Picking Deep Filter Responses for Fine-grained Image Recognition

Xiaopeng Zhang[1] Hongkai Xiong[1] Wengang Zhou[2] Weiyao Lin[1] Qi Tian[3]

[1] Shanghai Jiao Tong University [2] University of Science and Technology of China [3] University of Texas at San Antonio
{zxphistory,xionghongkai,wylin}@sjtu.edu.cn zhwg@ustc.edu.cn qitian@cs.utsa.edu

Abstract

Recognizing fine-grained sub-categories such as birds and dogs is extremely challenging due to the highly localized and subtle differences in some specific parts. Most previous works rely on object/part level annotations to build part-based representation, which is demanding in practical applications. This paper proposes an automatic fine-grained recognition approach which is free of any object/part annotation at both training and testing stages. Our method explores a unified framework based on two steps of deep filter response picking. The first picking step is to find distinctive filters which respond to specific patterns significantly and consistently, and learn a set of part detectors via iteratively alternating between new positive sample mining and part model retraining. The second picking step is to pool deep filter responses via spatially weighted combination of Fisher Vectors. We conditionally pick deep filter responses to encode them into the final representation, which considers the importance of filter responses themselves. Integrating all these techniques produces a much more powerful framework, and experiments conducted on CUB-200-2011 and Stanford Dogs demonstrate the superiority of our proposed algorithm over the existing methods.

1. Introduction

As an emerging research topic, fine-grained recognition aims at discriminating usually hundreds of sub-categories belonging to the same basic-level category. It lies between the basic-level category classification (*e.g.* categorizing bikes, boats, cars, and so on in Pascal VOC [8]) and the identification of individual instances (*e.g.* face recognition). An inexperienced person can easily recognize basic-level categories like bikes or horses immediately since they are visually very dissimilar, while it is difficult for him/her to tell a black bird from a crow without specific expert 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 similar birds). Hence localizing and describing object and the corresponding parts become crucial for fine-grained recognition.

Figure 1. Illustration of filter selectivity for a typical network VGG-M [4] on CUB-200-2011. We generate candidate patches with selective search [25] and compute response of each patch at conv4 layer. We show several top responding patches of some channels and observe that there exist some filters which respond to specific patterns (*e.g.*, the head or leg of bird), while most of them respond chaotically. This paper proposes to pick deep filters with significant and consistent responses, and learn a set of discriminative detectors for recognition.

In order to achieve accurate object and part locations, most existing works explicitly require object level or even part level annotations at both training and testing stages [3], [28], [33]. However, such a requirement is demanding in practical applications. Some works consider a more reasonable setting, *i.e.* object/part level annotations at only training stage but not at testing time [15], [32]. However, even with such a setup, it still requires expensive annotations at training stage, and is especially hard 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 we refer to automatic part discovery. However, discovering parts automatically is a classical chicken-and-egg problem, *i.e.* without an accurate appearance model, examples of a part cannot be discovered, and an accurate appearance model cannot be learned without having part examples. Some pioneering works begin to consider this issue [21], [27]. How-

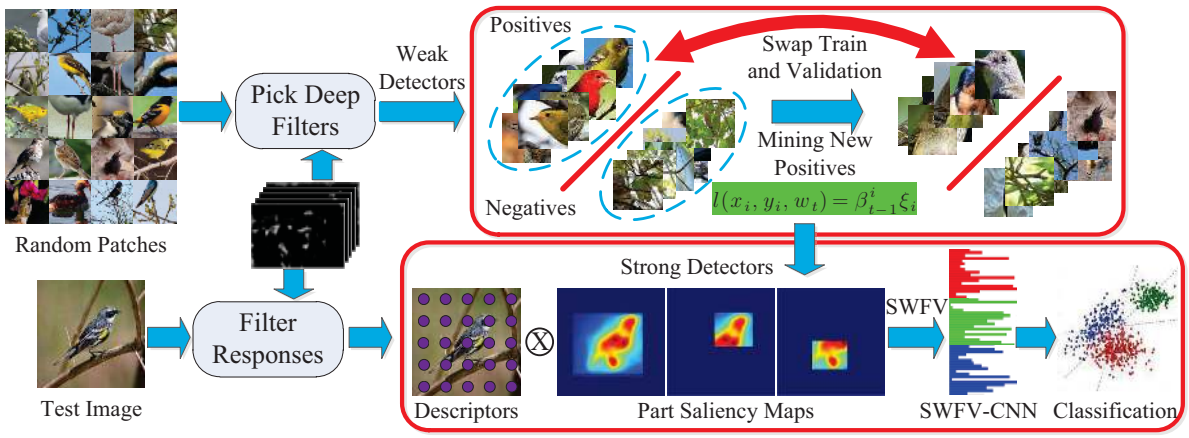


Figure 2. An overview of our proposed framework. Our approach consists of two picking steps. The first step aims at picking deep filters which respond to specific patterns significantly and consistently. Based on these picked filters, we choose positive samples and train a set of discriminative detectors iteratively. The second step is to pick filter responses via Spatially Weighted Fisher Vector (SWFV) encoding. We assign each Fisher Vector a weight and pool it into final image representation, which considers the importance of Fisher Vector itself.

ever, these methods either needs a network trained from scratch [27], or suffers complex optimization [21], and the performance is limited.

As our **first contribution**, we propose an automatic part detection strategy for fine-grained recognition (Sec. 3), which *is free of any object/part level annotation at both training and testing stages*. Our detection method consists of two main contributions. First, we propose a novel initialization method for detector learning, which is based on the selectivity of deep filters. As illustrated in Fig. 1, which shows some top responding patches of some filters on CUB-200-2011. It can be found that some filters work as part detectors and respond to specific parts (*i.e.*, the head of bird). However, these detectors are weak and most of them are not relevant to our task. The key insight of our initialization approach is to elaborately pick deep filters with significant and consistent responses. Second, we propose to learn a set of detectors via iteratively per-category positive sample mining and regularized part model retraining. We mine new positive samples by category and introduce a regularized term for each positive sample, which considers both the diversity and reliability of positive samples. The learned detectors tend to discover discriminative and consistent patches which are helpful for part-based recognition.

Feature representation is another key issue for fine-grained recognition. Recently, Convolutional Neural Network (CNN) has been widely used for feature extraction. However, there exist two challenges for fine-grained representation. The first is that traditional CNN representation requires fixed size rectangle as input, which inevitably includes background information. However, background is unlikely to play any major role for fine-grained recognition since all sub-categories share similar background (*e.g.* all birds usually inhabit on the tree or fly in the sky). The second is the gap between detection and classification. Due to large pose variation and partial occlusion, detection may be unreliable and lose crucial details for recognition.

To address the above challenges, as our **second contribution**, we propose a new kind of feature which is suitable for fine-grained representation (Sec. 4). We regard deep filter responses of a CNN as localized descriptors, and encode them via Spatially Weighted Fisher Vector (SWFV-CNN). The key insight is that not all filter responses are equally important for recognition. Our goal is to highlight the responses which are crucial for recognition and discount those which are less helpful. To this end, we propose a picking strategy which conditionally selects descriptors based on part saliency map, which indicates how likely a pixel belongs to a foreground part. Experimental results demonstrate that SWFV-CNN performs consistently better than traditional CNN, and is complementary with traditional CNN to further boost the performance.

• **Framework overview.** An overview of our proposed framework is shown in Fig. 2. Our approach consists of two picking steps. The first step aims at picking deep filters which respond to specific patterns significantly and consistently. Based on these filters, we elaborately select positive samples which are semantically similar and train a set of discriminative detectors. We use an iterative procedure 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 filters via Spatially Weighted combination of Fisher Vector, which we refer to SWFV-CNN. We compute spatial weights with part saliency map, which indicates how likely a pixel belongs to a foreground part. The part saliency map is used to weight

each Fisher Vector and pool it into final image representation, which considers the importance of Fisher Vector itself.

The rest of this paper is organized as follows. Sec. 2 describes related work on fine-grained categorization. The details of our proposed part discovery strategy is elaborated in Sec. 3. In Sec. 4, we describe our proposed Spatially Weighted FV-CNN. Experimental results and discussions are given in Sec. 5. Sec. 6 concludes the paper.

2. Related Works

Fine-grained recognition is a challenging problem and has recently emerged as a hot topic. In the following, we organize our discussion related to fine-grained recognition with two tasks: part localization and feature representation.

2.1. Part Localization

As fine-grained datasets are often provided with extra annotations of bounding box and part landmarks [18], [19], [26], most works rely on these annotations more or less.

Early works 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 [1], [17], [28]. Obviously, this kind of setting is demanding and a more reasonable setting only assumes the availability of object bounding box. Chai *et al.* [3] introduce techniques that improve both segmentation and part localization accuracy by simultaneous segmentation and detection. Gavves *et al.* [10] propose a supervised alignment method which retrieves nearest neighbor training images for a test image, and regresses part locations from these neighboring training images to the test image.

Later works 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.* [32] generalize the R-CNN [11] framework to detect parts as well as the whole object. Branson *et al.* [2] train a strongly supervised model in a pose normalized space. Further on, Krause *et al.* [15] propose a method which only need object level annotations at training time, and is completely unsupervised at the level of parts.

Recently, there have been some emerging works which 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 step towards making fine-grained recognition suitable for wide deployment. Xiao *et al.* [27] propose to use two attention models with deep convolutional networks, one to select relevant patches to a certain object, and the other to localize discriminative parts. Simon *et.al.* [21] propose to localize parts with constellation model, which incorporates CNN into deformable part model [9].

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 works [21], [27], we learn a set of discriminative detectors via elaborately selecting positive samples and iteratively updating part models.

2.2. Feature Representation

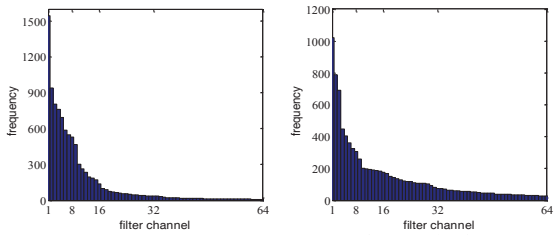
For the description of image, CNN features have achieved breakthrough on a large number of benchmarks [11], [20], [31], *etc*. Different from traditional descriptors which explicitly encode local information and aggregate them for global representation, CNN features represent global information directly, and can alleviate the requirement of manually designing a feature extractor. Though not specifically designed to model sub-category level differences, CNN features capture such information well [7].

Most works choose the output of a CNN as feature representation directly [2], [15], [27], [32]. However, CNN features still preserve a great deal of global spatial information. As demonstrated in [31], the activations from the fifth max-pooling layer can be reconstructed to form an image which 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.* [12] 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.* [6] pool features from the convolutional layers, and achieve considerable improvements for texture recognition.

Our approach regards responses from deep CNN filters as localized descriptors (similar with SIFT), and encodes these responses via Fisher Vector. Different from previous works which encode CNN descriptors globally [6], [12], 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.

3. Learning Part Detectors

In this section, we target at learning a collection of discriminative detectors that automatically discover discriminative object / parts. Our strategy consists of three modules: positive sample initialization, regularized detector training, and detector selection. The first module generates initial parts, each of which is defined by a set of potentially positive samples of image patches. In the second module, we train detectors for each set of positive samples with a regularized iterative strategy. To remove those noisy detectors, the third module select good detectors by measuring their predictive power in terms of recognition accuracy. Note that



(a) CUB-200-2011 (b) Stanford Dogs

Figure 3. Response distributions of the top scored 10K patches on VGG-M (512 channels). The top scored responses only focus on a few channels. We remove the channels with lower response frequency for better visualization.

the full procedure is weakly supervised, which only needs the labels of training examples, while does not need any object / part level annotation.

3.1. Picking Filters: Positive Sample Initialization

Learning a part detector requires a set of part examples, which should be identified in the training data. Most previous works employ some form of unsupervised clustering, such as k-means [23], [24], or template matching [30], to initialize a part model. However, running k-means or template matching on mid-level patches does not return very good clusters, and often produces clustered instances which are in no way visually similar.

Different from previous works, we propose a picking strategy which elaborately selects distinctive and consistent patches based on the responses of CNN filter banks. The key insight is that different layers of a CNN are sensitive to specific patterns. *e.g.*, the lower layers often respond to corners and other edge conjunctions, while the higher layers often correspond to semantically meaningful regions. In a sense, these convolutional filters work as part detectors. However, these detectors are usually weak, and most of them are not relevant to our fine-grained tasks.

In order to find which filters are distinctive for part discovery, we first generate a large pool of region proposals with selective search [25], and randomly sample a subset of one million patches. Each proposal is resized to a target size of 107×107, which makes the activation output of the 4th convolutional layer a single value (similar with detection score). Then, we sort responses over all channels and pick the top scored 10K responses. 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 10K regions. 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 5% channels). We refer to these channels as distinctive filters, which respond to specific patterns significantly. In our experiment, we select channels which include the top 90% responses as distinctive

Algorithm 1 Learning Discriminative Part Detector

Require: Disjoint training subsets $\{D_1, D_2\}$;
1: initialization $N = \{(x_i, y_i)\}_{i=1}^{m} \in D_1, \beta = [1, ...1]_m$
2: **while not** converged **do**
3: *Detector* $\boldsymbol{w} \Leftarrow$ svm_train (N, β)
4: $[N_{new}, \beta_{new}] \Leftarrow$ top $(\boldsymbol{w}, D_2,$ m$) \bigcup$ per_top $(\boldsymbol{w}, D_2,$ k$)$
5: $N \Leftarrow N_{new}, \beta \Leftarrow \beta_{new}$
6: swap (D_1, D_2)
7: **end while**
8: **Return** *Detector* $\boldsymbol{w}$

filters. For each distinctive filter, we select patches with the top m ($m = 100$) responses as initial positives for the corresponding part model. Fig. 1 visualizes some top responding regions for distinctive and non-distinctive channels. The responses of distinctive filters always focus on consistent parts, such as the head of birds. While non-distinctive filters pick up some cluttered samples.

3.2. Regularized Detector Training

With the initialization of positive samples, we learn the corresponding detector by optimizing a linear SVM classifier. We define the negatives based on Intersection over Union (IoU) overlap with the positives, and the regions with IoU overlap below 0.3 are treated as negative samples. Since negative samples are much larger than the positives, we adopt the standard hard negative mining method [9], which converges quickly after only a single pass over all images.

Iterative update. Since the initial positives are not very good to begin with (as shown in the first row of Fig. 4, some samples are biased), we train SVM detector iteratively. During each iteration, the top 10% firings of previous round detector are used as new positive samples. However, doing this directly does not produce much improvement since the detector tends to overfit to the initial positives, and would prefer these positives during the next round of validation. To solve this issue, we divide the training samples into two equal, non-overlapping subsets, which enables us to achieve better generalization by training on one subset while validating on another. We then exchange the role of training and validation and repeat this whole process until convergence (the learned detector does not change).

Regularized Loss Term. Another issue of training object / part detectors for all the fine-grained sub-categories is that the top detections always latch on a few easy detectable subcategories, and cannot discover positive samples from the majority of other sub-categories. Due to the large inter-class variations among sub-categories, if a detector does not see any positive sample of one sub-category, it would localize badly on that one. However, including patches that do not correspond to the same part as the exemplars will decrease the localization and discrimination power of part

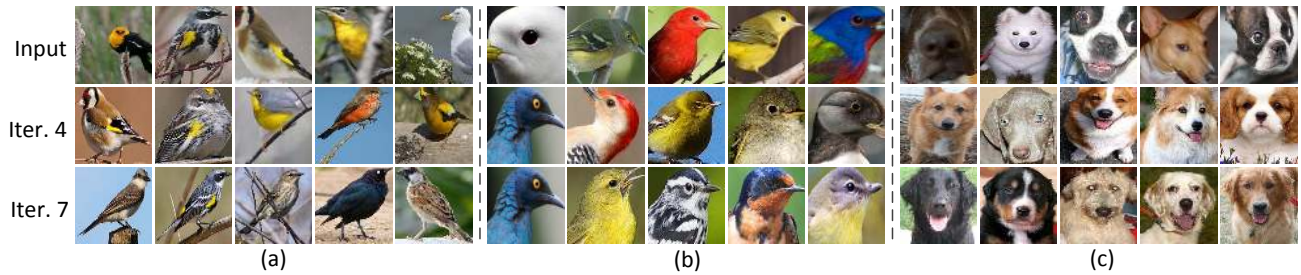


Figure 4. Positive samples during the detectors learning process in different iteration steps. The first row is the initial positive samples and rows 2-3 show new positive samples returned by the top detections of previous round detectors. Even though the initial positive samples are not well localized, our algorithm is able to mine new samples which exhibit visual consistency, and learn a set of discriminative detectors.

model. To solve this issue, we mine per-category positive samples with regularized loss during each round of training. Specifically, the top 10% detections per-category are used as positives as well as the top 10% detections among all subcategories. Since these potential positives are not equally reliable, we assign a weight term β to each positive sample, which measure the reliability of each positive.

Denote $D = \{(x_i, y_i)\}_{i=1}^{n}$ be the set of positive and negative training patches, and $\boldsymbol{x_i}$ its corresponding feature vector of x_i, where $y_i \in \{-1, 1\}$. The part detector $\boldsymbol{\omega_t}$ during round t can be learned by minimizing the following function:

$$\begin{aligned} min \quad & \frac{1}{2}\|\boldsymbol{\omega_t}\|^2 + C\sum_{i=1}^{n}\beta_{t-1}^{i}\xi_i \\ s.t. \quad & y_i(\boldsymbol{\omega_t}^T\boldsymbol{x_i} + b_t) \geq 1 - \xi_i, \quad i = 1, ..., n \\ & \xi_i \geq 0, \quad i = 1, ..., n, \end{aligned} \tag{1}$$

where

$$\beta_{t-1}^{i} = \begin{cases} \Psi(\boldsymbol{\omega_{t-1}^T}\boldsymbol{x_i} + b_{t-1}), & y_i = 1 \\ 1, & y_i = -1, \end{cases} \tag{2}$$

where $\Psi[\cdot]$ is a sigmoid function which maps the detection scores within range $(0, 1)$, and C controls relative weights of the loss terms. Note that we introduce an extra regularized term β_{t-1}^{i} for each positive sample x_i, which measures the reliability of x_i with detection score of previous round detector. The regularized term highlights the higher scored patches and downweights the lower scored patches.

Note that there are two benefits for our regularized detector learning. First, with per-category positive sample mining, the detector can see more diverse positives, which is beneficial for its generalization. Second, with the introduced regularized term β, the detector is able to avoid overfitting the less reliable positives, while focusing on the more reliable positives. Fig. 4 shows some detector learning process in different iteration steps. Our algorithm is able to mine positive samples which are visually consistent, even though the initial positives are not well localized. As the iteration goes, the positives become more and more consistent, which in turn boosts the discriminative power of part model. The full approach for detector learning is summarized in Algorithm 1.

3.3. Detector Selection

Our algorithm produces tens of detectors, and there is no guarantee that the part mining procedure will not return bad detectors. In order to discard those detectors which are poorly localized, we measure the discriminative power of detectors in terms of recognition accuracy. We equally divide the labeled training samples into training and validation subsets. For each detector, classification is performed based on the top scored region. Finally, we discard detectors with recognition rate below 40%, which reduces the detectors to only a few (less than ten in our experiments).

4. Bag of Parts Image Representation

With the above trained detectors, we can identify patches corresponding to the parts from each image. One direct method for part representation is to extract CNN features directly from the detected parts, and concatenate them for final representation. This kind of features are usually obtained from the penultimate Fully-Connected (FC) layer of a CNN, and are widely used in previous works. However, there are two limitations of FC-CNN for fine-grained recognition. The first is the background disturbance, as CNN requires a fixed rectangle as input, which includes cluttered background inevitably. The second comes from the inaccuracy of detections, which may lose crucial details for part-based representation. To deal with these issues, instead of extracting FC-CNN within a tight rectangle, we propose to compute part saliency map and pool CNN features with Spatially Weighted Fisher Vector (SWFV-CNN).

Part saliency map. The part saliency map is used to indicate how likely a pixel belongs to a foreground part. Our part saliency map consists of two sources, part map and saliency map. The part map indicates the spatial prior of a part, and is obtained simply from the top detection. The saliency map [13] is a topographically arranged map that represents visual saliency of a corresponding scene. Since fine-grained images are not cluttered with many objects, and

Figure 5. Sample detection results of our automatically discovered detectors. We select detections with top three recognition accuracies (shown in red, green, and blue in order), and overlay them to original image for better visualization (Row 1 and 3). We also show the detections directly returned by the picked filters (Row 2 and 4), which is similar with the method [27]. Our detectors improve localization power via iterative training, while detectors directly from the filters are weak, and in most situations localize inaccurately. The top two rows for CUB-200-2011, and the bottom two rows for Stanford Dogs. The last three columns show some failure cases.

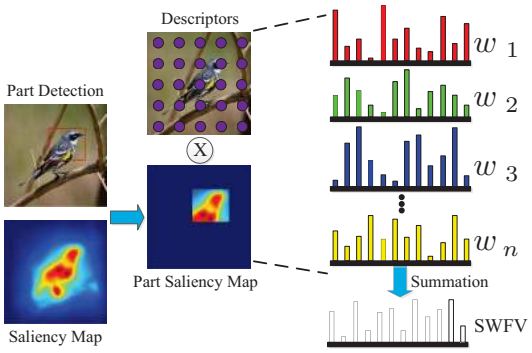


Figure 6. Illustration of how to compute SWFV-CNN. We first compute part saliency map with the top detections and saliency map. The part saliency map assign weight to each descriptor, and SWFV-CNN is the weighted combination of each Fisher Vector.

the object of interest is always the most salient region, we choose saliency map S to measure the existence probability of foreground object. The final part saliency map M is obtained as follows:

$$M(p) = \frac{S(p)\sum_{i=1}^{k} D_i(p)}{Z}, \tag{3}$$

where $D_i(p) = 1$ when the ith detection contains the pixel p, otherwise $D_i(p) = 0$. Z is a normalization constant which makes $max\ M(p) = 1$.

Spatially weighted FV-CNN. The Fisher Vector models the distribution of a set of vectors with gaussian mixture models and represents an image by considering the gradient with respect to the model parameters. Let $I = (\boldsymbol{z_1}, ..., \boldsymbol{z_N})$ be a set of D dimensional feature vectors extracted from an image. Define $\Theta = (\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}, \boldsymbol{\pi_k} : k = 1, ..., K)$ be the parameters of a gaussian mixture model fitting the distribution of descriptors, and q_{ik} be the posterior probability of each vector $\boldsymbol{z_i}$ $(i = 1, ..., N)$ to a mode k in the mixture model. For an image I, the Fisher Vector $\Phi(I) = [\boldsymbol{u_1}, \boldsymbol{v_1}, ..., \boldsymbol{u_k}, \boldsymbol{v_k}]$, which is the stacking of mean derivation vectors $\boldsymbol{u_k}$ and covariance deviation vectors $\boldsymbol{v_k}$ for each of the K modes. Each entry of $\boldsymbol{u_k}$ and $\boldsymbol{v_k}$ can be rewritten as follows:

$$\begin{aligned} u_{jk} &= \sum_{i=1}^{N} u_{ijk} = \sum_{i=1}^{N} \frac{q_{ik}}{N\sqrt{\pi_k}} \frac{z_{ji} - \mu_{jk}}{\sigma_{jk}} \\ v_{jk} &= \sum_{i=1}^{N} v_{ijk} = \sum_{i=1}^{N} \frac{q_{ik}}{N\sqrt{2\pi_k}} \left[\left(\frac{z_{ji} - \mu_{jk}}{\sigma_{jk}} \right)^2 - 1 \right], \end{aligned} \tag{4}$$

where $j = 1, ..., D$ spans the vector dimension. We formulate u_{jk} and v_{jk} as accumulated sum of the first and second order statistics of z_{ij}, respectively. However, this kind of representation considers each $\boldsymbol{z_i}$ equally important, which is often not the case. The vector $\boldsymbol{z_i}$ may lie in non-salient regions, or less reliable detected regions. Considering this issue, we introduce a spatially weighted term $M(p_i)$ for each vector $\boldsymbol{z_i}$, which indicates the importance of $\boldsymbol{z_i}$ itself. The weighted results of u_{jk} and v_{jk} can be expressed as:

$$u^w_{jk} = \sum_{i=1}^{N} M(p_i) \cdot u_{ijk}, \quad v^w_{jk} = \sum_{i=1}^{N} M(p_i) \cdot v_{ijk}, \tag{5}$$

with the introduced spatial weights, we are able to catch the important features for recognition. We would see its effectiveness in the following section. An illustration of how to compute SWFV-CNN of an image is shown in Fig. 6.

5. Experiments

5.1. Datasets

The empirical evaluation is performed on two benchmarks: Caltech-UCSD Birds-200-2011 (Birds) [26] and S-

tanford Dogs (Dogs) [14], which are the most extensive and competitive datasets in fine-grained literature. Birds dataset contains 11,788 images spanning 200 sub-species, while Dogs dataset consists of 20,580 images with 120 dog species. We use the default training/test split, which gives us around 30 training examples per class for Birds and around 100 training examples per class for Dogs.

5.2. Network

• **Supervised pre-training.** For Birds, two different models are used in our experiments: VGG-M [4] and a more accurate but deeper one VGG-VD [22]. Since Dogs 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 (AlexNet) from scratch to obtain the model specific to Dogs.

• **Fine-tuning with saliency-based sampling.** Fine-tuning is beneficial to adapt the network pretrained on ImageNet to our fine-grained tasks. Since 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 [11]. Instead, we propose a saliency-based sampling strategy without such annotation information. To this end, we compute a saliency map S [13] of an image. For each region proposal x generated with selective search [25], we compute the saliency score with $s(x|S) = \sum_{i \in x} S_i / \sum S$. The regions with saliency score above a threshold (set as 0.7 in our experiments, which expands the samples by approximately 20×) are chosen as augmented samples. This enables them to have high quality in containing the object of interest.

There are two benefits for network fine-tuning. First, the fine-tuned network is a better feature extractor for classification, *e.g.*, when fine-tuning on VGG-M [4], our proposed saliency-based sampling strategy achieves an accuracy of 66.97% on Birds, which is even better than the bounding box based sampling method in [11] (66.08%). This indicates that for fine-grained datasets, bounding box information is unnecessary for network fine-tuning. Second, the internal responses of convolutional filters are more domain specific, which helps for part selection in Sec. 3.

5.3. Implementation Details

• **Detector learning.** In Sec. 3, we choose pool5 features for detector training. In practice, the iteration process converges within several times, and we set the iteration times as 7. It only remains several detectors after selection (Sec. 3.3), and the number is 6 for Birds and 5 for Dogs.

• **FC-CNN.** FC-CNN is extracted from the penultimate Fully-Connected (FC) layer of a CNN. The input image is resized to fixed size and mean subtracted before propagating through the CNN. FC-CNN is widely used in previous works [2], [32], *etc.*, so we include it for fair comparison.

• **FV-CNN.** FV-CNN pools CNN features with Fisher Vector. We extract conv5 descriptors (512-d for VGG-M, VGG-VD, and 256-d for AlexNet) 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 to 128-d by PCA transformation and pool them into a FV representation with 256 Gaussian components, resulting in $65K$-d features.

Method	Birds VGG-M	Birds VGG-VD	Dogs AlexNet
FC-CNN BL	66.97%	73.98%	59.67%
FV-CNN BL	58.71%	70.21%	60.52%
FC+FV-CNN BL	71.03%	74.77%	63.75%
PD+FC-CNN	76.74%	82.60%	65.07%
PD+FV-CNN	73.83%	79.76%	63.11%
PD+FC+FV-CNN	78.58%	82.78%	69.84%
PD+SWFV-CNN	77.26%	83.58%	66.25%
PD+FC+**SWFV-CNN**	**80.26%**	**84.54%**	**71.96%**

Table 1. Recognition results of different variants of our method. We test models VGG-M [4] and VGG-VD [22] on CUB-200-2011 and AlexNet [16] on Stanford Dogs. "BL" refers to baseline method which extracts features directly from the whole image, without any knowledge of object or parts. "PD" refers to our proposed part detection method in Sec. 3, and "SWFV-CNN" refers to our spatially weighted FV-CNN method proposed in Sec. 4.

5.4. Results and Comparisons

We first conduct a detailed analysis of our method with regard to part detection and recognition performance, and move on to compare with prior works.

• **Part detection.** Fig. 5 shows some detection results (Row 1 and 3) of our learned detectors. We select detections with top three recognition accuracies (shown in red, green, and blue in order), and overlay them to the original image for better visualization. These detections exhibit surprisingly good visual consistency even without annotated training samples. For Birds, they fire consistently and represent a diverse set of parts (*e.g.*, object, head, and leg). While for Dogs, they usually focus around head, mainly due to the fact that other parts are either highly deformable or partial occluded. We also show detections (Row 2 and 4) directly returned by the picked filters, which is similar with the method [27]. These filters are not task relevant and usually return inferior localization results to ours, which demonstrates the effectiveness of our part detectors. Note that these detectors are redundant (*e.g.*, both detectors respond to dog's head) to some extent, however, their features have different representation and can enrich each other.

• **Recognition results.** The performance of part detection can be further demonstrated in terms of recognition accuracy. As shown in Table 1, we perform detailed analysis

Method	Train anno.	Test anno.	Accuracy
Ours PDFS	n/a	n/a	**84.54%**
GPP [28]	bbox+parts	bbox+parts	66.35%
Symbolic [3]	bbox	bbox	59.4%
POOF [1]	bbox	bbox	56.78%
Alignment [10]	bbox	bbox	67%
	n/a	n/a	53.6%
PN-CNN [2]	bbox+parts	bbox+parts	**85.4%**
	bbox+parts	n/a	75.7%
Part R-CNN [32]	bbox+parts	bbox+parts	76.37%
	bbox+parts	n/a	73.89%
FOAF [34]	bbox+parts	bbox+parts	81.2%
PG Alignment [15]	bbox	bbox	82.8%
NAC [21]	n/a	n/a	81.01%
TL Atten. [27]	n/a	n/a	77.9%

Table 2. Recognition performance comparisons on Birds. "bbox" and "parts" refer to object bounding box and part annotations.

by comparing different variants of our method. "BL" refers to baseline method, which extracts features directly from the whole image, without any knowledge of object or parts. "PD" refers to our proposed part detection method (Sec. 3), and "SWFV-CNN" refers to our spatially weighted FV-CNN method (Sec. 4). From Table 1 we observe that:

1) Part detection boosts the performance significantly. Comparing with the baseline, PD brings about a nearly 10% (66.97% → 76.74%) improvement for Birds, and an 5.5% improvement for Dogs. Note that the performance improvement on Dogs is less than that on Birds, mainly due to the larger deformations and more frequent occlusions on Dogs.

2) FC-CNN is usually better than FV-CNN. FC-CNN usually outperforms FV-CNN by around 2% ∼ 3% (76.74% vs 73.83% for Birds, and 65.07% vs 63.11% for Dogs). This is because FV-CNN usually includes background information, which is confused for fine-grained recognition. While FC-CNN alleviates this influence by max-pooling.

3) SWFV-CNN performs consistently better than FV-CNN, and even better than FC-CNN. We find that SWFV-CNN brings about over 3% improvement comparing with FV-CNN, and is even better than FC-CNN. The reason is that SWFV-CNN focuses on features which are important for recognition, and deemphasizes those which are not helpful. The results demonstrate that SWFV-CNN is more suitable for fine-grained recognition.

4) SWFV-CNN complements with FC-CNN. When combining SWFV-CNN with FC-CNN, we obtain an accuracy of 80.26% for Birds, and 71.96% for Dogs, which demonstrates the complementation of these features. Replacing VGG-M with VGG-VD improves the performance in all the cases, with a final accuracy of 84.54% for Birds.

• **Comparisons with prior works.** Table 2 shows the comparison results of our method with prior works on Birds. We list the amount of annotations of each method for fair comparison. Early works [1], [3], [28] choose SIFT as features, and the performance is limited. When switching to CNN features, our approach is best among methods under the same setting [21], [27], and obtains a 18% error reduction comparing with the best performing result [21] (81.01%). Moreover, our result even outperforms methods which use object [15] (82.8%) or even part [32] (76.37%), [34] (81.2%) annotations, only beaten by [2] (85.4%) which uses both object and part annotations at both training and testing time. Our method indicates that fully automatic fine-grained recognition is within reach.

Method	Train anno.	Test anno.	Accuracy
Ours PDFS	n/a	n/a	**71.96%**
Temp. Match [29]	bbox	bbox	38%
Symbolic [3]	bbox	bbox	45.6%
Alignment [10]	bbox	bbox	57%
	n/a	n/a	49%
Selec. Pooling [5]	bbox	bbox	52%
FOAF [34]	bbox	bbox	53.5%
NAC [21]	n/a	n/a	68.61%

Table 3. Recognition performance comparisons on Dogs.

Table 3 shows the comparison results on Dogs. Few works report results on this dataset, due to there are not off-the-shelf CNN models for feature extraction. The most comparable result with our method is [21], which also trains AlexNet model from scratch and obtain an accuracy of 68.61%. Our method improves it by over 3%, an error rate reduction of 10.7%.

6. Conclusion

In this paper, we propose a framework for fine-grained recognition which is free of any object/part annotation at both training and testing stages. Our method incorporates deep convolutional filters for both part detection and description. We claim two major contributions. Firstly, we propose to pick good filters which respond to specific parts significantly and consistently. Based on these picked filters, we elaborately choose positive samples and train a set of discriminative detectors iteratively. Secondly, we propose 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. Integrating the above approaches produces a powerful framework, and shows notable performance improvements on CUB-200-2011 and Stanford Dogs.

Acknowledgements. The work was supported in part by the National Science Foundation of China (NSFC), under contract No. 61425011, 61271218, U1201255, and 61529101, 61471235, and 61429201, in part to Dr. Zhou by Anhui Provincial Natural Science Foundation under contract No. 1508085MF109, and in part to Dr. Tian by ARO grants W911NF-15-1-0290 and W911NF-12-1-0057 and Faculty Research Gift Awards by NEC Laboratories of America and Blippar, respectively.

References

[1] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013. 3, 8

[2] S. Branson, G. Van Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. In *BMVC*, 2014. 3, 7, 8

[3] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013. 1, 3, 8

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 1, 7

[5] G. Chen, J. Yang, H. Jin, E. Shechtman, J. Brandt, and T. X. Han. Selective pooling vector for fine-grained recognition. In *AWACV*. IEEE, 2015. 8

[6] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *CVPR*, 2015. 3

[7] J. Donahue, Y. Jia, and O. e. Vinyals. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 3

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 1

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 3, 4

[10] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013. 3, 8

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 3, 7

[12] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014. 3

[13] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006. 5, 7

[14] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR FGVC workshop*, 2011. 7

[15] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *CVPR*, 2015. 1, 3, 8

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 7

[17] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur. Dog breed classification using part localization. In *ECCV*. Springer, 2012. 3

[18] O. M. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman. The truth about cats and dogs. In *ICCV*, pages 1427–1434. IEEE, 2011. 3

[19] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *CVPR*, 2012. 3

[20] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPR workshop*, 2014. 3

[21] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *ICCV*, 2015. 1, 2, 3, 8

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. 7

[23] S. Singh, A. Gupta, and A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 4

[24] J. Sun and J. Ponce. Learning discriminative part detectors for image classification and cosegmentation. In *ICCV*, 2013. 4

[25] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. 1, 4, 7

[26] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 3, 6

[27] T. Xiao, Y. Xu, and K. e. a. Yang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, 2015. 1, 2, 3, 6, 7, 8

[28] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang. Hierarchical part matching for fine-grained visual categorization. In *ICCV*, 2013. 1, 3, 8

[29] S. Yang, L. Bo, J. Wang, and L. G. Shapiro. Unsupervised template learning for fine-grained object recognition. In *NIPS*, 2012. 8

[30] B. Yao, G. Bradski, and L. Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *CVPR*, pages 3466–3473, 2012. 4

[31] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In *ECCV*, 2014. 3

[32] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, 2014. 1, 3, 7, 8

[33] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013. 1

[34] X. Zhang, H. Xiong, W. Zhou, and Q. Tian. Fused one-vs-all mid-level features for fine-grained visual categorization. In *ACM Multimedia*, 2014. 8