which enclose the given part. The output features of layer 21 are used for final representation of the given input part for its efficiency, which is a 4096 dimensional vector. In order to compute features for a part region, we must convert the image data in that region into a form that is compatible with the CNN (the structure requires inputs of a fixed $221 \times 221$ pixel size). Among the many possible transformations of arbitrary-shaped regions, we choose to resize the image to make the short side 221 and feed it to the CNN input. The output at this situation is no longer a 4096 dimensional vector, but $4096 \times n$ dimension. We chose max pooling on the multidimensional vector to get the feature vector with length 4096. Experiments show that this kind of resizing method gives higher performance than directly resizing it to $221 \times 221$ because it does not bring into aspect ratio deformation.

When it comes to encoding the images using Fisher vectors, we extract grayscale SIFT descriptors (the implementation [21] is adopted) at each part with a spatial stride of 5 and at four scales, defined by setting the width of the SIFT spatial bins to $12, 16, 20, 24$ pixels respectively. After extracting SIFT descriptors, we apply a PCA transformation to reduce the dimension to 80 and use a Gaussian mixture model with 128 components. In order to code the color information, we choose LLC [23] coded discriminative color descriptors [14] with a vocabulary of size 500. Both Fisher vectors and color features are $l_2$ normalized after encoding and then concatenated. For ease of expression, we denote Fisher vectors plus LLC coded color descriptors as Fisher vectors in the following experiments.

In situations with ground truth part annotations, since there are very few images in the dataset with 15 parts visible, according to the symmetry, we combine the left eye and right eye into one part, and apply the same operations on wings and legs. Thus we have 12 parts in total used for part-based recognition. In situations with detected parts, considering detecting more parts are inaccurate, we choose the same number of parts as in [6] to fix it at 4. For classification, we use a linear SVM classifier and the regularization strength is set by cross-validation.

## 5.3 One-vs-All vs Fused One-vs-All

We first compare the performance of our proposed two kinds of features, the one-vs-all and the fused one-vs-all mid-level features. The comparison is based on the ground truth part annotations and features extracted from convolutional neural network. By simply learning the one-vs-all mid-level features and performing another SVM classification, we could get an accuracy of 78.1%, which outperforms the state-of-the-art result [2]. One step further, we fuse the classes which share similar parts, and the number of classes decreases after fusing. We iteratively fuse the classes to a specified number according to their similarity rankings in Eq. (9) and learn the one-vs-all features based on the fused classes. For simplicity, each part has the same number of classes after fusing. The results are shown in Table 1. It is surprising that classification accuracy improves by 0.8% after the dimension for each part is reduced to 175 (which means the total dimension is 2275 after concatenation). It demonstrates that even the one-vs-all mid-level features (2600 dimension vector) are redundant to some extent, and the improvement lies in that some classes share similar parts and the fused method combines them into one to reduce the disturbance. The accuracy decreases when the fused number becomes smaller since the dimension is too small to classify correctly.

| Number | 200 | 175 | 150 | [200;175] | All |
|---|---|---|---|---|---|
| Accuracy | 78.1% | 78.9% | 77.0% | 79.6% | 78.3% |

**Table 1: Classification accuracies using the one-vs-all and the fused one-vs-all mid-level features.**

| Method | Accuracy |
|---|---|
| CNN (low level features) | 75.9% |
| CNN (mid-level features) | 79.6% |
| FV (low level features) | 72.3% |
| FV (mid-level features) | 75.8% |
| FV+CNN (low level features) | 77.1% |
| FV+CNN (mid-level features) | 81.2% |

**Table 2: Classification accuracies on Birds with ground truth annotations using different methods.**

At the last two columns in Table 1, we concatenate the mid-level features with the fused ones and get an accuracy of 79.6%, better than any single mid-level features. It implies us to use both features to complement each other. As shown in Fig. 7, the one-vs-all mid-level features could correctly recognize (c) *Blue Winged Warbler* but misclassify (a) *Red Winged Blackbird* as (b) *Groove Billed Ani*, mainly because the subtle discriminative features (red wings) are swamped by other irrelevant features. While the fused one-vs-all mid-level features (fused number 175) could correctly recognize (a) *Red Winged Blackbird* but misclassify (c) *Blue Winged Warbler* as (d) *Yellow Headed Blackbird* . By combining these two features, both classes can be recognized correctly. In the following experiments, we combine the one-vs-all and the fused one-vs-all mid-level features for classification. We only set the fused number as 175 , and concatenate them with the one-vs-all mid-level features.

## 5.4 Ground Truth Part Annotations Results

In this section, we first show the experimental results with ground truth part annotations. The concatenated mid-level features have dimension of 375 per part, so the total number of dimension is 4875. The classification results are shown in Table 2 for both CNN features and Fisher vectors. As a comparison, we also list the results directly based on the low level features, because the Fisher vector for one part is already high dimensional (tens of thousand), computing similarity based on Fisher vectors is time consuming, so we directly use the similarity measurement results based on the CNN features to compute the fused classes. From results in Table 2 we can see that the learned mid-level features can boost the classification accuracy by a large margin, achieving 3.7% and 3.5% improvements comparing with the results on the low level features respectively. The last line shows the result of combing Fisher vectors and CNN features, which gives another improvement and achieves an accuracy of 81.2%, which demonstrates that these two mid-level features are complementary with each other, as far as we know, it is the highest accuracy achieved so far.

## 5.5 Detected Part Locations Results

To enable automatic classification without the ground truth part annotations, we also perform experiments with detected parts. We segment the object into 4 parts, together with the whole image as an extra part, extracting features within each part, and learn the one-vs-all and fused one-vs-all mid-level features for classification. The results are listed in Table 3. We obtain an accuracy of 67.6% using both Fisher vectors and CNN features, which further demonstrates that the two features complement with each other. We also include the results on Sanford Dogs dataset in the last column of Table 3, and our mid-level features achieve 2.5% and 2.3% improvements comparing with the low level ones using CNN and Fisher vectors respectively, and get an accuracy of 53.5% when combining the two kinds of features.

| Method | Birds | Dogs |
|---|---|---|
| CNN (low level features) | 61.8% | 50.1% |
| CNN (mid-level features) | 65.3% | 52.6% |
| FV (low level features) | 58.7% | 44.8% |
| FV (mid-level features) | 61.2% | 47.1% |
| FV+CNN (low level features) | 62.9% | 50.9% |
| FV+CNN (mid-level features) | 67.6% | 53.5% |

**Table 3: Classification accuracies with detected parts using different methods.**

## 5.6 Time Complexity

Now we analyze time complexity of our proposed method. The training and testing based on the mid-level features is faster comparing with that on the low level ones. Taking the classification on the detected parts as an example, on our computer (Inter Core i7, CPU 3.2GHz), classification based on the mid-level features (dimension 4875) takes about 90s, mainly due to its low dimensionality, while direct classification based on the low level features costs about 190min for Fisher vectors (dimension around 100K) and 450s for CNN features (dimension around 20K). We need extra time to learn the mid-level features. For Fisher vectors, it spends about 8min in average for each part (dimension around 20K) and 40min in total, while the time on CNN features is 80s in average (dimension 4096) and 400s in total respectively. Note that the learning process is part by part, so the time consuming is far less than that based on the whole features.

## 5.7 Comparisons to Previous Works

Finally, we compare our method with existing works. Many researchers have reported classification results on Birds and Dogs dataset. The comparison results on Birds using ground truth part annotations are listed in Table 4, our method outperforms the state-of-the-art result (73.3%) by a large margin. To make fare comparisons, we use our mid-level features learning algorithm directly on the extracted features in [24], and obtain an improvement about 3% with an accuracy of 69.1%, which demonstrates the effectiveness of our method. Fig. 8 shows some easiest (accuracy 100%) and hardest (accuracy below 40%) classes of our method. Even with the help of ground truth part annotations, some subcategories are still challenging due to the existence of confusing counterpart, such as *American Crow* and *Fish Crow*, *California Gull* and *Herring Gull*. Exploiting methods overcoming these problems would be a challenging future work.

More works try to perform classification without the ground truth part annotations. We compare our method with these works under the same conditions, the results are listed in Table 5. It can be seen from the results that our method outperforms all the other results, partially benefit from convolutional neural network. The result in [10] could directly demonstrate our superior performance of using mid-level

| Method | Accuracy |
|---|---|
| Xie (hierarchical part matching)[24] | 66.35% |
| Xie (mid-level features) | 69.1% |
| Berg (POOF)[2] | 73.3% |
| Ours (FV+CNN mid-level features) | 81.2% |

**Table 4: Classification accuracies comparisons on Birds with ground truth part annotations.**

| Method | Birds | Dogs |
|---|---|---|
| Berg (POOF)[2] | 56.78% | - |
| Chai (symbiotic segmentation)[6] | 59.4% | 45.6% |
| E.Gavves (alignments)[13] | 62.7% | 50.1% |
| Zhang (deformable descriptors)[28] | 50.98% | - |
| Jia (DeCAF)[10] | 64.96% | - |
| Ours (FV+CNN mid-level features) | 67.6% | 53.5% |

**Table 5: Classification accuracies comparisons with detected part regions.**

features, which also extracts features from the convolutional neural network. By using the mid-level features, our method outperforms 2.7% comparing with their result. The improvement on Dogs is also obvious, our method exceeds the state-of-the-art result [13] by around 3.5%, to the best of our knowledge, this is the best accuracy reported so far in the literature.

At last, we compare our method with Berg's POOF [2] method, which also tries to learn some mid-level features for classification. Their features are also learned based on SVM classifier, but aim to learn one-vs-one mid-level features between any two classes for a given part. Though discriminative these features are, the shortcomings are that some entries are not informative. For example, if the binary classifier is trained based on class $i$ and $j$, the meaning of the scores for other classes are not so clear, and in most situations can't be used as discriminative features for other classes. On the other hand, since the one-vs-one part pairs are so large, in order to ensure effectiveness, it randomly samples some pairs and learns mid-level features based on these pairs, which may also decrease the accuracy. Different from the POOF method, we try to learn one-vs-all mid-level features at one time, which is more effective, and the fuse algorithm further boosts the discriminative power of our mid-level features. As a comparison, we learn POOF based on CNN features, the pairs are randomly sampled and the dimension is set as 5000, which is the same as [2]. The classification accuracy is 74.2%, which is comparable with Berg's but much lower than that of our method even without combing with Fisher vectors. Besides, our learning algorithm is independent of the feature extraction steps, and can be easily combined with other methods.

# 6. CONCLUSIONS

In this paper, we propose a novel method to learn one-vs-all mid-level features, which is more suitable for fine-grained visual categorization. It is dimension friendly since the dimension of the learned mid-level features is only related with the number of the classes and far less than that of the low level ones. The learned mid-level features are more compact representations of images since each entry of the features represents the signed distance to the corresponding class. The novel mid-level features are more robust than the low level ones. Considering different classes may share similar parts, we fuse classes with similar parts to super one and learn fused one-vs-all mid-level features based on the fused classes. We combine the two kinds of mid-level features into a powerful discriminative features for fine-grained visual categorization. Furthermore, we demonstrate the complementary characteristics of Fisher vectors and convolutional neural network features. Combining all the techniques we outperform the state-of-the-art results by a large margin on both Birds and Dogs dataset.

Our one-vs-all mid-level features are somewhat attribute features, which are similar with [16]. They are both mid-level semantic representations of the images. The fused one-vs-all mid-level features coincide with the characteristic of attributes that several classes share the same attributes. If we treat each class of a given part as an attribute, each entry of the mid-level features can be regarded as a continuous attribute value corresponding to that class. Different from the binary attributes extracted from [16] which need human annotations and have the form like black, water, eat fish, our features are relative attributes and don't need human interactions. Though overlapping to some extent (different classes may share similar attributes), they are applicable to detect unseen object classes, which needs to be verified in the future works.

# 7. ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation of China (NSFC) under contact No. U1201255, 61271218, 61228101 and 61128007, in part to Dr. Zhou by the Fundamental Research Funds for the Central Universities under contract No. WK2100060014, the start-up funding from the University of Science and Technology of China under contract No. KY2100000036, and in part to Dr. Tian by ARO grant W911NF-12-1-0057, and Faculty Research Award by NEC Laboratories of America, respectively.

# 8. REFERENCES

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. *Computer Vision and Pattern Recognition*, pages 2294–2301, 2009.

[2] T. Berg and P. N. Belhumeur. POOF: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. *Computer Vision and Pattern Recognition*, pages 955–962, 2013.

[3] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. *International Conference on Computer Vision*, pages 1–8, 2007.

[4] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. *Computer Vision and Pattern Recognition*, pages 2559–2566, 2010.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[6] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained

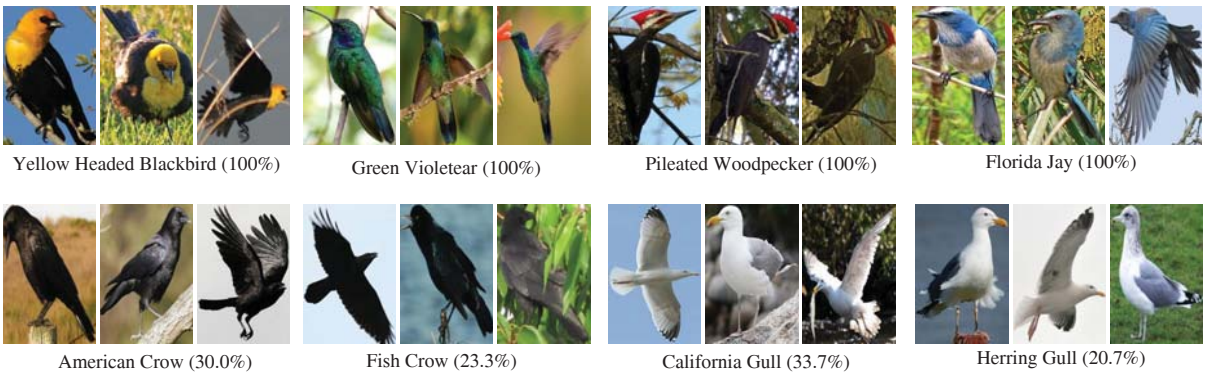


**Figure 8: Example classification results on Birds with ground truth part annotations. Top: some classes our method achieves** 100% **accuracy. Bottom: some classes our method performs poorly.**

categorization. *International Conference on Computer Vision*, pages 321–328, 2013.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, 1(1-22):1–2, 2004.

[8] E. I. D. Dataset. Novel datasets for fine-grained image categorization. *First Workshop on Fine Grained Visual Categorization, CVPR*, 2011.

[9] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. *Computer Vision and Pattern Recognition*, pages 580–587, 2013.

[10] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[11] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. *Computer Vision and Pattern Recognition*, pages 3474–3481, 2012.

[12] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[13] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. *International Conference on Computer Vision*, pages 1713–1720, 2013.

[14] R. Khan, J. Van de Weijer, F. S. Khan, D. Muselet, C. Ducottet, and C. Barat. Discriminative color descriptors. *Computer Vision and Pattern Recognition*, pages 2866–2873, 2013.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[16] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *Computer Vision and Pattern Recognition*, pages 951–958, 2009.

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[18] Y. Lu, L. Zhang, J. Liu, and Q. Tian. Constructing concept lexica with small semantic gaps. *IEEE Transactions on Multimedia*, 12(4):288–299, 2010.

[19] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. *Computer Vision and Pattern Recognition*, pages 3498–3505, 2012.

[20] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[21] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. *ACM Multimedia*, pages 1469–1472, 2010.

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. *Technical Report*, CNS-TR-2011-001.

[23] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.

[24] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang. Hierarchical part matching for fine-grained visual categorization. *International Conference on Computer Vision*, pages 1641–1648, 2013.

[25] L. Xie, Q. Tian, and B. Zhang. Spatial pooling of heterogeneous features for image applications. *ACM Multimedia*, pages 539–548, 2012.

[26] S. Yang, L. Bo, J. Wang, and L. G. Shapiro. Unsupervised template learning for fine-grained object recognition. *Advances in Neural Information Processing Systems*, pages 3122–3130, 2012.

[27] N. Zhang, R. Farrell, and T. Darrell. Pose pooling kernels for sub-category recognition. *Computer Vision and Pattern Recognition*, pages 3665–3672, 2012.

[28] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. *International Conference on Computer Vision*, pages 729–736, 2013.

[29] L. Zheng and S. Wang. Visual phraselet: Refining spatial constraints for large scale image search. *IEEE Transactions on Signal Processing Letters*, 20(4):391–394, 2013.