Picking Neural Activations for Fine-Grained Recognition

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

***Abstract*—It is a challenging task to recognize fine-grained subcategories due to the highly localized and subtle differences among them. Different from most previous methods that rely on object/part annotations, this paper proposes an automatic fine-grained recognition approach, which is *free of any object/part annotation at both training and testing stages*. The key idea includes two steps of picking neural activations computed from the convolutional neural networks, one for localization, and the other for description. The first picking step is to find distinctive neurons that are sensitive to specific patterns significantly and consistently. Based on these picked neurons, we initialize positive samples and formulate the localization as a regularized multiple instance learning task, which aims at refining the detectors via iteratively alternating between new positive sample mining and part model retraining. The second picking step is to pool deep neural activations via a spatially weighted combination of Fisher Vectors coding. We conditionally select activations to encode them into the final representation, which considers the importance of each activation. Integrating the above techniques produces a powerful framework, and experiments conducted on several extensive fine-grained benchmarks demonstrate the superiority of our proposed algorithm over the existing methods.**

***Index Terms*—Fine-grained recognition, regularized multiple instance leaning, spatially weighted Fisher Vectors (SWFV), weakly supervised part discovery.**

I. Introduction

FINE-GRAINED recognition aims at discriminating usually hundreds of subcategories belonging to the same basic-level category. Typical applications include discriminating different kinds of birds, dogs, and cars etc. It lies between the basic-level category classification (e.g., categorizing bikes, boats, and cars etc. in Pascal VOC [1]) and the identification of individual instances (e.g., face recognition). An inexperienced person can easily recognize basic-level categories such as bikes or horses immediately since there exist a large amount of cues for discriminating them, while it is difficult for him/her to tell a blackbird from a cowbird (c.f. Fig. 1) without specific guidance. As a matter of fact, fine-grained sub-categories often share the same parts (e.g., all birds should have wings, legs, etc.), and are often discriminated by the subtle differences in texture and color properties of these parts (e.g., only the breast color counts when discriminating some similar birds). Hence, localizing and describing object and the corresponding parts become crucial for fine-grained recognition.

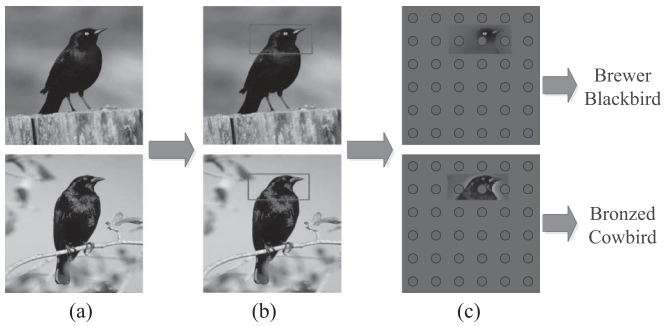


Fig. 1. How do we tell a blackbird from a cowbird? This paper proposes a weakly supervised fine-grained recognition method, which demands only the labels of training images. Given a test image, we first localize the discriminative parts with automatically discovered detectors (only one part detector is shown). For each detected part, a new kind of feature named SWFV-CNN is extracted for representation. (a) Test Image, (b) Part Detection, (c) SWFV-CNN.

In order to localize the discriminative parts, most existing methods explicitly require object or even part annotations at both training and testing stages [2]–[5]. However, such a requirement is demanding in practical applications. As a compromise, some methods consider object/part annotations at only training stage but not at testing time [6], [7]. However, it is still time consuming to acquire these annotations, especially for large scale recognition problems. Hence, one promising research direction is to free us from the tedious and subjective manual annotations for fine-grained recognition, which demands automatic part discovery. However, automatic part discovery is a classical chicken-and-egg problem: without an accurate appearance model, examples

Manuscript received August 19, 2016; revised March 5, 2017; accepted May 28, 2017. Date of publication June 1, 2017; date of current version November 15, 2017. This work was supported in part by the National Science Foundation of China under Grant 61425011, Grant 61622112, Grant 61529101, Grant 61472234, Grant 61471235, and Grant 61632019, and in part by the China Scholarship Council under Grant 201506230029. The work of H. Xiong was supported by the Program of Shanghai Academic Research Leader under Grant 17XD1401900, the work of W. Lin was supported by the Microsoft Research Aisa Collaborative Research Award, and the work of Q. Tian was supported in part by the ARO 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. Martha Larson. *(Corresponding author: Qi Tian.)*

X. Zhang, H. Xiong, and W. Lin are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zxphistory@sjtu.edu.cn; xionghongkai@sjtu.edu.cn; wylin@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/TMM.2017.2710803

1520-9210 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

of a part cannot be discovered, while an accurate appearance model cannot be learned without part examples.

Feature representation is another key issue for fine-grained recognition. Recently, Convolutional Neural Network (CNN) has been widely used for feature representation. However, there exist two challenges when directly applying CNN to fine-grained tasks. First, traditional CNN requires fixed size of rectangle as input, which inevitably includes background information. However, as demonstrated in [2], [8], background is unlikely to play any major role for fine-grained recognition since all sub-categories share similar backgrounds (e.g., all birds are usually found in trees or fly in the sky). Second, traditional CNN captures the spatial layout of an image, which may be useful for representing the shape of an object, while it may be not useful for describing fine-grained texture details.

Based on these observations, this paper presents a weakly supervised fine-grained recognition method, which *is free of any object/part annotation at both training and testing stages*. Suppose we are faced with one challenging task, say, telling a blackbird from a cowbird (c.f. Fig. 1). As our *first contribution*, an automatic part detection strategy is proposed to localize the discriminative parts (Section III). The detection method consists of two main points. First, an automatic initialization method is developed for detector learning, which is based on the selectivity of CNN neurons. The key insight of the initialization approach is to elaborately pick deep neurons with significant and consistent responses. Second, a set of detectors are learned via a regularized multiple instance learning strategy. We introduce a regularized term to consider the reliability of each positive sample. The learned detectors tend to discover discriminative and consistent patches that are helpful for part-based recognition.

There exist previous methods that make use of neuron activations for part discovery [9], [10]. The method of [9] directly groups all the intermediate neurons of a CNN via spectral clustering, and performs part detection via the grouped neurons. However, the intermediate neurons are implicitly trained and most of them are not discriminative. [10] initializes part locations via the neural activation maps, and models spatial constellations via deformable part models, while it is very hard to model highly deformable objects such as birds and dogs. Hence the detection performance is limited. Based on the observations, we propose a picking strategy to only select the distinctive neurons (Section III-A). Furthermore, since these neurons are implicitly trained from the network, which are weak in generalization. We propose to enhance these weak detectors (neurons) via a regularized MIL approach. Experimental results have demonstrated the advantages of our proposed method over the related methods [9], [10].

As the *second contribution*, we design a new kind of feature that is suitable for fine-grained representation (Section IV). The deep neural activations of a CNN are regarded as local descriptors, and encoded via Spatially Weighted Fisher Vector (SWFV-CNN). The key insight is that not all neural activations are equally important for recognition, and the goal is to highlight the activations that are crucial for recognition and discount those that are less helpful. To this end, a picking strategy is proposed to conditionally select descriptors based on a saliency map, which indicates how likely a pixel belongs to the salient regions. As shown in Fig. 1(c), irrelevant backgrounds are masked out and only the salient part around eyes are highlighted, which represent the most distinctive details to tell the two birds apart.

Previous approaches also consider fisher vectors over deep features [11], [12]. The differences are: first, previous approaches often treat deep features as local descriptors and encode them into a global image representation, while we record the mapping relationships between the original images and convolutional descriptors, and encode them into Fisher Vectors by part. Second and most importantly, different from previous approaches [11], [12], which treat each descriptor equally important, we propose a new kind of features named SWFV-CNN, which highlights the descriptors that are crucial for recognition and discounts those that are less helpful via a weighted combination of Fisher Vectors. Experimental results demonstrate that SWFV-CNN performs consistently better than FV-CNN for fine-grained recognition, and is complementary with traditional CNN to further boost the performance.

Framework overview: An overview of the proposed framework is shown in Fig. 2. Our approach consists of two picking steps. The first step aims at picking deep neurons that respond to specific patterns significantly and consistently. Based on these neurons, we select positive samples that are semantically similar and train a set of discriminative detectors. An iterative multiple instance learning procedure is executed, which alternates between selecting positive samples and training classifier, while applying cross-validation at each step to prevent classifier from overfitting the initial positive samples. The trained detectors are used to discover parts for recognition. The second step is to pick CNN activations via Spatially Weighted combination of Fisher Vectors, which we refer to SWFV-CNN. We compute spatial weights with part saliency maps, which indicates how likely a pixel belongs to a salient part. The part saliency map is used to weight each Fisher Vector and pool it into the final representation, which considers the importance of each descriptor.

This is an extension of our earlier work [13]. In this paper, we extend [13] in a number of ways. First, we propose a systematic picking strategy for object-level and part-level initialization, respectively. Second, we formulate the weakly supervised detector learning as a regularized multiple instance leaning issue, which aims at learning more generalized detectors. Third, for the theory part, we provide rationales for regularized multiple instance detector learning and SWFV-CNN based fine-grained description. The last but not least, more extensive experiments are presented, including new results on cars-196 [14] and aircrafts [15], ablation studies on the recognition performance versus the number of detectors, and the performance analysis step by step, which demonstrate the necessity of each module.

The rest of this paper is organized as follows. Section II describes related works on fine-grained recognition. The details of our proposed part discovery strategy are elaborated in Section III. In Section IV, we describe the Spatially Weighted FV-CNN. Experimental results and discussions are given in Sections V and VI, respectively. Finally, Section VII concludes the paper.

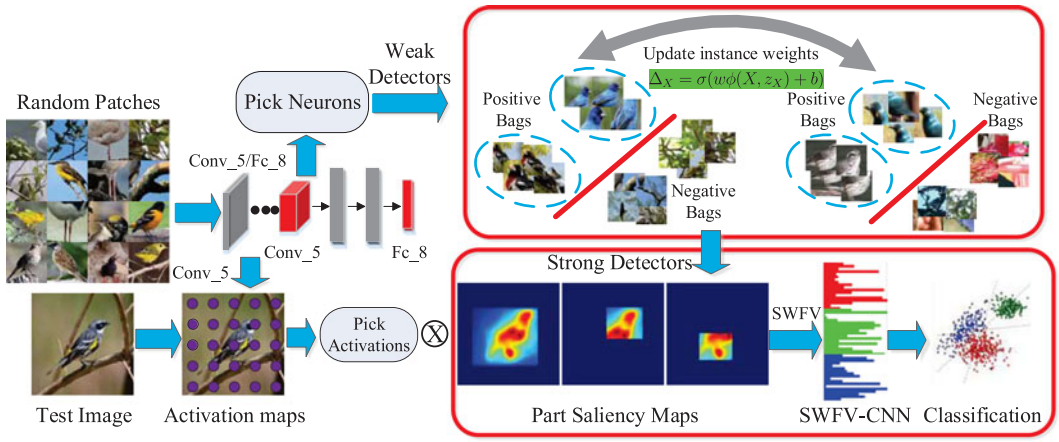


Fig. 2. Overview of our developed framework. The approach consists of two picking steps. The first step aims at picking deep neurons that are sensitive to specific patterns significantly and consistently. Based on these picked neurons, we choose positive samples and train a set of discriminative detectors via a regularized multiple instance leaning. The second step is to pick neural activations via spatially weighted Fisher Vector (SWFV).

II. Related Works

Fine-grained recognition is a challenging problem and has recently emerged as a hot topic, typical applications include discriminating different kinds of birds [16], dogs [17], cars [14], and handbags [18] etc. In the following, we organize our discussion related to fine-grained recognition within two tasks: part localization and feature representation.

A. Part Localization

As fine-grained datasets are often provided with extra annotations of bounding boxes and part landmarks [16], [19], most previous methods rely on these annotations more or less.

Early methods assume that annotations are available at both training and testing time. Among them the strongest supervised setting is to use both object and part-level annotations [3], [20]. Obviously, this kind of setting is demanding and a more reasonable setting only assumes the availability of object bounding boxes. Chai *et al.* [2] introduce techniques that improve both segmentation and part localization accuracy by simultaneous segmentation and detection. Bergs *et al.* [20] propose to automatically detect part locations with a consensus of exemplars, and Goering *et al.* [21] transfer part annotations from objects via performing a simple but very effective global matching and a subsequent ensemble learning. Huang *et al.* [22] propose to learn a fully convolutional network to locate multiple object parts based on the strong part annotations, and encode object-level and part-level cues via a two-stream classification network. Zhang *et al.* [4] propose to train two sub-networks, one for localization and the other for classification.

Later methods require annotations only during training, and no knowledge of annotations at testing time. These methods are supervised at the level of object and parts during training. Zhang *et al.* [7] generalize R-CNN [23] framework to detect parts as well as the whole object. Branson *et al.* [24] train a strongly supervised model in a pose normalized space. Further on, Krause *et al.* [6] propose a co-segmentation method, which only needs object-level annotations at training time, and is completely unsupervised at the level of parts.

Recently, there have been some methods that aim at a more general condition, e.g., without expecting any information about the location of fine-grained objects, neither during training nor testing time. This level of unsupervision is a big advance towards making fine-grained recognition suitable for wide deployment. Xiao *et al.* [9] propose to use two attention models with deep convolutional networks, one to select patches to a foreground object, and the other to localize discriminative parts. Simon *et.al.* [10] propose to localize parts with a constellation model, which incorporates CNN into the deformable part model [25]. Zhang *et al.* [26] generate multiple scale proposals, and select useful ones based on the importance for classification. Lin *et al.* [11] make use of bilinear models for network training, which outputs a bilinear vector that are the outer product of two sub-networks. Jaderberg *et al.* [27] train an end-to-end spatial transform network to discover and learn part detectors in a data-driven manner without any additional supervision.

Our approach belongs to the last setting, which is free of any object/part-level annotation at both training and testing stages. Different from previous approaches, we explicitly learn a set of discriminative detectors via a regularized multiple instance learning strategy [28]. Our part localization approach belongs to a family of weakly supervised detector learning, which have been widely studied for part discovery [29], [30]. Different from these methods that often employ some heuristic methods such as k-means for initialization, we propose a picking strategy to select patches that are consistent in appearance.

Our method is also related with multiple instance learning (MIL), which is a particular form of weak supervision. In MIL, labels are assigned to bags (sets of patterns), instead of individual patterns. The positive bags are sets of instances containing at least one positive example, while the negative bags are sets of instances that are all negative. MIL was originally introduced to solve a problem in biochemistry [31], and a variety of MIL algorithms have been developed over the past years. The simplest method is to transform MIL into a standard supervised learning problem by applying the bag's label to all instances in the bag [32]. However, such method assumes that the positive examples are rich in the positive bags. Andrews *et al.* [28] present a new

formulation of MIL as a max-margin SVM problem. Different from previous methods, we introduce a regularized term into MIL, which consider the positiveness of each positive bag, and obtain improved localization performance.

B. Feature Representation

Feature representation is one of the most central research directions over the past decades. The most widely used descriptors are color SIFT, gray SIFT plus color histogram [2], [33], [34] extracted from local patches. Boureau *et al.* [35] learn semantic representations of images by aggregating neighboring descriptors to form micro-features or visual phrases. Gao *et al.* [36] learn 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.

Recently, CNN features have achieved a breakthrough on a large number of benchmarks. Most approaches choose the output of a CNN as feature representation directly [6], [7], [24], [9]. However, CNN features still preserve a great deal of global spatial information. As demonstrated in [37], the activations from the fifth max-pooling layer can be reconstructed to form an image that looks very similar to the original one. The requirements of invariance to translation and rotation are weakly ensured by max-pooling. Though max-pooling helps improve invariance to small-scale deformations, invariance to larger-scale deformations might be undermined by the preserved global spatial information. To solve this issue, Gong *et al.* [38] propose to aggregate features of the fully connected layers via orderless VLAD pooling. Considering deeper layers are more domain specific and potentially less transferable than shallower layers, Cimpoi *et al.* [12] pool features from the convolutional layers, and achieve considerable improvements for texture recognition.

Our approach regards responses from deep CNN neurons as local descriptors (similar to SIFT), and encodes these localized responses via Fisher Vectors. Different from previous approaches that encode CNN descriptors globally [12], [38], we project each response back to the original image and encode each part separately. Most importantly, we propose a picking strategy which conditionally selects responses based on their importance for recognition, and encodes them via spatially weighted combination of Fisher Vectors.

III. Detector Learning: Picking Neurons for Localization

In this section, we target at learning a collection of detectors that could automatically discover discriminative object/parts. The strategy consists of two modules: positive sample initialization and regularized multiple instance detector learning. The first module generates initial sample groups, each of which is defined by a set of potentially positive samples of image patches. In the second module, we train detectors for each group with a regularized learning strategy, which iteratively updates positive samples via cross-validation, and meanwhile considers the reliability of each positive sample.

A. Picking Distinctive Neurons

Learning a detector requires a set of positive and negative examples, which should be identified in the training data. Different from previous methods, we develop a picking strategy which elaborately selects distinctive and consistent patches based on the responses of CNN neural activations. The key insight is that different layers of a CNN are sensitive to specific patterns, i.e., the initial lower layers often respond to corners and edge conjunctions, while the latter layers often correspond to more and more macro regions, from semantically meaningful parts to the whole object. In a sense, these deep neurons work as part detectors and the feature maps serve similar roles as detection scores. However, these part detectors are usually weak, and most of them are irrelevant to the fine-grained task. In order to adapt the pretrained network to the target domain, we continue stochastic gradient descent to fine-tune the CNN filters. We defer the details of network fine-tuning in Section VI and focus on positive sample mining in this section.

A typical CNN consists of several types of layers, e.g., convolutional, pooling, and fully connected layers. The convolutional layers are composed of several convolutional kernels to compute feature maps, and each neuron of a feature map is connected to a neighborhood of neurons in previous layer. The fully connected layers take all neurons in previous layers and connect them to every single neuron of current layer to perform global reasoning, while the pooling layers lower the computational burden of a CNN by reducing the number of connections between convolutional layers. For brief narrations, we refer the output of a CNN at the convolutional layers as *conv* layers (e.g., *conv5* for the 5th convolutional layers), and fully connect layers as *fc* layers (e.g., *fc7* for the 7th fully connected layers). According to the response properties of different patches, we discuss the mining process from two aspects, i.e., object-level positives and part-level positives.

1) Object-Level Positives: The object-level positive samples are obtained via the last soft-max regression layer of the fine-tuned network, which indicates how likely a proposal belongs to the corresponding subcategory. Given a training image I with label y ($y \in \{1, 2, ..., N\}$), we first generate T region proposals $X = \{x_1, ..., x_T\}$ with selective search [39]. Define the last regression layer output of a patch x_i as $f_{reg}(x_i)$ and its value at dimension y as $f_{reg}(x_i, y)$. The potential object-level positive x_o of image I is defined as the patch with the maximum response at dimension y

$$x_o = \arg\max_{x_i \in X} f_{reg}(x_i, y). \tag{1}$$

2) Part-Level Positives: The part-level positives cannot be obtained directly as the object-level ones, since they are not trained explicitly. Fortunately, the intermediate CNN neurons show clustering characteristics, i.e., there exist some neurons in the intermediate layers that are sensitive to the same part of an object (e.g., head of birds), and some others to another part (e.g., body of birds). However, only some of the neurons are responsible for our target parts [40], [41].

In order to find which neurons are distinctive for part discovery, we first generate a large pool of region proposals, and

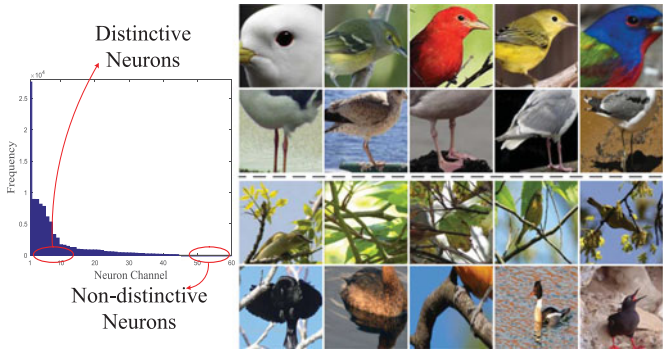


Fig. 3. Response distributions of the top scored $100K$ patches among randomly sampled one million patches from CUB-200-2011 (AlexNet, 256 channels). The top scored responses only focus on a few neuron channels, and we denote these channels as distinctive neurons. On the right we illustrate several top responding patches corresponding to the distinctive and non-distinctive neurons.

randomly sample one million patches. Each proposal is resized to a target size (e.g., 139×139 for AlexNet) to make the activation output of the last convolutional layer a single value per channel (similar to detection score). Then, we sort these patches by responses over all channels and pick the top scored 100 K patches. These responses are binned into corresponding channels according to which channel they respond most to. Finally, we get a response distribution of the top scored 100 K patches. As shown in Fig. 3, the response distributions are sparse, with most responses focusing on only a few channels (e.g., for CUB-200-2011, over 90% responses focus on the top 10% neural channels). We refer to the channels as distinctive neurons. In our experiment, the channels that include the top 90% responses are selected as distinctive neurons $\mathcal{F}$.

Although these picked neurons $\mathcal{F}$ are distinctive, they are rather weak and redundant, and we do not consider each single neuron performs well in detection. Inspired by the boosting algorithm in face detection [42], we aim to aggregate these weak neurons to boost detection performance for better initialization. We sort the distinctive neurons $\mathcal{F}$ via the quantity of responses over the top scored 100 K patches, and process them orderly. Specifically, for each neuron F_d ($F_d \in \mathcal{F}$) we find its k-nearest neighbors $N_k(F_d)$ ($N_k(F_d) \in \mathcal{F}$) in terms of cosine similarity and group them into a cluster. Every processed neuron F_d together with its k-nearest neighbors $N_k(F_d)$ are removed out of the queue. This procedure is repeated until no neuron exists in the queue. Formally, denote the convolutional output of a patch x_i at channel c as $f_{conv}(x_i, c)$, the potential part positive x_p corresponding to distinctive neuron F_d is obtained by

$$x_p = \arg\max_{x_i \in X} \sum_{F_c \in N_k(F_d)} f_{conv}(x_i, c). \tag{2}$$

The two-level initialization procedure is illustrated in Fig. 4. It shows that the initialization method can generate visually consistent patches, which is helpful for the following detector learning. On the contrary, k-means clustering usually behaves poorly in high dimensional space, producing visually inhomogeneous groups (shown Fig. 4). Moreover, in order to ensure the purity within each cluster, there are usually thousands of clusters [29], [30], which increases the complexity of the detector learning. As a comparison, our proposed initialization strategy only produces dozens of clusters, which is far less than that of k-means.

The neuron clustering is an extension of previous work [13], which reduces the number of detectors to be learned significantly. Before neuron clustering, the number of $\mathcal{F}$ usually ranges from 20 to 50, and varies according to different datasets and models. After neuron clustering process, the number of clusters feeded to MIL framework usually ranges from 10 to 20, e.g., for CUB-200-2011, the number of selected distinctive neurons is 27, after neuron clustering, we only need to learn 11 part detectors. While in [13], the number of detectors needed to learn is 27 (one detector for one neuron).

B. Regularized Multiple Instance Detector Learning

Given the initial positives with (1) or (2), we are able to learn the corresponding detector by explicitly optimizing a linear SVM classifier [23]. However, such a paradigm brings about two issues. First, due to the lack of annotations, the initial positives are not very good to begin with. Second, these initial positives often come from a few subcategories that are discriminative. Nevertheless, due to the large inter-class variations among subcategories, if a detector does not see any positive sample of one subcategory, it would localize badly on that one. On the other hand, including patches that do not correspond to the same part as the exemplars will decrease the localization and discrimination power of the part model.

To address these issues, we formulate the weakly supervised detector learning as a multiple instance learning (MIL) [28] problem, which aims at learning a more generalized detector. The initial positive samples are refined by an iterative update strategy which aims at mining better positive samples. Towards this goal, different from standard MIL which is based on alternatively selecting the highest detections as positive samples and refining the model on the same dataset, we employ a two-fold cross-validation to avoid overfitting the initial positive samples. Furthermore, since the mined positives are not equally reliable, a regularized strategy is introduced to measure the reliability of each positive sample. Different from [13] which simply mines per-category positive samples at each round of detector learning heuristically, we define the detector learning in a formal way, and prove a detailed analysis of the advantages of the proposed method.

Defining positive and negative bags: To use MIL for fine-grained detector learning, we divide the instances of each image into two bags, the positive bag that includes potential positive instances, and the negative bag that contains no or only a fractionlet of the object. We make use of the last regression output [defined in (1)] for defining the negatives. The patches with scores below a threshold (set as 0.2 for all experiments) are treated as negative instances and grouped as the negative bag, while all other instances are grouped as the positive bag. In the following, we define the problem in a formal way.

Problem formulation: Let $\mathcal{X}$ be the set of training bags, which consists of a set of positive bags $\mathcal{X}_p$ and negative bags $\mathcal{X}_n$, i.e.,

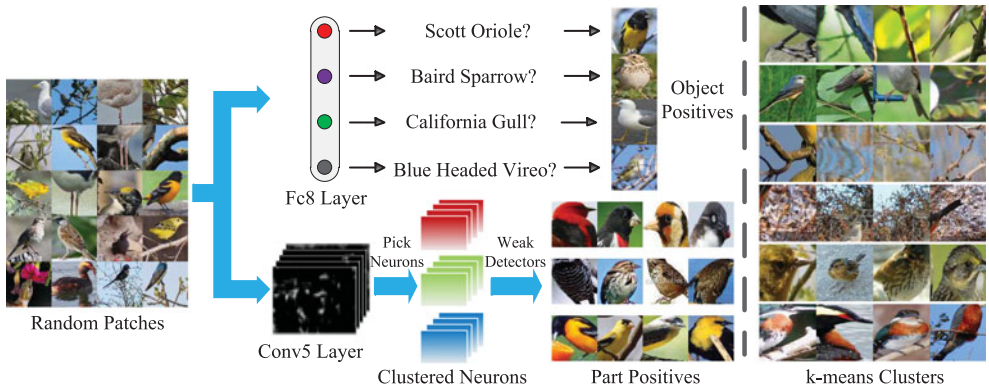


Fig. 4. Two-level initialization based on neuron selectivity. Our method can generate visually consistent patches, which is helpful for the following detector learning. On the contrary, clustering by k-means usually behaves poorly in high dimensional space, producing visually inhomogeneous groups.

$\mathcal{X} = \mathcal{X}_p \cup \mathcal{X}_n$. Let X be a bag of instances from an image. For any instance $x \in X$ from a bag $X \subseteq \mathcal{X}$, denote the corresponding feature vector as $\phi(x)$. The regularized MIL detector learning problem can be formulated as solving the following objective:

$$\begin{aligned} \min \quad & \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{X\subseteq\mathcal{X}_p}\Delta_X\xi_X + C\sum_{x\in\mathcal{X}_n}\xi_x \\ \text{s.t.} \quad & \boldsymbol{w}^T\Phi(X,z_X)+b \geq 1-\xi_X, \ \ \forall X\subseteq\mathcal{X}_p, \\ & \boldsymbol{w}^T\phi(x)+b \leq -1+\xi_x, \ \ \forall x\in\mathcal{X}_n, \\ & \xi_X \geq 0, \quad \xi_x \geq 0, \ \ \forall X\subseteq\mathcal{X}_p, \forall x\in\mathcal{X}_n \end{aligned} \tag{3}$$

where z_X indicates the location in X with maximum response: $z_X = \text{argmax}_{z\in X}(\boldsymbol{w}^T\Phi(X,z)+b)$, and Δ_X is the latent variable which measures the positiveness of a bag $X \subseteq \mathcal{X}_p$. ξ_X, ξ_x are the slack variables, and C is the control parameter of the loss term. In this formulation, only one instance per positive bag matters, while all the negative instances are taken into consideration.

Optimization: The regularized MIL leads to a non-convex optimization problem due to the introduction of the latent location variable z_X and the latent confidence variable Δ_X. However, this problem is semi-convex since the optimization problem becomes convex once these latent variables are fixed. In the following, we solve (3) via an iterative procedure which alternates between updating the latent variables z_X, Δ_X and optimizing the detector $\boldsymbol{w}$. In order to avoid overfitting the initial positive samples when updating and optimizing are performed on the same dataset, we introduce cross-validation which optimizes detector on one subset and updates latent variables on the other disjoint subset.

Specifically, the training set $\mathcal{D}$ is equally divided into two disjoint and complementary subsets $\mathcal{D}_1$ and $\mathcal{D}_2$. Given the initial positives on $\mathcal{D}_1$ [obtained with (1) and (2)], we first train a standard SVM detector $\boldsymbol{w}_{\mathcal{D}_1} = \boldsymbol{w}_0$. Then the detector $\boldsymbol{w}$ is refined via iteratively *Updating* latent variables and *Optimizing* (3).

1) Updating: The latent variables on $\mathcal{D}_2$ are determined by previous round detector $\boldsymbol{w}_{\mathcal{D}_1}$ trained on $\mathcal{D}_1$, i.e., $z_X = \text{argmax}_{z\in X}(\boldsymbol{w}_{\mathcal{D}_1}^T\Phi(X,z)+b)$, $\Delta_X = \sigma[\boldsymbol{w}_{\mathcal{D}_1}^T\Phi(X,z_X)+b]$,

Algorithm 1: Regularized Multiple Instance Detector Learning

Require: Disjoint training set $\mathcal{D}_1$ and $\mathcal{D}_2$;
Initialization: Select initial positive samples with (1) or (2), and train standard SVM detector $\boldsymbol{w}_0$.
Learning: Given $\boldsymbol{w}_{\mathcal{D}_1} = \boldsymbol{w}_0$, solving the regularized MIL issue in (3) via iteratively updating and optimizing on $\mathcal{D}_1$ and $\mathcal{D}_2$.
a). Updating: fix latent variables on $\mathcal{D}_2$ via $\boldsymbol{w}_{\mathcal{D}_1}$, i.e., $z_X = \text{argmax}_{z\in X}(\boldsymbol{w}_{\mathcal{D}_1}^T\Phi(X,z)+b)$, $\Delta_X = \sigma[\boldsymbol{w}_{\mathcal{D}_1}^T\Phi(X,z_X)+b]$.
b). Optimizing: solving (3) on $\mathcal{D}_2$ with updated latent variables. Then switch $\mathcal{D}_1$ and $\mathcal{D}_2$.
Ensure: Detector $\boldsymbol{w}_{\mathcal{D}_1}$.

where σ is a sigmoid function which maps the value into the range of $(0,1)$.

2) Optimizing: The detector is optimized according to the fixed latent variables on $\mathcal{D}_2$ via hard negative mining [23].

After each round of *Updating* and *Optimizing*, we switch $\mathcal{D}_1$ and $\mathcal{D}_2$ for further iteration. The whole detector learning strategy is summarized in Algorithm 1.

Corollary: The solution $\boldsymbol{w}$ of (3) is a linear combination of the positive samples $\phi(X,z_X)$ and the negative samples $\phi(x)$, i.e., $\boldsymbol{w} = \sum_{X\subseteq\mathcal{X}_p}\alpha_X\phi(X,z_X) + \sum_{x\in\mathcal{X}_n}\alpha_x\phi(x)$, where the coefficients α_X and α_x are bounded by: $0 \leq \alpha_X \leq C\Delta_X$, $0 \leq \alpha_x \leq C$, respectively.

Proof: The constrained minimization problem in (3) can be solved with a classical Lagrangian method. The Lagrangian operator can be represented as

$$\begin{aligned} \mathscr{L} = & \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{X\subseteq\mathcal{X}_p}\Delta_X\xi_X + C\sum_{x\in\mathcal{X}_n}\xi_x \\ & + \alpha_x(\boldsymbol{w}^T\phi(x)+b+1-\xi_x) - \sum_{x\in\mathcal{X}_n}\gamma_x\xi_x \\ & - \alpha_X(\boldsymbol{w}^T\Phi(X,z_X)+b-1+\xi_X) - \sum_{X\subseteq\mathcal{X}_p}\gamma_X\xi_X \end{aligned} \tag{4}$$

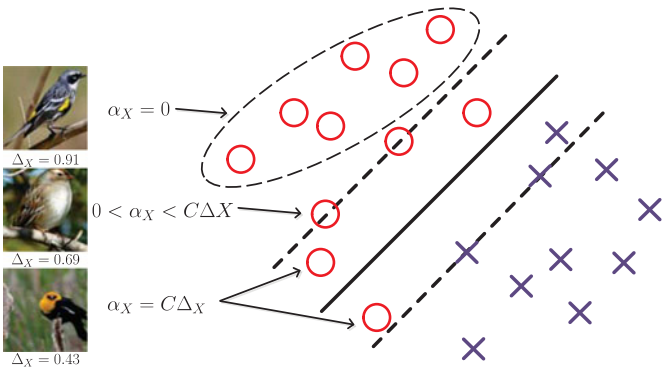


Fig. 5. Illustration of separating hyperplane in the feature space for SVM with regularized loss. The circles represent positive samples and the crosses represent negative ones. Support vectors are weighted by Δ_X which measures the reliability of detection in previous round.

where α_X, α_x, γ_X, and γ_x denote Lagrange multipliers. The minimization of Lagrangian operator in (4) with respect to $\boldsymbol{w}, \xi_X, \xi_x$ is

$$\begin{cases} \dfrac{\partial \mathscr{L}}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{X \subseteq \mathcal{X}_p} \alpha_X \phi(X, z_X) - \sum_{x \in \mathcal{X}_n} \alpha_x \phi(x), \\ \dfrac{\partial \mathscr{L}}{\partial \xi_X} = 0 \Rightarrow \gamma_X = \Delta_X C - \alpha_X, \\ \dfrac{\partial \mathscr{L}}{\partial \xi_x} = 0 \Rightarrow \gamma_x = C - \alpha_x. \end{cases} \tag{5}$$

Due to the nonnegativity of γ_X and γ_x, we have $0 \leq \alpha_X \leq C\Delta_X$ and $0 \leq \alpha_x \leq C$. For a test example $\tilde{x}$, the detection score can be represented as

$$f(\tilde{x}) = \left(\sum_{X \subseteq \mathcal{X}_p} \alpha_X \phi(X, z_X) - \sum_{x \in \mathcal{X}_n} \alpha_x \phi(x) \right) \phi(\tilde{x}) + b. \tag{6}$$

It can be seen that the final detection score $f(\tilde{x})$ is a weighted combination of the inner product between training features $\phi(X, z_X)$, $\phi(x)$ and test feature $\phi(\tilde{x})$, and is only determined by samples with nonzero coefficients α_i $(i = X, x)$. These α_i s are called support vectors, since they are the only training samples necessary to define the separating hyperplane. Note that for positive samples α_X is bounded by $C\Delta_X$, with KKT conditions, it is also possible to see when an example is a support vector, this happens only if the example is on the margin, or it does not respect the separation conditions in (3). According to [43], the coefficient α_X for positive samples in different locations is defined as

$$\begin{cases} \alpha_X = 0, & \boldsymbol{w}^T \phi(x_Z) + b > 1, \\ \alpha_X = C\Delta_X, & \boldsymbol{w}^T \phi(x_Z) + b < 1, \\ 0 < \alpha_X < C\Delta_X, & \boldsymbol{w}^T \phi(x_Z) + b = 1. \end{cases} \tag{7}$$

As shown in Fig. 5, for positive samples that do not respect the classification hyperplane, the corresponding coefficient α_X is bounded by $C\Delta_X$, which takes the reliability of z_X into consideration. The regularized term Δ_X helps to boost the detection performance. If a positive sample z_X is not reliable at previous round, its contribution to the classification hyperplane at current round would be lowered. MIL introduces more diverse samples for detector learning, while the regularized term encourages the detector focusing on positive samples that are good enough and downweighting those samples with lower reliability.

IV. SWFV-CNN: Picking Neurons for Description

With the above trained detectors, we can identify corresponding parts from each image. One intuitive method for part description is to directly extract features from the penultimate Fully-Connected (FC) layer of a CNN, which is widely used in previous methods. However, FC-CNN is not suitable to describe fine-grained details for two reasons. First, FC-CNN captures the spatial layout of an image, which is useful for representing the shape of an object, while it may be not useful for describing fine-grained details. Second, FC-CNN requires a fixed rectangle as input, which includes cluttered background inevitably. To deal with the first issue, we regard CNN activations as local descriptors [12] (similar to SIFT), and orderless pool these descriptors via Fisher Vector coding, which is apt at describing fine-grained details. For the second issue, a saliency map is utilized to pool CNN descriptors with Spatially Weighted Fisher Vectors (SWFV-CNN), which downweights the influences of the backgrounds.

A. Spatially Weighted FV-CNN

We give a brief introduction to Fisher Vectors for convenient narrations, and more details can be found in [44]. Let u_λ be a probability density function which models the generative process of descriptors across the dataset, and $X = \{\boldsymbol{x}_t, t = 1, ..., T\}$ be a set of T local descriptors extracted from an image. The Fisher Vector is defined by the gradient of the log-likelihood of X on the model

$$\mathscr{G}_\lambda^X = L_\lambda G_\lambda^X = \frac{1}{T} L_\lambda \sum_{i=1}^{T} \nabla_\lambda \log \mu_\lambda(\boldsymbol{x}_t) \tag{8}$$

where L_λ is the square-root of the inverse of the Fisher information matrix, which can be treated as a scale factor of the gradient vector. In our case, μ_λ is chosen as a gaussian mixture model with parameter $\lambda = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k : k = 1, ..., K\}$. Let $\gamma_t(k)$ be the posterior probability of each vector $\boldsymbol{x}_t$ to a mode k in the mixture model. The Fisher Vector $\mathscr{G}_\lambda^X = [\mathscr{G}_{\boldsymbol{\mu}_1}^X, ..., \mathscr{G}_{\boldsymbol{\mu}_K}^X, \mathscr{G}_{\boldsymbol{\sigma}_1}^X, ..., \mathscr{G}_{\boldsymbol{\sigma}_K}^X]$, which is the stacking of mean derivation vectors $\mathscr{G}_{\boldsymbol{\mu}_k}^X$ and covariance deviation vectors $\mathscr{G}_{\boldsymbol{\sigma}_k}^X$ for each of the K modes. Each entry of $\mathscr{G}_{\boldsymbol{\mu}_k}^X$ and $\mathscr{G}_{\boldsymbol{\sigma}_k}^X$ can be rewritten as follows:

$$\begin{aligned} \mathscr{G}_{\boldsymbol{\mu}_k}^X &= \frac{1}{T} \sum_{i=1}^{T} \mathscr{G}_{\boldsymbol{\mu}_k}^{\boldsymbol{x}_t} = \frac{1}{T\sqrt{w_k}} \sum_{i=1}^{T} \gamma_t(k) \left(\frac{\boldsymbol{x}_t - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right) \\ \mathscr{G}_{\boldsymbol{\sigma}_k}^X &= \frac{1}{T} \sum_{i=1}^{T} \mathscr{G}_{\boldsymbol{\sigma}_k}^{\boldsymbol{x}_t} = \frac{1}{T\sqrt{2w_k}} \sum_{i=1}^{T} \gamma_t(k) \left[\frac{(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1 \right] \end{aligned} \tag{9}$$

where the division should be understood as term-by-term operations. We formulate $\mathscr{G}_{\boldsymbol{\mu}_k}^X$ and $\mathscr{G}_{\boldsymbol{\sigma}_k}^X$ as accumulated average of the first and second order statistics of $\boldsymbol{x}$, respectively. However,

this kind of representation considers each $\boldsymbol{x}_t$ equally important, which is often not the case. The vector $\boldsymbol{x}_t$ may lie in non-salient regions. Considering this issue, a spatially weighted term $I_s(p_t)$ (p_t denotes the receptive field center of descriptor $\boldsymbol{x}_t$) is introduced for each vector $\boldsymbol{x}_t$, which indicates the importance of $\boldsymbol{x}_t$. The weighted results of $\tilde{\mathscr{G}}^X_{\boldsymbol{\mu}_k}$ and $\tilde{\mathscr{G}}^X_{\boldsymbol{\sigma}_k}$ can be expressed as

$$\tilde{\mathscr{G}}^X_{\boldsymbol{\mu}_k} = \sum_{i=1}^{T} I_s(p_t)\mathscr{G}^{\boldsymbol{x}_t}_{\boldsymbol{\mu}_k}, \quad \tilde{\mathscr{G}}^X_{\boldsymbol{\sigma}_k} = \sum_{i=1}^{T} I_s(p_t)\mathscr{G}^{\boldsymbol{x}_t}_{\boldsymbol{\sigma}_k}. \tag{10}$$

The weight term I_s is simply chosen as the saliency map [45] of an image, which indicates how likely a pixel belongs to salient regions. Since fine-grained images are not cluttered with many objects, and the object of interest is always the most salient region. The introduced spatial weights are able to catch the important details for recognition.

B. Distribution Analysis of SWFV-CNN

The proposed SWFV-CNN is a weighted version of FV-CNN, which is simple but effective. In this section, we provide one interpretation to explain why such a weighted combination can lead to improved results. Following [44], for a given image, we assume that the local descriptors can be decomposed into a mixture of two parts: an image independent background which follows the distribution μ_λ and an image specific part with distribution q. Let $\omega \in [0, 1]$ be the portion of image specific information, the generative model can be rewritten as: $p(X) = \omega q(X) + (1-\omega)\mu_\lambda(X)$, hence we have

$$\begin{aligned}\lim_{T\to\infty} G^X_\lambda &\approx \omega \nabla_\lambda E_{X\sim q} \log \mu_\lambda(X) \\ &\quad + (1-\omega)\nabla_\lambda E_{X\sim \mu_\lambda} \log \mu_\lambda(X)\end{aligned} \tag{11}$$

the second background term can be canceled out via a maximum likelihood estimation of parameter λ, which shows that Fisher Vectors discard the image independent information implicitly. However, the distribution depends on the image-specific proportion ω. This signifies that even for two images containing the same object but different scales, Fisher Vectors will have different signatures. Hence, the class separability is degraded by ω. Based on this observation, we introduce a saliency map which cancels the effect of ω, since we only pick descriptors focused on salient regions, then we have $\omega \approx 1$ for all images.

V. Experiments

A. Datasets

The empirical evaluation is performed on four fine-grained benchmarks: Caltech-UCSD Birds-200-2011 [16], Stanford Dogs [17], Aircraft [15], and Cars-196 [14], which are the most extensive and competitive datasets in fine-grained literature. Each dataset is endowed with specific statistic properties, which is crucial for recognition performance.

CUB-200-2011: This is the most widely used fine-grained dataset, which contains 11,788 images spanning 200 bird subspecies. Birds species are highly deformable, and usually occupy a small portion of the image area, which make the recognition challenging.

Stanford Dogs: This is a collection of dog species, which consists of 20,580 images with 120 classes. Compared with Birds dataset, the dogs in this dataset are more deformable, and usually suffers from partial occlusion and more complex backgrounds. Indeed, Dogs dataset is the most difficult one among the four datasets.

Aircraft: The dataset Aircraft is compound of 100 variants of airplanes with a total of 10,000 images. The task involves discriminating variants of a model such as Boeing 737–300 and 737–400, etc. Unlike birds and dogs, airplanes are rigid objects, and tend to occupy a long and narrow area of the image. Furthermore, the background is relatively clean, e.g., always sky or airport.

Cars-196: The dataset Cars-196 contains 16,185 images of 196 classes of cars, which are produced by different manufacturers. Similar to airplanes, cars are also rigid objects, which makes detection more feasible. Anther property is that cars usually occupy a relatively larger portion of areas in an image comparing with the above three datasets.

B. Network

1) Supervised Pre-training: Two typical network models are used in our experiments: AlexNet [46] and the more accurate but deeper one VGG-VD [47]. Note that for Stanford Dogs, since the complete dataset is a training subset of ILSVRC 2012, simply choosing the pre-trained network brings about cross-dataset redundancy. Considering this issue, we check ILSVRC 2012 training data and remove samples that are used as test in Dogs, then we train a network from scratch to obtain the model specific to Dogs. The pretrained network (AlexNet) nearly matches the performance of [46], obtaining a top-1 error rate of 44.2% on ILSVRC 2012 validation set.

2) Fine-Tuning With Saliency-Based Sampling: Fine-tuning is beneficial to adapt the network pretrained on ImageNet to the fine-grained tasks. Since fine-grained labels are expensive to obtain, most existing fine-grained datasets only contain a few thousand training samples, which is far from enough for fine-tuning. A common strategy is to introduce many "jittered" samples around the ground truth bounding box [23]. Instead, we develop a saliency-based sampling strategy to augment training samples, which does not need such annotations. The principle is based on the specific property of fine-grained datasets that the image is not cluttered with too many objects, and the object of interest is always the most salient regions. To this end, a saliency map I_s [45] of image I is computed, and for each region proposal $x_i \in X$, the saliency score $s(x_i \mid I_s)$ is defined as

$$s(x_i \mid I_s) = \frac{\sum I_s(x_i)}{\sum I_s} \tag{12}$$

where $\sum I_s(x_i)$ denotes the sum of the pixels within x_i. The patches with saliency scores above a threshold (set as 0.7) are defined as augmented samples, which expands the samples by approximately $20\times$. Table I shows the recognition results of the

TABLE I
RECOGNITION RESULTS OF THE SALIENCY-BASED FINE-TUNING METHOD ON NETWORKS ALEXNET [46] AND VGG-VD [47]

Dataset	AlexNet		VGG-VD	
	Original	fine-tune	Original	fine-tune
CUB-200-2011	57.0	61.7	66.6	74.0
Stanford Dogs	59.6	59.7	–	–
Aircraft	65.3	70.9	70.1	82.6
Cars-196	56.9	70.4	64.4	85.6

Accuracies are based on features (FC-CNN) extracted directly from the whole images.

fine-tuned networks. We observe that fine-tuning with saliency-based sampling consistently outperforms the pre-trained network, which indicates that for fine-grained datasets, bounding box information is unnecessary for network fine-tuning. The only exception is for Stanford Dogs, since the target training data is already included in the ImageNet data.

C. Implementation Details

Parameter settings: The k nearest neighbors in (2) is set to be 5. For both object-level and part-level positives, we choose the top 5% scored patches for initialization. In Section III, the pool5 features are chosen for detector training. In practice, the iteration process converges within several times, and we set the iteration times as 7. These parameters are fixed for all datasets for easy implementation.

FC-CNN: FC-CNN is extracted from the penultimate Fully-Connected (FC) layer of a CNN. The input image is resized to fixed size with mean subtracted before propagating through the CNN. similar to [46], FC-CNN is extracted from 10 view crops and then averaged for final representation.

FV-CNN: FV-CNN pools CNN features with Fisher Vectors. We extract *conv*5 descriptors (256-d for AlexNet, and 512-d for VGG-VD) at 3 scales ($s = \{256, 384, 512\}$), with each image rescaled to the target size so that $\min(w, h) = s$. We reduce the dimension of the descriptors to 80-d by PCA transformation and pool them into a FV representation with 256 Gaussian components, resulting in 40 K-d features. In practice, the *conv*5 descriptors are mapped back to the original image (the mapping in [48] is used), and each detected part is pooled into FV separately.

Feature combination: Since FV-CNN is high dimensional, combination of FV-CNN from different parts results in tremendously high dimensional features. To ease this issue, we adopt the method of [49] to learn one-vs-all features per part, and concatenate the learned features after normalization. The dimension of one-vs-all features scales with the number of categories, which is far less than that of FV-CNN.

Evaluation measurement: A conventional one-vs-all linear SVM is used for final classification. For accuracy evaluation, we use the default training/test split of the corresponding datasets, and choose the standard metric, i.e., the average classification accuracy by categories.

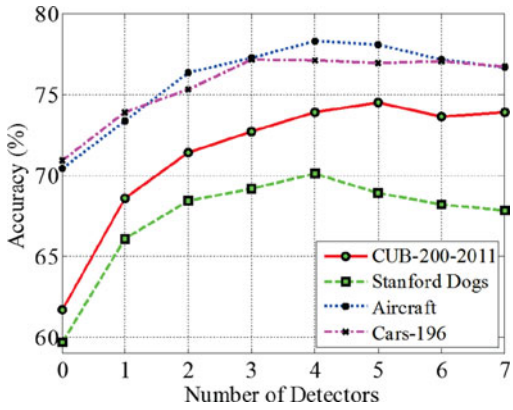


Fig. 6. Recognition performance when different number of detectors are added in. Number 0 refers to baseline without parts. Features are based on AlexNet with FC-CNN.

D. Ablation Study

1) *Performance versus number of detectors:* Our algorithm produces several to a dozen detectors, and there is no guarantee that these detectors will not return poor localization. In order to discard those detectors that are poorly localized, we measure the discriminative power of each detector in terms of recognition accuracy. Specifically, we equally divide the training set into two disjoint subsets, and perform cross-validation to measure the discriminative power of each detector. For each detector, classification is performed on the top scored regions, and the detectors are sorted based on their recognition accuracy. For final recognition, we progressively add the sorted detectors to uncover how it affects the performance. As shown in Fig. 6, the performance improves fast when the first several detectors are added in, while it tends to be stable and eventually drops as the number of detectors grows. It is intuitive that the bad detectors degrade the performance. For simplicity, we discard detectors that degrade the performance (e.g., the number of detector for Stanford Dogs is 4) and follow these settings for further experiments.
Comparing with [13] which only considers the part positives via the intermediate convolutional layers, we added the object-level positives via the last regression layer. Experimental results show that the introduction of object detectors improves the recognition results. We have reimplemented [13] with AlexNet model, and the improvement is obvious, e.g., for CUB-200-2011, previous method [13] achieves an accuracy of 72.1% with FC-CNN, which is inferior to the result of using additional object detector (74.5%).
2) *Probe performance step by step:* To understand which step is critical for recognition, we analyze results on CUB-200-2011 with different variants. As shown in Fig. 7, we set the number of detectors as 5 for fair comparisons (c.f. Fig. 6, which achieves the highest performance). The baseline denotes the method extracting features from the whole image, without any object/part information, and

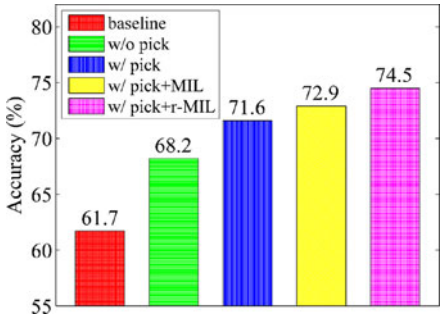


Fig. 7. Performance comparisons of different variants on CUB-200-2011. Features are based on AlexNet with FC-CNN.

used the features for recognition. As a comparison, w/o pick (w/pick) represents detection directly based on all neurons (picked neurons), MIL is the standard multiple instance learning method, and r-MIL is the regularized MIL strategy. The performance improves from 68.2% to 71.6% after neuron picking, which demonstrates the necessity of the picking strategy. As a matter of fact, the intermediate neurons are implicitly trained, and there is no guarantee that all the neurons serve as part detectors. It has been demonstrated that only a small portion of neurons emerge as semantic part detectors (e.g., 34 out of 123 on Pascal VOC 2007) even after network fine-tuning with ground truth part annotations [40]. On the other hand, the proposed regularized MIL (74.5%) achieves better performance than detector learning with standard MIL (72.9%), which demonstrates the effectiveness of the proposed detector learning strategy.

Fig. 8 shows some detection results of the learned detectors. We represent detections with red for discovered whole object and other colors for parts, and overlay them on the original image for better visualization. These detections exhibit surprisingly good visual consistency even without annotated training samples. For birds, they fire consistently on some visual meaningful structures such as head and body. While for dogs, they usually focus around head, mainly because other parts are either highly deformable or partially occluded. For rigid objects such as airplanes and cars, detections perform better. The discovered parts focus on the head, body, and tail of airplanes surprisingly well, and mainly concentrate on the head of cars. As a comparison, we also show the detections directly returned by the clustered neurons (the bottom row for each dataset), which is similar to the method [9]. These neurons usually return inferior localizations, which demonstrates the effectiveness of our part detectors. Note that these detectors are redundant (e.g., both detectors respond to the head of dogs) to some extent. However, their features have different representation and can enrich each other.

E. Fine-Grained Recognition Results

The performance of part detection can be further demonstrated in terms of recognition accuracy. As shown in Table II, we perform detailed analysis by comparing different variants of our method. "BL" refers to the baseline method, which extracts features directly from the whole image, without any knowledge of object or parts. "PD" refers to the proposed part detection method (Section III), and "SWFV-CNN" refers to the spatially weighted FV-CNN method (Section IV).

Due to the introduction of FOAF [49], the feature dimension scales with the number of categories. The advantage is that it reduces the classifier training time, e.g., for CUB-200-2011, the dimension of PD+SWFV-CNN is about 240K, and training classifier based on the high dimensional feature vector costs over 6 hours, while the dimension of FOAF is 1.2 K and the training time costs about 1.5 hours. Let N refer to the number of categories in each dataset (e.g., $N = 200$ for CUB-200-2011), and P refer to the number of feature parts concatenated. For convenient description, the number of feature parts includes features extracted from the whole image, the detected object and parts. Hence it equals 6 (c.f. Fig. 6) for CUB-200-2011, and is 5 for Stanford Dogs. From Table II we observe that:

1) Part detection boosts the performance significantly. Comparing with the baseline, PD brings about a 12.7% (61.7% $\rightarrow$ 74.5%) improvement on CUB-200-2011, a 10.4% (59.7% $\rightarrow$ 70.1%) improvement on Stanford Dogs. For Aircraft and Cars-196, the corresponding improvements are 5.4% (70.9% $\rightarrow$ 76.3%) and 6.7% (70.4% $\rightarrow$ 77.1%), respectively. Similar trends can be found when switching to a more powerful network VGG-VD.
2) FC-CNN is usually better than FV-CNN. FC-CNN usually outperforms FV-CNN by a considerable margin, e.g., on AlexNet, the differences are 74.5% vs 72.9% on CUB-200-2011, 70.1% vs 65.1% on Stanford Dogs, and 76.3% vs 74.4% on Aircraft. This is mainly because FV-CNN usually includes background information, which is confusing for fine-grained recognition, while FC-CNN alleviates this influence by max-pooling. The performance gap is large for Stanford Dogs, probably due to the fact dogs are usually with more cluttered backgrounds in this dataset. The only exception is for Cars-196, where FV-CNN is in turn better than FC-CNN (80.1% vs 77.1%). This is because in this dataset cars usually occupy a large portion of image and the background is relatively clean.
3) SWFV-CNN performs consistently better than FV-CNN, and even better than FC-CNN. We find that SWFV-CNN brings about at most over 4% improvement comparing with FV-CNN (65.1% $\rightarrow$ 69.3% on Stanford Dogs), and is even better than FC-CNN (e.g., on CUB-200-2011). The reason is that SWFV-CNN only focuses on features that are important for recognition, and deemphasizes those that are not helpful. The results demonstrate that comparing with FV-CNN, SWFV-CNN is more suitable for fine-grained recognition.
4) SWFV-CNN complements with FC-CNN. SWFV-CNN treats features as local descriptors and encodes them via orderless pooling, while FC-CNN represents images globally and still preserves rough spatial layout of images. The above properties make the two features complementary

Fig. 8. Sample detection results of the learned detectors on four fine-grained datasets. We represent detections with red for discovered whole object and other colors for parts, and overlay them on the original image for better visualization. As a comparison, We also show the detections directly returned by the picked filters (the bottom row for each dataset), which is similar to the method of [9]. Note that different detectors may return very similar results, which makes some part illustration be overwhelmed. The last two columns show some failure cases.

TABLE II
RECOGNITION RESULTS OF DIFFERENT VARIANTS OF OUR METHOD

Method	Dim.	CUB-200-2011		Stanford Dogs	Aircraft		Cars-196	
		AlexNet	VGG-VD	AlexNet	AlexNet	VGG-VD	AlexNet	VGG-VD
FC-CNN BL	N	61.7	74.0	59.7	70.9	82.6	70.4	85.6
FV-CNN BL	N	56.3	70.2	60.5	65.8	81.4	76.1	89.0
FC+FV-CNN BL	2N	66.0	74.8	63.8	72.3	84.7	79.5	89.3
PD+FC-CNN	NP	74.5	83.3	70.1	76.3	84.0	77.1	88.8
PD+FV-CNN	NP	72.9	79.8	65.1	74.4	83.4	80.1	89.2
PD+SWFV-CNN	NP	75.6	83.7	69.3	77.8	85.4	82.5	91.1
PD+FC+**SWFV-CNN**	2NP	**77.1**	**84.7**	**72.4**	**78.8**	**87.3**	**83.8**	**91.7**

"BL" refers to baseline which extracts features directly from the whole image. "PD" refers to part detection in Section III, and "SWFV-CNN" refers to the spatially weighted FV-CNN in Section IV. "N" refers to the number of categories in each dataset, and "P" refers to the number of feature parts.

TABLE III
RECOGNITION PERFORMANCE COMPARISONS ON CUB-200-2011

Method	Anno.		Accuracy(%)	
	Train	Test	Alex	VGG-VD
Ours PNA	n/a	n/a	**77.1**	**84.7**
PN-CNN [24]	bbox + parts	n/a	75.7	–
Part R-CNN [7]	bbox + parts	n/a	73.9	–
PS-CNN [22]	bbox + parts	bbox	–	76.2
SPDA-CNN [4]	bbox + parts	bbox	**81.0**	**85.1**
PG Alignment [6]	bbox	bbox	74.9	82.8
NAC [10]	n/a	n/a	68.5	81.0
TL Atten. [9]	n/a	n/a	69.7	77.9
STN [27]	n/a	n/a	84.1	
Bilinear CNN [11]	n/a	n/a	84.1	

"n/a" refers to not available, while "bbox" and "parts" refer to object bounding box and part annotations.

with each other. Experimental results show that it brings about 1%∼3% improvement when combining SWFV-CNN with FC-CNN. We obtain an accuracy of 77.1% on Birds, 72.4% on Dogs, 78.8% on Aircraft, and 83.8% on Cars-196. Replacing AlexNet with VGG-VD improves the performance in all the cases, with a final accuracy of 84.7% for Birds, 87.3% for Aircraft, and 91.7% for Cars-196.

F. Comparisons With Prior Methods

We now move on to compare our results with some typical previous methods. Each dataset is provided with object annotations, which are more or less used in most previous methods. We categorize each method according to whether the annotations are used at train/test time, which influences the performance significantly.

CUB-200-2011: Table III shows the comparison results of our method with prior approaches on CUB-200-2011. There are a large number of previous approaches reporting results on this dataset, we only include results with CNN features for fair comparisons. Most previous approaches rely on object-level or even part-level annotations for recognition [4], [7], [24]. Our approach is better than those methods which rely on object-level [6] or even part-level [7], [22] annotations, and is 0.4% worse than [4] (84.7% vs 85.1%) which needs annotations at both training and test time.

We now compare our results with other methods. We are not the first to apply CNN neurons for part localization [9], [10]. The differences are that [9] trains the network from scratch and directly groups all the intermediate neurons for part detection, which is inferior to our method consisting of two activation picking steps. [10] constructs neural activation constellation model by selecting neurons that fire at similar relative locations, which is similar to deformable part model [25]. However, animals such as birds and dogs are highly deformable to model, which limits the recognition performance. Our results (77.1% on AlexNet and 84.7% on VGG-VD) are much better than these two methods (69.7% on AlexNet and 77.9% on VGG-VD in [9], and 68.5% on AlexNet and 81.0% on VGG-VD in [10]). Our method is also slightly better than the bilinear CNN [11] (84.1%) and the spatial transform networks [27] (84.1%). However, these two methods are trained based on much larger networks (448×448 input image), and the bilinear networks are the combinations of two deep models. The results indicate that fully automatic fine-grained recognition is within reach.

TABLE IV
RECOGNITION PERFORMANCE COMPARISONS ON STANFORD DOGS

Method	Train anno.	Test anno.	Accuracy
Ours PNA	n/a	n/a	**72.4**
Temp. Match [50]	bbox	bbox	38
Symbolic [2]	bbox	bbox	45.6
Alignment [33]	bbox	bbox	57
	n/a	n/a	49
FOAF [49]	bbox	bbox	71.6
NAC [10]	n/a	n/a	68.6

TABLE V
RECOGNITION PERFORMANCE COMPARISONS ON AIRCRAFT

Method	Train anno.	Test anno.	Accuracy
Ours PNA	n/a	n/a	**87.3**
Symbolic [2]	bbox	bbox	75.8
Revisit FV [51]	n/a	n/a	80.7
Bilinear CNN [11]	n/a	n/a	84.1
Multi-grained [52]	n/a	n/a	82.5

TABLE VI
RECOGNITION PERFORMANCE COMPARISONS ON CARS-196

Method	Train anno.	Test anno.	Accuracy
Ours PNA	n/a	n/a	**91.7**
Symbolic [2]	bbox	bbox	78.0
Revisit FV [51]	n/a	n/a	82.7
Bilinear CNN [11]	n/a	n/a	91.3
PG Alignment [6]	bbox	bbox	**92.8**

Stanford Dogs: Table IV shows the comparison results on Dogs. Few methods report results on this dataset, since off-the-shelf CNN models cannot be used for feature extraction. The most comparable result with our method is [10], which also trains AlexNet model from scratch and obtain an accuracy of 68.6%. Our method improves it by over 3%, with an error rate reduction of 12.2%.

Aircraft and Cars-196: The comparison results on Aircraft and Cars-196 are shown in Tables V and VI, respectively. For Aircraft, our approach brings about an improvement of 3.2% over the previous best performing method [11]. For cars-196, our result (91.7%) is comparable with the best performing method [11] (91.3%) under the same supervision, and a little short (about 1%) of the state-of-the-art results of [6] which uses extra annotations of object at both training and test time.

VI. DISCUSSION

A. Why Bother to Train SVMs After Network Fine-Tuning?

One issue is that: why bother to train SVMs after fine-tuning? It would be cleaner to simply apply the output of the fine-tuned

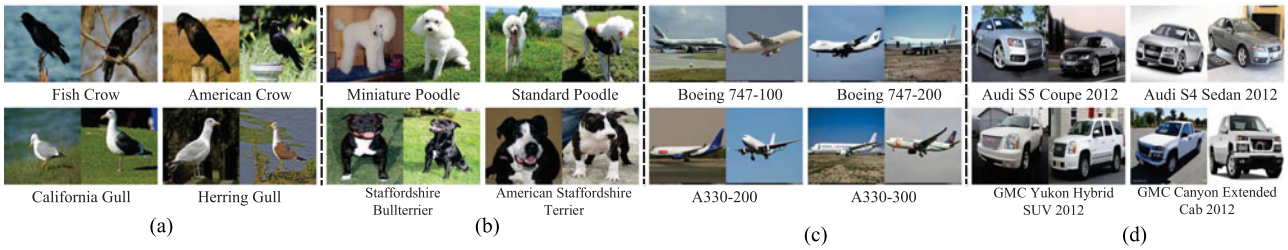


Fig. 9. Top two pairs of subcategories that are most confused with each other in each dataset. In hindsight, these subcategories are difficult for recognition merely by appearances. (a) CUB-200-2011, (a) CUB-200-2011, (c) Aircraft, and (d) Cars-196.

network as detectors, which is the last soft-max layer for object and the fifth convolutional layer for parts. We try this setting and find that performance drops dramatically. For comparison, we show some detection results directly based on the responses of the neurons. As shown in Fig. 8, the detectors directly from the neurons are weak, and in most situations localize inaccurately. Accordingly, the recognition performance drops considerably, e.g., when extracting features with FC-CNN on AlexNet, it drops from 74.5% to 71.6% on CUB-200-2011, and from 70.1% to 67.9% on Stanford Dogs. The performance differences likely arise from a combination of several factors. First, the positive samples used for fine-tuning do not emphasize precise localization and the soft classifier is trained on randomly sampled negative samples, rather than "hard negative mining" used for SVM training. Second, the fifth convolutional layer is still much weaker in part mining than that of the soft-max classifier layer in object mining, since they are not trained explicitly. Third, for part initialization, an image patch is resized to the theoretical size of the receptive field in *conv*5 layer, and we hope that the activation is responsible for the whole patch. However, as demonstrated in [40], the actual receptive field of *conv*5 is much smaller than the theoretical one, which inevitably introduce inaccurate positive samples. Based on these observations, we mine positive samples by way of boosting weak detectors, and iteratively train SVMs via regularized loss term to improve the robustness of detectors.

B. What Are the Limits of Visual Features?

Despite the promising performance achieved by our approach, there is still a certain gap from practical application, e.g., on CUB-200-2011, the accuracy 84.7% is much lower than that of expert birders (93%) [53]. Fig. 9 shows several subcategories that are the least successful in our results. The failure of classification is mainly 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. The observations suggest that fine-grained species classification is a difficult problem and is not always possible with a single image. In this case, a better solution may need human intervention such as questions posed to the users.

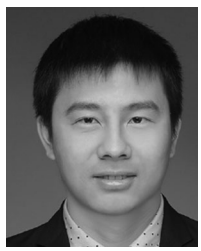
VII. Conclusion

In this paper, we develop a framework for fine-grained recognition which is free of any object/part annotation at both training and testing stages. Our method incorporates deep neural activations for both part localization and description. We claim two major contributions. Firstly, a picking strategy is utilized to select distinctive neurons that respond to specific parts significantly and consistently. Based on these picked neurons, we choose positive samples and train a set of discriminative detectors via a regularized multiple instance detector learning. Secondly, we develop a simple but effective feature encoding method, which we call SWFV-CNN. SWFV-CNN packs local CNN descriptors via spatially weighted combination of Fisher Vectors, which considers the importance of Fisher Vector for recognition. Integrating the above schemes produces a powerful framework, and shows notable performance improvements on several widely used fine-grained datasets.

References

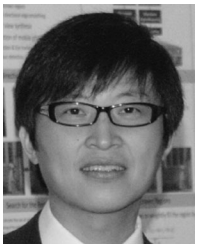
[1] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[2] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *Proc. Int. Conf. Comput. Vis.*, pp. 321–328, 2013.
[3] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, "Hierarchical part matching for fine-grained visual categorization," in *Proc. Int. Conf. Comput. Vis.*, pp. 1641–1648, 2013.
[4] H. Zhang *et al.*, "SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1143–1152, 2016.
[5] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep LAC: Deep localization, alignment and classification for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1666–1674, 2015.
[6] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5546–5555, 2015.
[7] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNS for fine-grained category detection," in *Proc. Eur. Conf. Comput. Vis.*, pp. 834–849, 2014.
[8] X.-S. Wei, J.-H. Luo, J. Wu and Z. H. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2868–2881, 2017.
[9] 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.*, pp. 842–850, 2015.
[10] M. Simon and E. Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," in *Proc. Int. Conf. Comput. Vis.*, pp. 1143–1151, 2015.
[11] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. Int. Conf. Comput. Vis.*, pp. 1449–1457, 2015.

[12] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3828–3836, 2015.
[13] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian, "Picking deep filter responses for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1134–1142, 2016.
[14] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. Int. Conf. Comput. Vis. Workshops*, pp. 554–561, 2013.
[15] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *CoRR*, 2013. [Online]. Available: http://arxiv.org/abs/1306.5151
[16] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, CNS-TR-2011-001, 2011.
[17] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog., Fine-Grained Vis. Categorization Workshop*, vol. 2, 2011.
[18] Y. Wang, S. Li, and A. C. Kot, "Deepbag: Recognizing handbag models," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 2072–2083, Nov. 2015.
[19] O. M. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman, "The truth about cats and dogs," in *Proc. Int. Conf. Comput. Vis.*, pp. 1427–1434, 2011.
[20] 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.*, pp. 955–962, 2013.
[21] C. Göring, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric part transfer for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2489–2496, 2014.
[22] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked CNN for fine-grained visual categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1173–1182, 2016.
[23] 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.*, pp. 580–587, 2014.
[24] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Improved bird species reognition using pose normalized deep convolution nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–14.
[25] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8, 2008.
[26] Y. Zhang *et al.*, "Weakly supervised fine-grained categorization with part-based image representation," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1713–1725, Apr. 2016.
[27] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Neural Inf. Process. Syst.*, pp. 2017–2025, 2015.
[28] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Neural Inf. Process. Syst.*, pp. 561–568, 2002.
[29] J. Sun and J. Ponce, "Learning discriminative part detectors for image classification and cosegmentation," in *Proc. Int. Conf. Comput. Vis.*, pp. 3400–3407, 2013.
[30] S. Singh, A. Gupta, and A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proc. Eur. Conf. Comput. Vis.*, pp. 73–86, 2012.
[31] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, no. 1, pp. 31–71, 1997.
[32] S. Ray and M. Craven, "Supervised versus multiple instance learning: An empirical comparison," in *Proc. Int. Conf. Mach. Learn.*, pp. 697–704, 2005.
[33] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *Proc. Int. Conf. Comput. Vis.*, pp. 1713–1720, 2013.
[34] L. Xie, J. Wang, B. Zhang, and Q. Tian, "Fine-grained image search," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 636–647, May 2015.
[35] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Comput. Soc. Conf. Vis. Pattern Recognit.*, pp. 2559–2566, 2010.
[36] 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.
[37] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, pp. 818–833, 2014.
[38] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Proc. Eur. Conf. Comput. Vis.*, pp. 392–407, 2014.
[39] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.
[40] A. Gonzalez-Garcia, D. Modolo, and V. Ferrari, "Do semantic parts emerge in convolutional neural networks?," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1607.03738
[41] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNS," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[42] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
[43] R. Collobert, "Large scale machine learning," Idiap Res. Inst., Martigny, Switzerland, RR-04-42, 2004.
[44] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher Vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, 2013.
[45] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Neural Inf. Process. Syst.*, pp. 545–552, 2006.
[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, pp. 1097–1105, 2012.
[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2014.
[48] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. Eur. Conf. Comput. Vis.*, pp. 346–361, 2014.
[49] X. Zhang, H. Xiong, W. Zhou, and Q. Tian, "Fused one-vs-all features with semantic alignments for fine-grained visual categorization," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 878–892, Feb. 2016.
[50] S. Yang, L. Bo, J. Wang, and L. G. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *Proc. Neural Inf. Process. Syst.*, pp. 3122–3130, 2012.
[51] P.-H. Gosselin, N. Murray, H. Jégou, and F. Perronnin, "Revisiting the fisher vector for fine-grained classification," *Pattern Recognit. Lett.*, vol. 49, pp. 92–98, 2014.
[52] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang, "Multiple granularity descriptors for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2399–2406, 2015.
[53] S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie, "The ignorant led by the blind: A hybrid human–machine vision system for fine-grained categorization," *Int. J. Comput. Vis.*, vol. 108, nos. 1/2, pp. 3–29, 2014.



Xiaopeng Zhang received the B.S. degree in electronic engineering from Sichuan University, Sichuan, China, in 2011, and is currently working toward the Ph.D. degree in electronic engineering at the Shanghai Jiao Tong University, Shanghai, China.

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



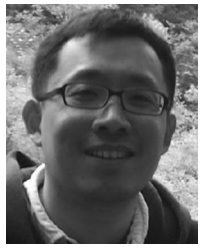
Hongkai Xiong (M'01–SM'10) received the Ph.D. degree in communication and information system from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2003.

Since then, he has been with the Department of Electronic Engineering, SJTU, where he is currently a Full Professor. From December 2007 to December 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, San Diego, CA, USA. He has authored or coauthored more than refereed journal/conference papers. His research interests include source coding/network information theory, signal processing, computer vision, and machine learning.

Dr. Xiong served as a TPC Member for prestigious conferences such as ACM Multimedia, ICIP, ICME, and ISCAS. Since 2012, he has been a member of Innovative Research Groups of the National Natural Science. He was the recipient of the Top 10% Paper Award at the 2016 IEEE Visual Communication and Image Processing (IEEE VCIP16), the Best Student Paper Award at the 2014 IEEE Visual Communication and Image Processing (IEEE VCIP14), the Best Paper Award at the 2013 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (IEEE BMSB13), and the Top 10% Paper Award at the 2011 IEEE International Workshop on Multimedia Signal Processing (IEEE MMSP11). In 2016, he was granted the Yangtze River Scholar Distinguished Professor from the Ministry of Education of China, and the Youth Science and Technology Innovation Leader from the Ministry of Science and Technology of China. He was awarded as the Shanghai Academic Research Leader. In 2014, he was granted National Science Fund for Distinguished Young Scholar and Shanghai Youth Science and Technology Talent as well. In 2013, he was the recipient of the Shanghai Shu Guang Scholar. In 2011, he was the recipient of the First Prize of the Shanghai Technological Innovation Award for Network-oriented Video Processing and Dissemination: Theory and Technology. In 2010 and 2013, he was the recipient of the SMC-A Excellent Young Faculty Award of Shanghai Jiao Tong University. In 2009, he was the recipiet of the New Century Excellent Talents in University, Ministry of Education of China.

Wengang Zhou received the B.E. degree in electronic information engineering from Wuhan University, Wuhan, 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.

From September 2011 to 2013, he was a Postdoctoral Researcher with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA. He is currently an Associate Professor with the Department of Electronic Engineering and Information Science, USTC. His research interests include multimedia information retrieval and computer vision.



Weiyao Lin (M'10–SM'16) received the B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2003 and 2005, respectively, and the Ph.D. degree from the University of Washington, Seattle, WA, USA, in 2010.

He is currently an Associate Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. He has authored or coauthored more than 100 technical papers, including over 40 refereed journal papers and one book chapter. His research interests include image/video processing, video surveillance, and computer vision.

Prof. Lin served as an Associate Editor of the *Journal of Visual Communication and Image Representation, Signal Processing: Image Communication*, the *Journal of Circuits, Systems, and Signal Processing*, and IEEE ACCESS.

Qi Tian (M'96–SM'03–F'16) received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 1992, the M.S. degree in electronics and communication engineering from Drexel University, Philadelphia, PA, USA, in 1996, and the Ph.D. degree in electronics and communication engineering from the University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA, in 2002.

He is currently a Full Professor with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA. He was a tenured Associate Professor from 2008 to 2012 and a tenure-track Assistant Professor from 2002 to 2008. During 2008 to 2009, he took a one-year Faculty Leave at Microsoft Research Asia, Beijing, China, as a Lead Researcher with the Media Computing Group. He was a Visiting Scholar with the MIAS Center of UIUC in 2007 and a Visiting Professor with NEC Laboratories of America in 2003. His research interests include multimedia information retrieval, computer vision, pattern recognition, and bioinformatics, and he has authored or coauthored more than 310 refereed journal and conference papers.

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