Fused One-vs-All Features With Semantic Alignments for Fine-Grained Visual Categorization

Xiaopeng Zhang, Hongkai Xiong, *Senior Member, IEEE*, Wengang Zhou, and Qi Tian, *Fellow, IEEE*

***Abstract*— Fine-grained visual categorization is an emerging research area and has been attracting growing attention recently. Due to the large inter-class similarity and intra-class variance, it is extremely challenging to recognize objects in fine-grained domains. A traditional spatial pyramid matching model could obtain desirable results for the basic-level category classification by weak alignment, but may easily fail in fine-grained domains, since the discriminative features are extremely localized. This paper proposes a new framework for fine-grained visual categorization. First, an efficient part localization method incorporates semantic prior into geometric alignment. It detects the less deformable parts, such as the head of birds with a template-based model, and localizes other highly deformable parts with simple geometric alignment. Second, we learn one-vs-all features, which are simple and transplantable. The learned mid-level features are dimension friendly and more robust to outlier instances. Furthermore, in view that some subcategories are too similar to tell them apart easily, we fuse the subcategories iteratively according to their similarities, and learn fused one-vs-all features. Experimental results show the superior performance of our algorithms over the existing methods.**

***Index Terms*— Part-based alignments, mid-level features, convolutional neural networks, image similarity.**

I. Introduction

AS AN emerging research topic, fine-grained visual categorization targets at discriminating typically hundreds of subcategories belonging to the same basic-level category. Applications include distinguishing different types of flowers, birds, and dogs, *etc.* It lies between the basic-level category classification (*e.g.*, categorizing bikes, boats, cars, and so on in PASCAL VOC [7]) and the identification of individual instances (*e.g.*, face recognition). A layperson can recognize basic-level categories like bikes or horses immediately since what often differentiates them is the presence or absence of some parts (*e.g.*, a bicycle has two wheels, while a horse has four legs). In contrast, fine-grained subcategories often share the same parts (*e.g.*, all birds should have wings, legs, *etc.*), and are often discriminated by subtle variations in the shape, texture, and color properties of these parts (*e.g.*, only the shape of beak or color of breast counts when discriminating similar birds). The challenge not only results from the subtle and localized inter-class differences, but also from the great intra-class variations. Taking the widely used fine-grained dataset CUB-200-2011 as an example (Fig. 1), it is not easy even for human beings to recognize them accurately.

Fig. 1. Sample images from CUB-200-2011, which shows (a) great intra-class differences and (b) small inter-class variations, it is hard even for humans to recognize them accurately.

Traditional Bag-of-Words [1] framework has been widely used in various applications [3], [4] due to its simplicity and effectiveness. However, with the ignorance of spatial layout information, it suffers severely from limited descriptive capability. A standard way to introduce weak geometry in Bag-of-Words representation is the use of spatial histogram [8], which defines pooling regions based on a uniform grid at predefined scales (typically the whole image, then quadrants, sixteenths, *etc.*). The spatial pyramid matching is effective for basic-level category classification since the composition of particular object or scene typically shares common layout properties, and there are plenty of clues that can distinguish them. However, it is at odds with that in the fine-grained domains due to the highly localized and subtle nature of distinguished features. Hence, localizing and

Manuscript received April 14, 2015; revised September 11, 2015; accepted December 2, 2015. Date of publication December 17, 2015; date of current version January 8, 2016. This work was supported by the National Science Foundation of China under Grant 61425011, Grant 61429201, Grant 61472378, and Grant U1201255. The work of W. Zhou was supported by the Anhui Provincial Natural Science Foundation under Contract 1508085MF109. The work of Q. Tian was supporetd in part by the Army Research Office under Grant W911NF-15-1-0290 and Grant W911NF-12-1-0057 and in part by the Faculty Research Gift Awards by NEC Laboratories of America and Blippar. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jean-Philippe Thiran.

X. Zhang and H. Xiong are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zxphistory@sjtu.edu.cn; xionghongkai@sjtu.edu.cn).

W. Zhou is with the CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System, University of Science and Technology of China, Hefei 230027, China (e-mail: zhwg@ustc.edu.cn).

Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: qitian@cs.utsa.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2509425

describing object's parts has become crucial for fine-grained recognition.

Most of previous works [19], [21], [22], [37] follow the idea of part-based localization. Among them the deformable part model (DPM) [10] is one of the most widely used template-based methods these years. While this kind of standard parametric model is suitable for structures that are relatively stable, it is insufficient to tackle the large variations and performs poorly for those highly deformable objects. As revealed by the detection performance [10] on PASCAL VOC 2010 challenge data, DPM obtains an average precision of only 13.1% on birds, far below that of bicycles (53.8%) and buses (53.4%) which are less deformable. On the other hand, Gavves *et al.* [26] propose to localize parts by roughly aligning objects with overall shape, and equally dividing the foreground object into several parts via ellipse fitting. However, this kind of alignment can easily fail since it does not take semantic information into consideration.

Features matter, and the choice of visual features for image representations may deserve the most important research in current state-of-the-art image classification. One of the greatest achievements during the past dozen years is the introduction of SIFT descriptors [24], and the improved aggregation methods from soft Vector Quantization [25], Locally-constrained Linear Coding (LLC) [38] to Improved Fisher Vector (IFV) [36]. Most of current classification tasks follow the pipeline of extracting basic low-level descriptors like SIFT, aggregating the descriptors into compact visual words and pooling the visual word histograms via Bag-of-Words for image representation [11]. These low-level features may not be optimal for specific classification tasks due to the well-known semantic gap [2] between low-level features and high-level image semantics. Furthermore, extremely high dimension of low-level features, especially Fish Vector, may lead to overfitting in a discriminative hyperplane spanned by the linear classifier, which degrades the classification accuracy and increases the computational complexity.

In the light of all these evidences, A new framework is proposed to cope with part localization and description in fine-grained domains. For part localization, the less deformable parts are first detected with template-based model, which can be regarded as semantic prior of the object. Then the other parts are obtained by geometric alignment of foreground mask under such semantic prior. The semantic prior is incorporated into geometric alignment, which enables more accurate part localization. For description, we learn One-vs-All Features [6], which are simple and transplantable, and the learned mid-level features are dimension friendly and more robust to outlier instances. Considering that some subcategories are too similar to tell them apart easily, we fuse them iteratively using Neighbor Joining method [43], and learn Fused One-vs-All Features (FOAF) based on these fused subcategories. Integrating all these techniques makes a powerful framework for fine-grained visual categorization, which outperforms the existing methods by a considerable margin.

This paper makes the following contributions:

1) We propose a new method for part localization, which incorporates semantic prior into geometric alignment. The semantic prior is obtained by detecting the less deformable parts with template-based model, while the geometric alignment is conducted on the foreground object mask with the guidance of semantic prior.

2) We propose to learn a range of mid-level features, which is called Fused One-vs-All Features (FOAF). Comparing with the low-level features, FOAF enjoys low-dimensionality, and is robust to outliers. Moreover, FOAF is ready to be integrated with existing techniques to further boost the performance.

3) We report classification results under different levels of automation (with and without object bounding boxes), in order to meet various requirements in real applications. To the best of our knowledge, previous works seldom focus on this situation (no object bounding boxes at both training and testing time).

4) We make an elaborative comparison to uncover the effect of part localization, and demonstrate the effectiveness of the proposed part localization scheme. Besides, we reveal that excessive parts, though accurate, are harmful for recognition.

The rest of this paper is organized as follows. Section II briefly reviews related works on fine-grained visual categorization. Section III overviews the proposed framework. In Section IV, we elaborate the proposed semantic and geometric alignment method. The detailed algorithm of learning One-vs-All Features are presented in Section V. Experimental results and discussion are shown in Section VI and VII, respectively. Finally, we draw conclusions in Section VIII.

II. Related Work

Fine-grained visual categorization is a challenging problem and has recently emerged as a hot research topic. There has been a great number of works dealing with different species of birds [21], [22], dogs [19], and cats [16], *etc*. Based on the vision tasks, we organize the discussion related to fine-grained recognition with two aspects: part localization and feature representation.

A. Fine-Grained Part Localization

The detection of objects in fine-grained domains ranges from template-based models to exemplar-based models. Inspired by the pictorial structure [9] which provides a powerful framework for representing objects by non-rigid constellations of parts, many methods have been applied to jointly localize geometrically related parts. Among them, DPM [10] has become one of the most effective template-based approaches to date. In the scenario of fine-grained categorization, it also conveniently enables part-based approaches since objects belonging to same basic-level often share the same parts. Chai *et al.* [19] utilized simultaneous segmentation (Grab-Cut [13]) and detection (DPM) to improve integrated accuracies. Yang *et al.* [17] accomplished unsupervised learning of DPM to find discriminative parts for fine-grained categorization. Since DPM largely relies on the careful initialization of parts, [37] and [41] adopted strongly supervised DPM [23] with additional part-level supervision to construct better, class-specific object models. For highly deformable objects such as birds and dogs, however, DPM and

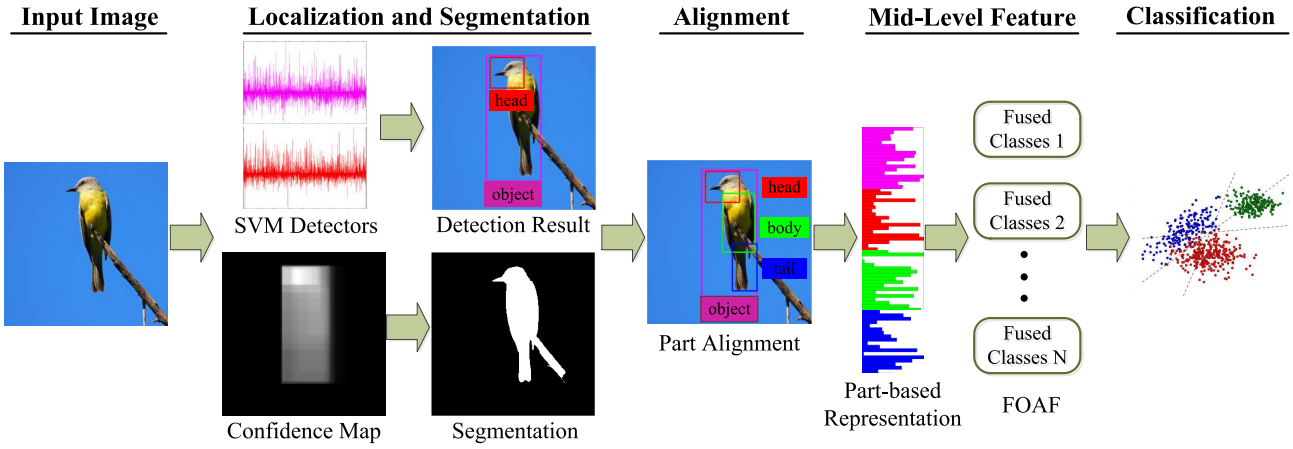


Fig. 2. Flowchart of the proposed fine-grained classification framework. Given an input image, we first detect the less deformable part (such as the head of birds) as well as the object using template-based detectors. The confidence map is obtained via weighting top scored detection locations of object and used as segmentation prior. Based on the localization and segmentation results, we perform part alignment and obtain several semantic meaningful parts, such as "head", "body", and "tail". We learn mid-level features according to the part-based low-level representation for classification. FOAF is obtained via fusing similar subcategories into macro-classes, and learning One-vs-All Features with SVM.

other template-based models are even inferior to simple Bag-of-Words models since they suffer a much weaker notion of geometry.

Inspired from the exemplar-SVMs [20] which bridges parametric and nonparametric models, transferring information from training to test images has been successfully used in several applications. The exemplar models do not learn template-like detectors for individual parts, instead, they localize distinctive details by roughly aligning the objects. Gavves *et al.* [26] developed a supervised alignment method which retrieved nearest neighbor training images for a test image, and regressed the locations from these neighbors to get predicted parts. Similarly, Berg and Belhumeur [21] proposed to automatically detect part locations using a consensus of exemplars. Recently, Christoph *et al.* [39] suggested to transfer part annotations from objects via performing a simple but very powerful global matching and a subsequent ensemble learning.

B. Fine-Grained Representation

For the description of fine-grained objects, different proposals have been made in literature. The most widely used descriptors are color SIFT, gray SIFT plus color histogram [19], [26]. A common characteristic of these descriptors is that they can be largely handcrafted. Namely, all of them comprise dense sampling of local image patches, describing by means of low-level visual descriptors, encoding into a high-dimensional representation, and pooling over images. Since single descriptor might fail to capture the rich information within local pathes, it is reasonable to fuse multiple descriptors [37] for compensation. Boureau *et al.* [5] learned semantic representations of images by aggregating neighboring descriptors to form micro-features or visual phrases. Gao *et al.* [27] learned category-specific dictionary for each category and shared-dictionary for all the categories. The category-specific dictionaries encode subtle visual differences among different categories, while the shared-dictionary encodes common visual patterns among all the categories. However, they are *de facto* low-level features, and suffer from dimension dilemma.

Recently, these hand-crafted descriptors have been substantially outperformed by the features learned with convolutional neural networks [18], which have a more complicated structure than traditional representations. They contain several layers of non-linear feature extractors, and are considered 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 results on a large of benchmarks. Though not specifically designed to model subcategory level differences, it has been demonstrated [15] that convolutional features capture such information well and obtain the state-of-the-art results for fine-grained categorization so far.

III. Framework Overview

Fig. 2 shows the diagrammatic flowchart of the proposed approach, which consists of two modules - part alignment (Sec. IV) and Fused One-vs-All Features (FOAF) learning (Sec. V). Different from previous works which assume that object bounding boxes are provided, we tackle the problem in a more general case - *no bounding boxes provided at both training and testing time*.

Given an input image, we first detect the less deformable part (such as the head of birds) as well as the object with R-CNN [28], respectively, followed by a geometric refinement which restricts the detected head within the region of detected object. In order to segment foreground from background without object bounding boxes, we compute object confidence map which denotes the possible locations of foreground object, and works as prior for subsequent segmentation. The final alignment is under the guidance of the detected head and segmented foreground mask, which guarantees the accurate part alignment.

For FOAF learning, we extract features based on the aligned parts, and train one-vs-all SVM classifiers to obtain

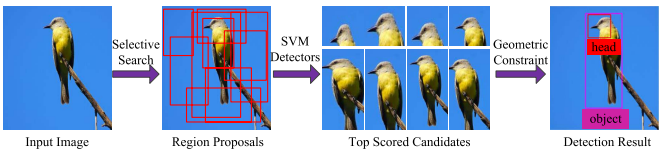


Fig. 3. Overview of the object and head detection method. Given an input image, we (1) extract region proposals using selective search [30], (2) obtain top scored candidates for object and head using the trained SVM detectors and (3) perform geometric constraint to update the detection results.

mid-level features. Furthermore, similar subcategories are fused iteratively using the Neighbor-Joining [43] method and treated as micro-classes, then FOAF is learned according to the fused subcategories. We show that FOAF is more powerful than traditional features for classification.

IV. Semantic and Geometric Alignment

For animals such as birds and dogs, extreme articulations, atypical viewpoints, and partial occlusions induce variations of the appearance that cannot be well captured by a template-based detector. Hence, it is inappropriate to model and detect each part of such highly deformable objects. Fortunately, the head is demonstrated to be distinctive and can be detected very reliably by template-based detectors [16]. Inspired by this observation, we only detect the head of animals, and bypass other parts (*e.g.* the wings and legs of birds) which are highly deformable. Different from [16] which makes use of the detected head to localize the whole object via homogenous color and texture propagation, we align the rest parts by way of the head prior and foreground segmentation. The semantic and geometric alignment method includes three steps, *i.e.*, object and head detection, object confidence map generation, and consistent part alignment.

A. Object and Head Detection

The overall detection framework is illustrated in Fig. 3. Given an input image, the whole object (in case object bounding box is not provided) and head are first detected. Since object and head are detected independently and do not incorporate any knowledge about how they should be constrained geometrically, the detected head may discord with the object and even respond entirely outside the detected object. To tackle this issue, we return several high scored regions for object and head, and identify the best detections by introducing geometric prior which constrains head responses within the object.

1) Model Fine-Tuning and Detector Learning: To adapt the CNN pretrained on ImageNet to the fine-grained detection task, we continue stochastic gradient descent (SGD) to fine-tune the network as [28]. Without loss of generality, the annotated external data of PASCAL VOC [23] are chosen for network fine-tuning. There are about 1100 annotated images for birds and 2000 images for dogs. Fig. 4 shows some example images with object and head annotations for birds and dogs. The network is fine-tuned for object and head, respectively.

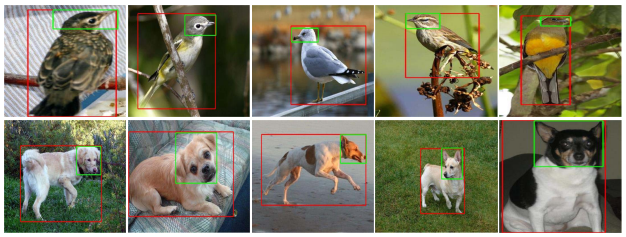


Fig. 4. Example figures in PASCAL VOC with corresponding object and head annotations, the top row for birds and the bottom row for dogs.

For detector learning, features are extracted from the training samples with the fine-tuned network. Only the ground-truth boxes are treated as positive examples and proposals with intersection-over-union (IoU) overlap below 0.3 are treated as negative ones. The others with IoU overlap between 0.3 and 1.0 are discarded. We independently optimize linear SVM classifiers for head and object, and obtain two kinds of detectors $\{\mathbf{w_0}, \mathbf{w_1}\}$, respectively.

2) Geometric Constraint: Denote $\mathbf{X} = \{\mathbf{x_0}, \mathbf{x_1}\}$ as the top scored candidates for object p_0 and head p_1, and $\phi(x_0)$, $\phi(x_1)$ as corresponding features. Given the detectors $\{\mathbf{w_0}, \mathbf{w_1}\}$ for object and head, the detections are refined via solving the following optimization problem:

$$\arg\max_{\mathbf{X}} \quad \Psi(\mathbf{w_0^T}\phi(\mathbf{x_0})) + [\lambda]_\epsilon \Psi(\mathbf{w_1^T}\phi(\mathbf{x_1})) \tag{1}$$

where

$$\Psi(z) = \frac{1}{1+e^{-z}}, \quad \textit{and} \quad [\lambda]_\epsilon = \begin{cases} \lambda, & \text{if } \lambda \geq \epsilon \\ 0, & \text{if } \lambda < \epsilon \end{cases} \tag{2}$$

where $\Psi[\cdot]$ is a nonlinear function which maps the score to range $[0, 1]$, and $[\cdot]_\epsilon$ is a hinge loss function at ϵ (which is 0.8 in our experiment). The parameter λ measures the ratio of overlapping region (intersection between head p_1 and object p_0) to head p_1, with range $[0, 1]$. The overall score is a weighted sum of object and head scores. The parameter $[\lambda]_\epsilon$ is introduced to penalize the inconsistency between the head and object. In case that the overlap is less than 0.8, the second term becomes zero, thus lowers the overall score. The goal is to find the detection pair which maximizes Eq. (1).

B. Object Confidence Map

In realistic scenarios object bounding box is unavailable, which makes it difficult to segment the foreground from the background. To overcome this issue, object confidence map is computed and used as prior for the subsequent segmentation.

The object confidence map indicates the possible locations of object in an image. According to the geometric constraint 1, we return several top scored pairs $\{X_i^{x_0}, X_i^{x_1}\}$ with corresponding detection scores $\{S_i\}$. These pairs indicate the most possible locations of object and head. We do not try to obtain accurate object bounding box from these possible locations. Instead, we collect these most possible locations via soft voting to obtain object confidence map O, which can be represented as

$$O(p) = \frac{\sum_i S_i(X_i^{x_0}(p) + X_i^{x_1}(p))}{Z} = \frac{\sum_i S_i(X_i^{x_0}(p) + X_i^{x_1}(p))}{\sum_i S_i(X_i^{x_0} + X_i^{x_1})} \tag{3}$$

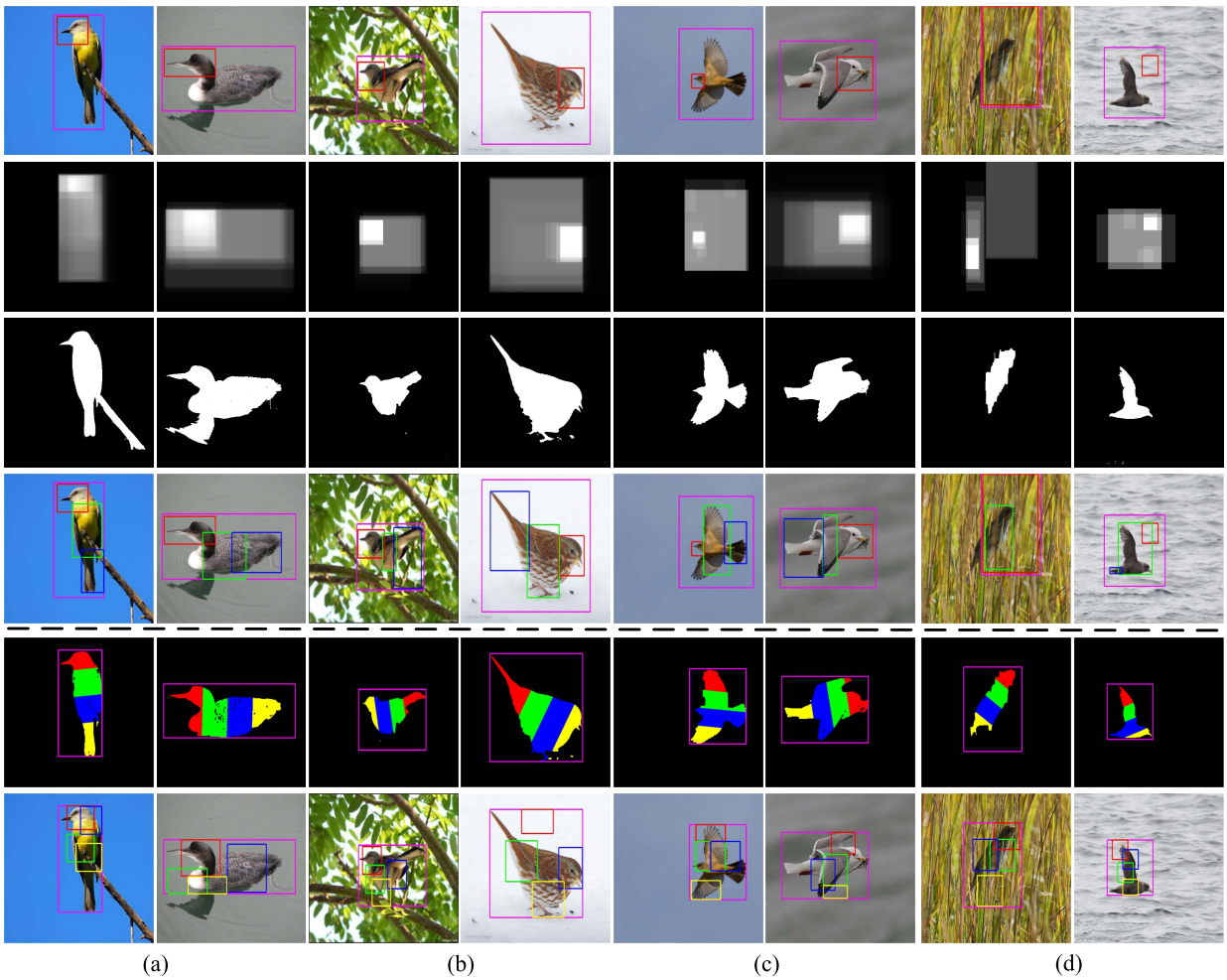


Fig. 5. Some consistent part alignment results (fourth row) of the proposed method. For completeness, the first row shows refined detection results (Eq. (1)) of object and head, and the second row is the generated object confidence maps. GrabCut segmentation with object confidence prior is shown in the third row. Our method incorporates semantic prior into foreground object alignment, which is robust in different situations including (a) easy case (b) irregular deployment and (c) highly deformable case. The last two columns show some failure cases of the proposed method. For comparisons, the last two rows exhibit other alignment methods in [19] and [26]. Note that these two methods suppose object bounding boxes are provided at both training and testing time, while our method does not require these fussy annotations. Best viewed in color.

where the binary variable $X_i(p)$ indicates whether the i-th bounding box (object or head) contains the pixel p or not, and the score S_i indicates the confidence coefficient. The value Z is a constant for normalization which enables the maximum value in object confidence map equal to 1. Each pixel in the object confidence map O indicates the probability of containing the object, and can be used as spatial prior for the following segmentation. Some example object confidence maps are shown in the second row of Fig. 5.

After obtaining object confidence map, we proceed with GrabCut segmentation [13]. GrabCut segmentation groups pixels with similar appearance together via Gaussian mixture model, such that the foreground is separated from the background. The foreground model is estimated from the pixels with object confidence higher than a threshold, set to the 95% quantile of the confidence distribution in the image, and the background model is estimated from the pixels with confidence smaller than 30% quantile. Sample segmentation masks are shown in the third row of Fig. 5. Although GrabCut is not always accurate and in rare cases fails to recover a basic contour, in the vast majority it is able to return a rather precise contour of the object.

C. Consistent Part Alignment

The head is successfully detected by template-based methods due to its stable property. However, for highly deformable parts such as the body of birds, it is not preferred to locate these regions with such template-based methods. In this section, a geometric partition method is proposed for highly deformable part alignment.

The principle is based on the fact that all subcategories in fine-grained domains share similar global characteristics after pose normalization. The corresponding parts can be obtained by consistently dividing the aligned object. Based on the segmentation, we compute the centroid of the foreground object, which is obtained by averaging the coordinates of foreground pixels. It is relatively stable to random fluctuations. Together with the center of the detected head, we obtain

two centers of mass. The line connecting the two centroid points are regarded as the semi-principal axis of object, which is similar with the "spine" of object, and each anatomical part is arranged along the principal axis in order. Specifically, starting from the intersection point between head bounding box and semi-principal axis, the foreground object is divided into two parts along the principal axis, and could capture, for example, the "body" and "tail" of birds. Together with the detected head, we have three parts in total. In order to be compatible with the CNN input, all pixels within each part are enclosed with a minimal rectangular. The method imposes semantic prior when performing geometric alignment, which can be regarded as an improved method of SPM [8] and ellipse fitting [26].

Fig. 5 shows some alignment examples. The first row shows the input images with detected object and head, while the second row illustrates the corresponding object confidence maps. GrabCut segmentations under object confidence map prior are shown in the third row. The fourth row shows the alignment results. Besides, the last two rows show the results of [26] (ellipse fitting) and [19] (symbolic segmentation) for comparison. Ellipse fitting method simply localizes parts by roughly aligning objects with the overall shape, while symbolic segmentation resorts to template-based models. We illustrate three kinds of situations, which are denoted as (a) easy case, (b) irregular deployment, and (c) highly deformable case. The alignment regions are labeled with different colors, *i.e.* red, green, and blue in order (object marked with magenta bounding box). For easy case when segmentation is accurate and object is less deformable, all three methods perform well. In the latter two cases, the other two methods completely fail to locate parts. However, our method is relatively robust to these situations and performs well. The last two columns show some failure cases. Note that different from the two methods which assume object bounding boxes are provided, our part localization method is fully automatic.

V. Fused One-vs-All Features

Based on the above aligned parts, different features (*e.g.* SIFT, HOG, and CNN, *etc.*) can be extracted from each part. The image I can be represented as a set of region features:

$$D = \{(\mathbf{f}_1, R_1), (\mathbf{f}_2, R_2), \ldots, (\mathbf{f}_M, R_M)\} \tag{4}$$

where $\mathbf{f}_i$, R_i, $i \in \{1, 2, \ldots, M\}$ denote the ith feature vector and the occupied region, respectively, and M denotes the total number of regions.

Intuitively, we could get the representation of image by simply concatenating different part features into a long vector, and train a classifier for recognition. Such a pipeline has served as a routine for most classification tasks. 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 high as tens of thousands, and even higher when concatenating different parts into a longer one) but also entry meaningless, from which we could not interpret what each entry of the features means. Furthermore, with these high-dimensional features, a discriminative hyperplane can be easily obtained even using a linear classifier, but it may easily introduce overfitting and perform poorly on test data. To tackle these issues, we learn a new kind of mid-level features, which is called Fused-One-vs-All Features (FOAF). The FOAF is dimensional friendly, semantic meaningful, and robust for classification.

A. One-vs-All Features

Our method requires low-level features extracted from the same part of different objects, and annotated with class labels. We refer to low-level features as those directly extracted from the images, hence SIFT, HOG, and CNN can all be treated as low-level features. To enhance the semantic representation for accurate recognition, it is intuitive to learn mid-level features since they share more semantic meanings than the low-level ones. Given the reference dataset, the training set consists of images belonging to N classes $\{1, \ldots, N\}$ and includes M parts $\{1, \ldots M\}$. The one-vs-all mid-level features are learned as follows:

1. Selecting any part $p \in \{1, \ldots, M\}$ from the objects, and all the none zero features corresponding to part p, these features are denoted as $\mathbf{f}^p$. We learn a one-vs-all SVM classifier based on the part features $\mathbf{f}^p$, and project $\mathbf{f}^p$ 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 in the definition.

2. According to the projection transformation, all the low-level part features are mapped into a new mid-level feature space. These mid-level features are concatenated part by part after normalization.

3. For a test image, extract low-level features from part p, and project the corresponding part features to N dimensional vector according to the learned SVM weights. Concatenating the mid-level features in the same way as the training ones.

After a simple projection, all the part features are projected to the mid-level feature space. The advantages of the learned mid-level features over the low-level ones are as follows:

1) 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, the total dimension of the mid-level features is MN, which is only several thousand in most situations. In contrast, the dimension of the low-level features, such as Fisher Vector, can be as high as hundreds of thousands. We will demonstrate the super-performance of the learned mid-level features in the following sections.

2) Comparing with the low-level features which are entry meaningless, every entry of the mid-level features is semantic-aware. For example, given a reference image x, and denote the mid-level features for part p as $\mathbf{X}^p = \{\mathbf{X}_1^p, \mathbf{X}_2^p, \ldots \mathbf{X}_N^p\}$, The score $\mathbf{X}_i^p (i \in \{1, \ldots N\})$ represents the signed distance from current image to category i for part p. The larger the value $\mathbf{X}_i^p$ is, the more similar of image x with category i. This kind of representation is very helpful and understandable for classification.

B. Deep Insight Into One-vs-All Features

In this section, we elaborately clarify the classification process of One-vs-All Features. For a certain part $p \in \{1, \ldots, M\}$, given a set of part-level features together with their corresponding labels $(\mathbf{x}_i^p, y_i^p)$, $i = 1, \ldots, l$, $\mathbf{x}_i^p \in R^n$, and $y_i^p \in \{1, \ldots, N\}$, where l, n, and 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 tries to solve the following optimization problem:

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

where

$$e_i^{pm} = \begin{cases} 0, & \text{if } y_i^p = m \\ 1, & \text{if } y_i^p \neq m. \end{cases} \tag{6}$$

It can seen that SVM classifier tends to fit the positive samples and makes the positive sample scores larger than that of the negative ones. The One-vs-All Features are the projection of the low-level ones, which are the signed distance to the decision hyperplane. Denote the mid-level features of $\mathbf{x}_i^p$ as $\mathbf{X}_i^p = \{\mathbf{X}_{i1}^p, \mathbf{X}_{i2}^p, \ldots, \mathbf{X}_{iN}^p\}$, each entry of $\mathbf{X}_i^p$ is obtained according to the following equation:

$$X_{im}^p = \frac{\mathbf{w}_m^T \mathbf{x}_i^p}{\|\mathbf{w}_m\|_2}, \quad m = 1, 2, \ldots, N, \tag{7}$$

hence, for the mid-level features $\mathbf{X}_i^p$, $\mathbf{X}_{iy_i^p}^p$ is usually larger than all the other entries. As shown in Fig. 6, the upper row is an example of magnitude distribution of the low-level features and learned mid-level ones for a given part. Different colors represent different magnitudes, which are sorted in descending order and correspond to colors of red, yellow, and blue. 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, which means that this part is more likely to belong to the corresponding subcategory.

Then, another SVM classifier is trained based on these mid-level features for final classification. As Fig. 6(b) shows, the magnitudes at the specific dimensions are large and stable, so the coefficients trained on the mid-level features are large at these dimensions. More specifically, for a given part p, the largest coefficients in the learned model $\mathbf{W}_{(N\times N)}^p$ are focused on $\{W_{(1,1)}, W_{(2,2)}, \ldots, W_{(N,N)}\}$. As shown in Fig. 6(d), The weight coefficients lying on the diagonal from upper left to lower right are large and regarded as dominant weight coefficients. As a comparison, Fig. 6(c) shows the weights learned directly based on the low-level features, the coefficients are irregular and it is difficult to find any dominant weight coefficients.

Given a test image with One-vs-All Features $\mathbf{X} \in R^{MN}$. Denote the weight coefficients learned from One-vs-All Features as $\mathbf{W}_{N\times MN}$ (an example is shown in Fig. 7, and $M = 4$). One-vs-All Features are robust for two reasons:

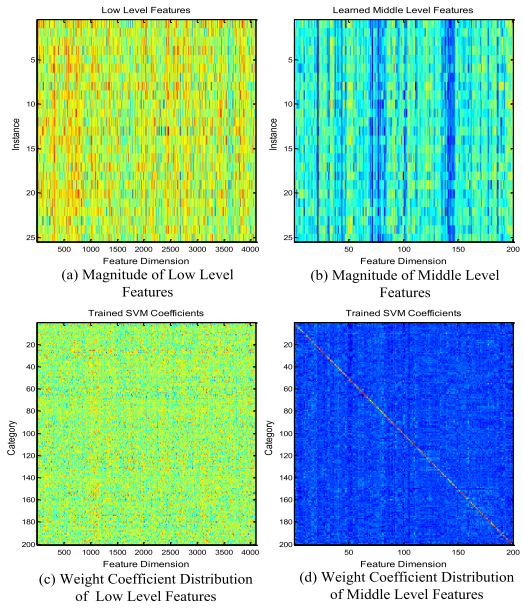


Fig. 6. The upper row shows the magnitude distribution of (a) the low-level features and (b) the learned mid-level features for a given part, both from the same class. The lower row shows the corresponding SVM weights learned from this two kinds of features. Different colors represent different magnitudes, which are sorted in descending order and correspond to colors of red, yellow, and blue. Note that in (a), the magnitude 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).

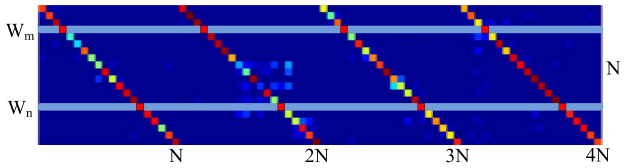


Fig. 7. Illustration of the robustness of One-vs-All Features. The dominant weight coefficients lie in the diagonal locations for each part.

1) In terms of the mid-level features, each entry of One-vs-All Features is obtained by the weighted sum of the low-level features, which emphasizes significant entries and suppresses nonsignificant entries. Thus One-vs-All Features are roust to the disturbance in the nonsignificant low-level entries.

2) In terms of the final classification, note that for weight vector $\mathbf{W}_m$, the dominant coefficients lie in $\{\mathbf{W}_{m,m}, \mathbf{W}_{m,m+N}, \ldots, \mathbf{W}_{m,m+(M-1)N}\}$, which contribute most to the final decision scores, while the dominant coefficients for class n lie in $\{\mathbf{W}_{n,n}, \mathbf{W}_{n,n+N}, \ldots, \mathbf{W}_{n,n+(M-1)N}\}$. If there exists some parts which can tell class m from class n, the score differences based on these parts are significant, and the score corresponds to the correct class increases. In other words, classification is still successful when a small amount of parts are not distinguishable.

One-vs-All Features include two stages of SVM training, and can be regarded as "deep" SVM to some extent. Recent researches on convolutional networks [18] have demonstrated that deep representation really counts for objet recognition.

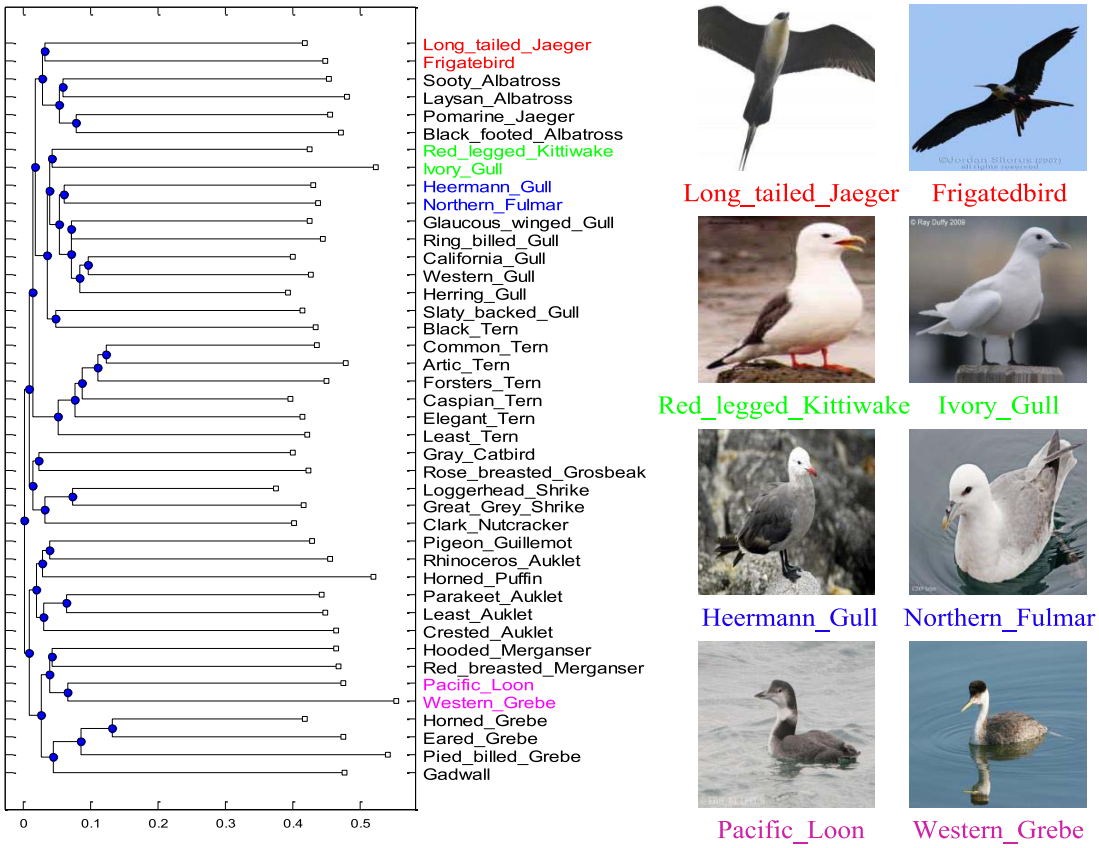


Fig. 8. A subset of "tree of similarity" obtained by Neighboring Joining method. From the tree we find that it can generally discover subcategories which are close in terms of animal taxonomy (such as Terns and Gulls), but there are exceptions. We show some example pairs which are close in terms of "tree of similarity", but not close in terms of animal taxonomy. Such cases may be examples of convergent evolution, in which two different species independently evolve into similar traits.

Although we can perform multi-stages of One-vs-All Features learning iteratively, experimental results show that two stages of SVM training suffices.

C. Fused One-vs-All Features

A fundamental problem in fine-grained recognition is how to handle subcategories that are nearly indistinguishable. In the bird world, an example of this problem is *Terns*, in CUB-200-2011, there are seven kinds of subcategories all belonging to *Terns*, and sharing similar appearance (see Fig. 1). If we regard them as independent subcategories respectively, say *Common Tern*, and train a discriminative one-vs-all classifier in usual way, the negative set would include other *Terns* which are very similar with the positive *Common Tern*. A classifier in this situation is very likely to latch on to accidental features that distinguish *Common Tern* from other *Terns* only in this particular training set and de-emphasize significant features that distinguish *Terns* from non-*Terns*.

To mitigate this issue, we fuse these similar subcategories into a bigger one, and learn Fused One-vs-All Features. Towards this goal, we first need to measure similarity between any two classes. A direct distance-based measurement is appealing for its simplicity, but considers all features to be equally important, which is unlikely to be optimal. The low-level features are high-dimensional, and some of them are not helpful to discriminate the classes. We expect to suppress the features that are not discriminative, and emphasize those that are. A standard tool for this goal is linear discriminant analysis. For a specified part p, given a set of part-level pairs $(\mathbf{x}_i^p, y_i^p)$, $i = 1, \ldots, l$, $\mathbf{x}_i^p \in R^n$, and $y_i^p \in \{1, \ldots, N\}$, linear discriminant analysis projects the n dimensional features $\mathbf{x}_i^p$ to K $(K < N)$ dimensional subspace via finding the optimal projection matrix $\mathbf{W} = [\mathbf{w}_1|\mathbf{w}_2|\ldots|\mathbf{w}_K], \mathbf{w}_i \in R^n, i = 1, 2, \ldots, K$ that minimizes the ratio of within-class variance to between-class variance. After projecting the features into K dimensional subspace, for any two classes, the similarity is measured as

$$s_f = \frac{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}{\sigma_1^2 + \sigma_2^2} \tag{8}$$

where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are mean feature values for the two classes, and σ_1 and σ_2 are the corresponding standard deviations.

The next is to fuse classes according to the similarities. Towards this goal, a Neighbor Joining [43] method is adapted for its simplicity and robustness. The Neighbour Joining method is a bottom-up clustering method for the creation of phylogenetic trees, which is widely used in bioinformatics. It computes the lengths of the branches based on the knowledge of distances between pairs of taxa. In each stage,

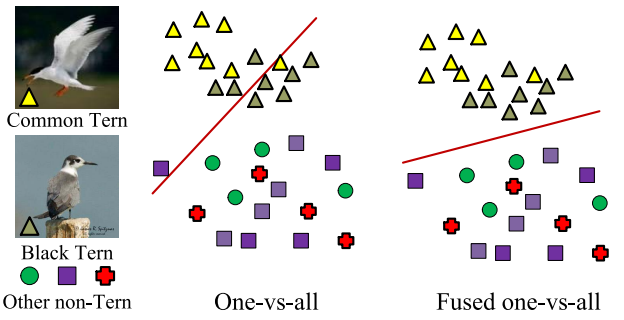


Fig. 9. A comparison of one-vs-all and fused one-vs-all classifier. One-vs-All Features regard similar subcategories such as *Common Tern* and *Black Tern* as two classes, and corresponding classifier latches on to accidental features that distinguish them but de-emphasize significant features that distinguish *Terns* from none *Terns*. However, Fused One-vs-All Features regard these two subcategories as one, and could obtain more discriminative classifier.

the two nearest nodes of the tree are chosen and defined as neighbours. This is done recursively until all of the nodes are paired together. The trees are constructed from the perspective of evolutionary, which with a root represents the origin of life, and a leaf for every extant species or evolutional dead end. Species close to each other in the tree are in the sense more similar than species that are not close. Inspired from the construction method in terms of phylogenetic similarity, we replace the distance metric with the visual similarity calculated in (8), and produce a tree in terms of visual similarity which is denoted as "tree of similarity".

Comparing with the animal taxonomy which is based on a combination of the fossil evidence, habitats, and genetic data, *etc.*, the "tree of similarity" is generated automatically. However, we find that such automatic method tends to rediscover taxonomies. Fig. 8 illustrates a subset of "tree of similarity" obtained by Neighboring Joining method. From the tree we find that it can generally discover subcategories which are close in terms of animal taxonomy (such as Terns and Gulls), but there are exceptions. The right column shows some example pairs which are close in terms of "tree of similarity", but not close in terms of animal taxonomy. Such cases may be examples of convergent evolution, in which two different species independently evolve into similar traits.

Fig. 9 illustrates the advantage of Fused One-vs-All Features over One-vs-All Features. Two subcategories *Common Tern* and *Black Tern* are very similar with each other. One-vs-All Features try to discriminate *Common Tern* from *Black Tern*, but suffer from overfitting on training set and perform poorly on testing set. However, when regarding *Common Tern* and *Black Tern* as one macro-class, the separating plane is more discriminative than that of the one-vs-all classifier.

FOAF is most related to the one proposed by Berg and Belhumeur [21]. There POOF, which is defined by specifying two parts, one for feature extraction, and another for alignment, followed by training a one-vs-one classifier. However, POOF makes use of SVM weights to learn discriminative masks, which is sensitive to misalignment, and is high-dimensional and redundancy since it learns features for every pair of classes and every pair of parts. Comparing with POOF, FOAF learns a one-vs-all classifier only once for each part, which is more natural, and significantly more efficient.

FOAF is somewhat an attribute feature [12], in the sense that both are mid-level semantic representations of images. FOAF coincides 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 [18] which need human annotations and have the form like "black", "water" and "eat fish", FOAF reflects relative attribute and does not need extra human interactions.

VI. Experimental Results

A. Datasets

In this section, we evaluate our proposed method on four publicly available fine grained animal datasets: *i.e.* CUB-200-2011 [35], Stanford Dogs [29], Oxford-IIIT Pets [16], and Columbia Dogs [32]. We use the default training/test split and follow the evaluation protocol of the corresponding paper. We not only report results with object bounding boxes provided (if provided, such as CUB-200-2011 and Stanford Dogs) as most previous works we compare to, but also consider a more general situation when *object bounding boxes are not provided at both training and testing time*. To the best of our knowledge, there are few related works reporting in this situation. Last, note that only for CUB-200-2011 there exists ground truth part locations as well as ground truth object bounding boxes. Therefore, for experiments where ground truth part locations are needed, we only report results on this dataset.

B. Network Fine-Tune

Network fine tune has been demonstrated an effective way to boost recognition performance [28]. We fine-tune the network to adapt it to the specific fine-grained task. Aside from replacing the network's ImageNet-specific 1000-way classification layer with a randomly initialized C-way layer (C denotes the number of sub-categories to be classified, *e.g.* 200 for CUB-200-2011), the architecture is unchanged. In case of ground truth bounding box is given, we follow the setting as suggested in [28], region proposals with IoU overlap above 0.5 with ground truth bounding box are treated as positives for that box's class and the rest as negatives. However, when ground truth bounding box is not provided, we set positive samples with detection scores above a threshold (in our implementation, it is set as -0.5, which both considers the reliability of detection and the number of positives). A stochastic gradient descent algorithm is used at a learning rate one tenth of the initial pre-training rate. The network is trained for 60K iterations, which takes about 12 hours (Intel Core i7 with GPU 3.2GHz).

C. Experimental 1: How Many Subcategories for FOAF?

FOAF regards several subcategories as one macro-class, and the remaining problem for FOAF is the choice of subcategory

TABLE I

CLASSIFICATION ACCURACY OF FOAF WITH DIFFERENT MAXIMAL ALLOWABLE NUMBER WITHIN A FUSED CLASS

No.	5	10	15	20	25	All
Acc.	79.6%	79.8%	**80.40%**	80.02%	79.66%	79.38%

TABLE II

SPECIES RECOGNITION PERFORMANCE ON CUB-200-2011. WE LIST A DETAILED COMPARISON OF OUR METHOD WHEN DIFFERENT SUB-MODULES ARE ADDED IN, AND REPORT RESULTS BY STEP IN ORDER TO RECORD HOW DIFFERENT SUB-MODULES CONTRIBUTE TO THE FINAL RESULTS. "FT" REFERS TO FINE TUNE

Method	Train Anno.	Test Anno.	Accuracy
Head Detection (Sec. IV-A)	bbox	bbox	72.92%
	n/a	n/a	68.38%
Alignment (Sec. IV-C)	bbox	bbox	75.31%
	n/a	n/a	69.61%
FOAF (Sec. V)	bbox	bbox	77.27%
	n/a	n/a	72.40%
FOAF-ft	bbox	bbox	**83.08%**
	n/a	n/a	**75.98%**

TABLE III

SPECIES RECOGNITION PERFORMANCE ON CUB-200-2011. "BBOX" AND "PARTS" REFER TO USING OBJECT BOUNDING BOX AND PART ANNOTATIONS. "ALEX", "VGG", AND "GOOGLE" REFER TO DIFFERENT CNN MODELS

Method	Train Anno.	Test Anno.	Accuracy
Ours (Alex)	bbox	bbox	**83.08%**
	n/a	n/a	**75.98%**
Ours (VGG)	bbox	bbox	**86.34%**
	n/a	n/a	**84.63%**
Xie *et al.* [22]	bbox+parts	bbox+parts	66.35%
Xie *et al.* [22] + FOAF	bbox+parts	bbox+parts	69.10%
Chai *et al.* [19]	bbox	bbox	59.4%
Chai *et al.* [19] + FOAF	bbox	bbox	62.13%
Berg *et al.* [21]	bbox	bbox	56.78%
Gavves *et al.* [26]	bbox	bbox	67%
	n/a	n/a	53.6%
Branson *et al.* [41] (Alex)	bbox+parts	n/a	75.7%
	bbox+parts	bbox+parts	85.4%
Zhang *et al.* [42] (Alex)	bbox+parts	n/a	73.89%
	bbox+parts	bbox	76.37%
Simon *et al.* [33] (Alex)	n/a	n/a	68.50%
Simon *et al.* [33] (VGG)	n/a	n/a	81.01%
Krause *et al.* [40] (Alex)	bbox	bbox	74.9%
Krause *et al.* [40] (VGG)	bbox	bbox	82.8%
Xiao *et al.* [31] (Alex)	n/a	n/a	69.7%
Jad. *et al.* [34] (Google)	n/a	n/a	84.1%

number in the fused class. The Neighboring Joining method iteratively fuses similar subcategories into bigger classes, and the number of subcategories grows as we moves towards the root of the tree (as an extreme, root node contains all the subcategories). However, too many subcategories within a fused class leads to disturbance for classification.

In this experiment, we evaluate how many subcategories are suitable for FOAF. In order to make a fair comparison, and to test the maximum recognition capacity of parts for such a task, the experiment is conducted on CUB-200-2011 with ground truth object bounding boxes and part annotations available. Table I shows the classification results of FOAF with different maximal allowable number in a fused class. The maximal classification accuracy reaches 80.40% with a fused class restriction at 15 (dimension around 180 per part). However, as the number grows, more classes are included in one fused class and the performance decreases. The number marked as "All" includes all the fused classes for FOAF learning, which is 199 dimension per part. For simplicity, we choose the value of k as 15 for the following experiments.

D. Experimental 2: Fine-Grained Categorization Results

CUB-200-2011 CUB-200-2011 is the most widely used fine grained dataset, which contains 11, 788 images spanning 200 sub-species. Each image is labeled with its species, a bounding box for the bird, and the key points of fifteen parts (which we do not use). Table II shows the experimental results of our method on CUB-200-2011 when different sub-modules are added in. The first setting is semi-automatic, where the object bounding box is provided, as most previous methods assume. We extract features from the detected head (Sec. IV-A) as well as the ground truth object bounding box, which brings an recognition accuracy of 72.92%. Geometric alignment (Sec. IV-C) introduces extra parts and boosts the performance to 75.31%. Finally, FOAF (Sec. V) brings about another improvement around 2%, with an accuracy of 77.27%, Fine tune improves this result by a large margin, to over 83%.

The second setting is fully automatic where the object bounding box is unknown at both training and testing time. We detect the head as well as the object, followed by a geometric update, the corresponding accuracy is 68.38%. Geometric alignment and FOAF improve the result to 69.61% and 72.40%, respectively. We achieve a final accuracy of 75.98% after network fine tune, which is an encouraging result considering the difficulty of this task.

There are many previous works reporting results on CUB-200-2011, Table III shows the comparison results of our method with some other related works. According to the part localization techniques, these works can be categorized into three types of methods. The first type tries to train supervised detectors for each part [41], [42], which is similar with our method. However, we do not localize highly deformable parts by such template-based models. Our method could achieve an accuracy of 83.08% in semi-automatic setting, an 28% relative error reduction comparing with the highest performing method 76.37%. Furthermore, the fully automatic classification result is 75.98%, even comparable with the best result (75.7%) when object and part annotations are available at training time. The second method considers the selectiveness of CNN filter banks, and tries to find part detectors automatically by grouping filters [31], [33]. However, such detectors are not discriminative enough and the returned detections are cluttered, the highest accuracy is 69.7% among this kind of method, which is much lower than our method (75.98%). The third method aligns parts based on segmentation [26], [40], with corresponding highest accuracy 74.9%.

Our proposed FOAF is independent of the features. To verify this, we learn FOAF based on the low-level

TABLE IV
SPECIES RECOGNITION PERFORMANCE ON STANFORD DOGS

Method	Train Anno.	Test Anno.	Accuracy
Head Detection (Sec. IV-A)	bbox	bbox	68.62%
	n/a	n/a	64.92%
Alignment (Sec. IV-C)	bbox	bbox	69.95%
	n/a	n/a	65.38%
FOAF (Sec. V)	bbox	bbox	71.59%
	n/a	n/a	67.73%
FOAF-ft	bbox	bbox	**74.49%**
	n/a	n/a	**68.66%**
Yang *et al.* [17]	bbox	bbox	38%
Chai *et al.* [19]	bbox	bbox	45.6%
Chai *et al.* [19] + FOAF	bbox	bbox	49.23%
Gavves *et al.* [26]	bbox	bbox	57%
	n/a	n/a	49%
Simon *et al.* [33]	n/a	n/a	68.61%

features of [19] and [22], and achieve accuracies of 69.10% and 62.13%, respectively, a noticeable improvement over the low-level features (66.35% and 59.4%, respectively), which demonstrates the effectiveness and transportability of our proposed FOAF. Recently, the performance is boosted again by switching to more powerful CNN models. To make fair comparisons, we also report results based on a more deep CNN structure (VGGNet). The accuracies are 86.34% and 84.63% in case of semi-automatic and fully automatic settings, respectively, which is higher than the highest performing methods [40] (82.8%) and [34] (84.1%) under the same level of annotations.

Stanford Dogs: This dataset consists of $20,580$ images with 120 dog species. The default train/test split gives us around 100 training images and 70 test images per class. Since Stanford Dogs dataset is extracted from ImageNet, simply choosing the pre-trained network brings about cross-dataset redundancy. Considering this issue, we check the ILSVRC 2012 training data and remove samples that are used as test in Stanford Dogs dataset, and train a network from scratch to get the model specific to Stanford Dogs. Our pretrained network nearly matches the performance of [18], with a validation accuracy of 55.78%.

Table IV shows the classification results of our method and some related works. The overall framework achieves accuracies of 74.49% and 68.66% in semi-automatic and fully automatic settings, respectively, which is comparable with the result in [33]. The performance improvement is less than that on CUB-200-2011, mainly due to a greater pose variability. At the same time, most dogs have a nice roundish shape and some parts are occluded, which makes alignment difficult. Noticeably, FOAF boosts the performance of [19] with nearly 4% over the low-level features, which demonstrates the effectiveness of FOAF.

Oxford IIIT Pets: Oxford-IIIT Pets dataset is a collection of $7,349$ images of cats and dogs of 37 different breeds, of which 25 are dogs and 12 are cats. All images have an associated ground truth annotation of breed, head ROI, and pixel level trimap segmentation. Here, we do not use the head ROI and segmentation information. Table V shows the results on this dataset. We achieve an accuracy of 91.39%, which is higher than the best result [48] (88%).

TABLE V
SPECIES RECOGNITION PERFORMANCE ON OXFORD IIIT PETS

Method	Accuracy
Head Detection (Sec. IV-A)	85.31%
Alignment (Sec. IV-C)	86.92%
FOAF (Sec. V)	**89.70%**
FOAF-ft	**91.39%**
Bo *et al.* [46]	53.4%
Angelova *et al.* [45]	54.3%
Murray *et al.* [47]	56.8%
Azizpour *et al.* [48]	88.1%
Simon *et al.* [33] (Alex)	85.20%

TABLE VI
SPECIES RECOGNITION PERFORMANCE ON COLUMBIA DOGS

Method	Accuracy
Head Detection (Sec. IV-A)	79.31%
Alignment (Sec. IV-C)	80.03%
FOAF (Sec. V)	**83.64%**
FOAF-ft	**85.87%**
Liu *et al.* [32]	67%

Columbia Dogs: This dataset contains $8,351$ real-world images of 133 American Kennel Club (AKC) recognized dog breeds. The amount of data is smaller than Stanford Dogs, but with more sub-categories. Each image is also provided with head part information, such as eyes, nose, and ears, *etc.*. However, object bounding box annotations are not provided. Similarly with the setting in Oxford-IIIT Pets, we do not use any part level annotations. The results for the dataset is shown in Table VI. Few works report accuracies on this dataset. We achieve an accuracy of 87.72%, comparing with the method in [32], which means an absolute improvement of more than 20%.

E. Complexity Analysis

In this section, we discuss the computational complexity of the scheme. For candidate region proposal generations, we choose selective search's "fast" mode to produce around 2200 region proposals in average for each image. Each proposal is forward propagated through the CNN to get fc_7 features. In current implementation, it takes about 9s (AlexNet) in average for object and head detection. The time cost is mainly attributed the feature computation. However, the cost can be decreased to about $1/40$ using recent proposed spatial pyramid pooling CNN [44], with comparable results. The number of detection candidates in Eq. (1) is set to 100 for both object and head, taking both performance and complexity into consideration. In Eq. (3), the top 1000 pairs are used for object confidence map generation. Both geometric constraint and object confidence map generation are fast, with dozens of images per second.

For features learning, Neighboring Joining method is fast, which takes less than $5s$ to construct the "tree of similarity". FOAF learning takes about $70s$ in average per part, and around $560s$ in total. Classification based on FOAF is fast due to its low-dimensional (around $1.5k$) property, with about $68s$. On the other hand, classification directly based on the extracted

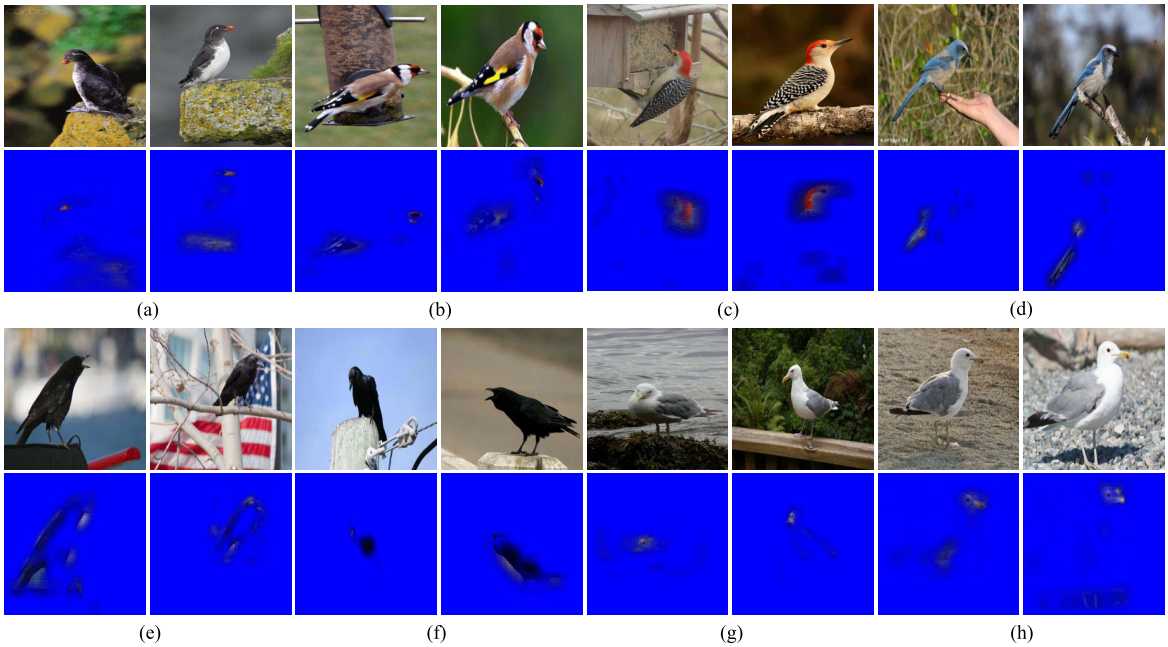


Fig. 10. Example fully automatic classification results on CUB-200 − 2011. We show some well (top two rows) and poorly (bottom two rows) recognized subcategories, together with the saliency maps explaining why these birds are recognized as certain subcategories. (a) Parakeet Auklet (100%). (b) European Goldfinch (100%). (c) Red Bellied Woodpecker (96.7%). (d) Florida Jay (96.7%). (e) American Crow (23.3%). (f) Fish Crow (30.0%). (g) Glaucous Winged Gull (31.0%). (h) California Gull (36.7%).

low-level features (with dimension 9216 per part and 36864 in total) costs about $297s$. It can be seen that FOAF increases the total time cost about twice as much time as classification directly based on the low-level features.

VII. Discussion

A. What Makes a Red Bellied Woodpecker Look Like a Red Bellied Woodpecker?

What makes a *Red Bellied Woodpecker* look like a *Red Bellied Woodpecker*? To answer this question, we need to investigate the classification process and uncover why success and failure cases happen. As an illustration, Fig. 10 shows some 'easiest' (accuracy above 95%) and 'hardest' (accuracy below 40%) classes of our fully automatic classification results on CUB-200-2011. In order to find why these birds are recognized as certain subcategories, we find distinctive details which contribute most to the final classification score. Given a subcategory c and its corresponding classification model w_c, we only reserve the top d ($d = 80$) dimensional positive weight values w_c^d, and identify patches which give the strongest activations for classifier w_c^d. The CNN features are extracted from sliding window patches at four scales with window size $w \in \{16, 24, 32, 40\}$ and stride 8, which is similar as dense SIFT feature extraction. The top activation pathes are averaged to obtain the saliency maps, as shown in Fig. 10.

We make several observations from the saliency maps. It appears that successfully recognized subcategories activate on consistently parts. For *European Goldfinch*, the distinctive parts are red forehead and yellow and black wings. For *Florida Jay*, it has sapphire long tail. An interesting case is *Parakeet Auklet*, besides red beak, the most distinctive part is mossy rocks, the findings are reasonable since *Parakeet Auklet* usually inhabits on mossy rocks, which is rare for other subcategories. Now we can answer that a *Red Bellied Woodpecker* is best recognized by its red patches around crowns.

However, the least successful classes are due to the existence of confusing counterparts, such as *Fish Crow* and *American Crow*, *California Gull* and *Herring Gull*. It is hard to tell them apart merely from appearance. In fact, through the description of Wiki, the main difference between *Fish Crow* and *American Crow* is their voice, the call of *Fish Crow* has been described as a nasal "ark-ark-ark", while *American Crow* is a distinct "caw caw". This suggests that recognizing these confusing subcategories needs human intervention such as questions posed to the user.

B. Does Part Localization Really Count?

A large number of preceding works have declared that their part-based models are effective for fine-grained categorization. However, they did not test these representations on a common ground. As noted by Chatfield *et al.* [14] in their comparison of visual encoding, the performance of computer vision systems depends significantly on implementation details. In this section, we compare different part localization methods under the same baseline on Birds dataset, which has extensive part annotations available apart from object bounding boxes. Given these extra annotations, we evaluate what would be achieved under different part-based models and when moving away from coarse alignments to accurate alignments.

For fair comparison, we extract features part by part with "AlexNet" model, and concatenating all the part features as well as object features after normalization. The "ground truth"

TABLE VII

CLASSIFICATION ACCURACY COMPARISONS WITH DIFFERENT PART LOCALIZATION METHODS FROM COARSE TO FINE ORDER, ALL THE RESULTS ARE BASED ON THE SAME BASELINE FEATURES FOR FAIR COMPARISONS. WE RE-IMPLEMENT [42] TO OBTAIN HEAD AND OBJECT LOCALIZATION RESULTS, WHILE THE PART LOCALIZATION RESULTS OF [19] AND [26] ARE PROVIDED BY THE CORRESPONDING AUTHORS. "GT" REFERS TO GROUND TRUTH

Method	SPM [8]	Alignments [26]	Symbolic [19]	Part RCNN [42]	ours	gt	gt subset
Accuracy	63.23%	65.30%	68.31%	73.91%	75.31%	78.56%	80.61%

method using the ground truth annotations as part regions. There are 15 part locations annotated per image which include beak, eyes, feet, *etc.*, since there are few images with all fifteen parts visible. In particular, most birds have only one eye and one wing visible. In order to enable better correspondences between parts, we combine the features of left and right eyes (if visible, and the same with wings and legs) via max pooling. Thus we have 12 parts in total for "ground truth" method.

The classification accuracies are listed in Table VII. The results show that the performance could be improved after introducing better alignment methods [19], [26], [42], and our alignment method outperforms the others noticeably. Specifically, [42] also makes use of CNN features for part localization, which localizes all parts with template-based model, without considering the deformation degree of parts. In contrast, we only detect the stable parts with template-based model and localize other highly deformable parts with simple geometric alignment. The classification accuracy of our alignment method is 75.31%, which is higher than the method in [42] (73.91%). Intuitively, the performance could be further improved if the ground truth part annotations are available.

Moreover, we order the classification accuracies by part and disregard the parts which are the least discriminative (back, legs, and tails), and the accuracy reaches 80.61% (denoted as ground truth subset), which is better than using all the parts. The results are intuitive since the least discriminative parts such as tails and legs are mostly overwhelmed by the background and sometimes invisible. Thus we conclude that part localization counts for fine-grained visual categorization only when these parts are discriminative enough themselves.

VIII. CONCLUSION

In this paper, we propose a novel method for fine-grained visual categorization. First, we combine semantic prior with geometric information for part alignment. The stable, less deformable regions are firstly detected with template-based method, and geometric alignment is performed to localize highly deformable parts. Secondly, we learn one-vs-all features, which are dimension friendly and entry meaningful. The dimension of one-vs-all features scales with the number of classes and is far less than that of the low-level ones. Each entry of the learned features represents the signed distance from current part to the target one. Furthermore, we fuse similar subcategories and learn Fused One-vs-All Features for classification. Combining all the techniques we achieve superior performance on several fine grained animal dataset for semi-automatic and fully automatic classification.

REFERENCES

[1] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. Eur. Conf. Comput. Vis. Workshop*, 2004, pp. 1–16.

[2] Y. Lu, L. Zhang, J. Liu, and Q. Tian, "Constructing concept lexica with small semantic gaps," *IEEE Trans. Multimedia*, vol. 12, no. 4, pp. 288–299, Jun. 2010.

[3] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4269–4279, Sep. 2012.

[4] W. Zhou, Q. Tian, Y. Lu, L. Yang, and H. Li, "Latent visual context learning for Web image applications," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2263–2273, 2011.

[5] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2559–2566.

[6] X. Zhang, H. Xiong, W. Zhou, and Q. Tian, "Fused one-vs-all mid-level features for fine-grained visual categorization," in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 287–296.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[8] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2169–2178.

[9] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, vol. 22, no. 1, pp. 67–92, Jan. 1973.

[10] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[11] L. Xie, Q. Tian, M. Wang, and B. Zhang, "Spatial pooling of heterogeneous features for image classification," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1994–2008, May 2014.

[12] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 951–958.

[13] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2011.

[14] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods," in *Proc. Brit. Mach. Vis. Comput.*, 2011, pp. 76.1–76.12.

[15] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[16] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3498–3505.

[17] S. Yang, L. Bo, J. Wang, and L. G. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 3131–3139.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[19] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 321–328.

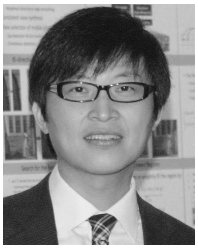
[20] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 89–96.

[21] T. Berg and P. N. Belhumeur, "POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 955–962.

[22] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, "Hierarchical part matching for fine-grained visual categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1641–1648.
[23] H. Azizpour and I. Laptev, "Object detection using strongly-supervised deformable part models," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 836–849.
[24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[25] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek, "Visual word ambiguity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1271–1283, Jul. 2010.
[26] E. Gavves, B. Fernando, C. G. Snoek, A. W. M. Smeulders, and T. Tuytelaars, "Local alignments for fine-grained categorization," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 191–212, Jan. 2015.
[27] S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 623–634, Feb. 2014.
[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[29] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. IEEE Comput. Vis. Pattern Recognit. Workshop*, Jun. 2011, pp. 1–2.
[30] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.
[31] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 842–850.
[32] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 172–185.
[33] M. Simon and E. Rodner. (2015). "Neural activation constellations: Unsupervised part model discovery with convolutional networks." [Online]. Available: http://arxiv.org/abs/1504.08289
[34] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. (2015). "Spatial transformer networks." [Online]. Available: http://arxiv.org/abs/1506.02025
[35] P. Welinder *et al.*, "Caltech-UCSD birds 200," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2010-001, 2010.
[36] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.
[37] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 729–736.
[38] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.
[39] C. Goering, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric part transfer for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2489–2496.
[40] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5546–5555.
[41] S. Branson, G. Van Horn, S. Belongie, and P. Perona. (Jun. 2014). "Bird species categorization using pose normalized deep convolutional nets." [Online]. Available: http://arxiv.org/abs/1406.2952
[42] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.
[43] N. Saitou and M. Nei, "The neighbor-joining method: A new method for reconstructing phylogenetic trees," *Molecular Biol. Evol.*, vol. 4, no. 4, pp. 406–425, 1987.
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 346–361.
[45] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 811–818.
[46] L. Bo, X. Ren, and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 660–667.
[47] N. Murray and F. Perronnin, "Generalized max pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2473–2480.
[48] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. (2014). "From generic to specific deep representations for visual recognition." [Online]. Available: http://arxiv.org/abs/1406.5774

Xiaopeng Zhang received the B.S. degree in electronics engineering from Sichuan University, Sichuan, China, in 2011. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University.

His current research interests include object recognition, detection, and multimedia signal processing.



Hongkai Xiong (M'01–SM'10) received the Ph.D. degree in communication and information system from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2003. Since then, he has been with the Department of Electronic Engineering, SJTU, where he is currently a Distinguished Professor. From 2007 to 2008, he was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA, as a Research Scholar. From 2011 to 2012, he was a Scientist with the Division of Biomedical Informatics, University of California at San Diego, San Diego, CA, USA.

He has published over 140 refereed journal/conference papers. His research interests include source coding/network information theory, signal processing, computer vision, and machine learning. He was a recipient of the Best Student Paper Award at the 2014 IEEE Visual Communication and Image Processing, the best paper award at the 2013 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, and the Top 10% Paper Award at the 2011 IEEE International Workshop on Multimedia Signal Processing.

Since 2012, he has been a member of Innovative Research Groups of the National Natural Science. In 2014, he was granted the National Science Fund for Distinguished Young Scholar and Shanghai Youth Science and Technology Talent as well. In 2013, he was awarded a recipient of Shanghai Shu Guang Scholar. In 2011, he received the First Prize of the Shanghai Technological Innovation Award for Network-Oriented Video Processing and Dissemination: Theory and Technology. In 2010 and 2013, he also received the SMC-A Excellent Young Faculty Award of Shanghai Jiao Tong University. In 2009, he was awarded a recipient of New Century Excellent Talents in University, Ministry of Education of China. He served as a TPC Member for prestigious conferences, such as ACM Multimedia, ICIP, ICME, and ISCAS.

Wengang Zhou received the B.S. degree in electronic information engineering from Wuhan University, China, in 2006, and the Ph.D. degree in electronic engineering and information science from the University of Science and Technology of China (USTC), Hefei, China, in 2011. He was a Research Intern with the Internet Media Group, Microsoft Research Asia, from 2008 to 2009. From 2011 to 2013, he was a Post-Doctoral Researcher with the Computer Science Department, University of Texas at San Antonio. He is currently an Associate Professor with the Department of Electronic Engineering and Information Science, USTC. His research interests include computer vision and multimedia content analysis and retrieval.

Qi Tian (M'96–SM'03–F'16) was a Tenured Associate Professor from 2008 to 2012 and a Tenure-Track Assistant Professor from 2002 to 2008. From 2008 to 2009, he took one-year faculty leave with Microsoft Research Asia as a Lead Researcher in the Media Computing Group. He was a Visiting Scholar with the MIAS Center, University of Illinois at Urbana–Champaign (UIUC), in 2007, and a Visiting Professor with NEC Laboratories of America in 2003. He is currently a Full Professor with the Department of Computer Science, University of Texas at San Antonio (UTSA).

He received the B.E. degree in electronics engineering from Tsinghua University, in 1992, the M.S. degree in ECE from Drexel University, in 1996, and the Ph.D. degree in ECE from UIUC, in 2002. His research interests include multimedia information retrieval, computer vision, pattern recognition, and bioinformatics, and published over 310 refereed journal and conference papers. He was the co-author of a Best Paper in ACM International Conference on Multimedia Retrieval (ICMR) 2015, a Best Paper in PCM 2013, a Best Paper in MMM 2013, a Best Paper in ACM ICIMCS 2012, a Top 10% Paper Award in MMSP 2011, and a Best Student Paper in ICASSP 2006, and co-author of a Best Student Paper Candidate in ICME 2015, and a Best Paper Candidate in PCM 2007.

Dr. Tian served as a Founding Member of the International Steering Committee for ACM ICMR (2009–2014), the ACM Multimedia Conference Review Committee Member (2009-), and the International Steering Committee Member for ACM MIR (2006–2010), the Best Paper Committee Member for ACM Multimedia 2009, ACM ICIMCS 2013, ICME 2006 and 2009, PCM 2012, and the IEEE International Symposium on Multimedia 2011. He will/has served as the General Chair for ACM Multimedia 2015, the Program Coordinator for ACM Multimedia 2009, and the Program Chair for various international conferences, including ACM CIVR 2010, ACM ICMCS 2009, MMM 2010, IMAI 2007, VIP 2007 and 2008, and MIR 2005. He has also served in various organization committees as Panel and Tutorial Chair, the Publicity Chair of the Special Session Chair, the Track Chair in numerous ACM and IEEE conferences, such as ACM Multimedia, VCIP, PCM, CIVR, and ICME, and served as a TPC Member for prestigious conferences, such as ACM Multimedia, SIGIR, ICCV, and KDD.

He was a member of ACM (2004). He received the 2014 Research Achievement Awards from the College of Science, UTSA. He received the 2010 ACM Service Award. He is the Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the *Multimedia System Journal*, and on the Editorial Board of the *Journal of Multimedia* and the *Journal of Machine Vision and Applications*. He is the Guest Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and the *Journal of Computer Vision and Image Understanding*. His research projects are funded by ARO, NSF, DHS, Google, FXPAL, NEC, SALSI, CIAS, Akiira Media Systems, HP, and UTSA.