Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization

Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, Chunhua Shen

PII: S0031-3203(17)30399-0
DOI: 10.1016/j.patcog.2017.10.002
Reference: PR 6314

To appear in: *Pattern Recognition*

Received date: 17 June 2017
Revised date: 3 September 2017
Accepted date: 6 October 2017

Please cite this article as: Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, Chunhua Shen, Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization, *Pattern Recognition* (2017), doi: 10.1016/j.patcog.2017.10.002

Highlights

- To the best of our knowledge, Mask-CNN is the first end-to-end model that selects deep convolutional descriptors for object recognition, especially for fine-grained image recognition.
- We present a novel and efficient part-based three-stream model for fine-grained recognition. By discarding the fully connected layers, the proposed M-CNN is computationally efficient (cf. Table 1 and Table 4 in experiments). Additionally, comparing with state-of-the-art methods, M-CNN has smaller feature dimensionality. Beyond those, it achieves the highest classification accuracy on *CUB200-2011* and Birdsnap among published methods.
- The part localization performance of the proposed model outperforms other part-based finegrained approaches which requires additional bounding boxes. In particular, M-CNN is 12.76% higher than state-of-the-art for head localization on *CUB200-2011*.

Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization

Xiu-Shen Wei[a,1], Chen-Wei Xie[a,1], Jianxin Wu[a,*], Chunhua Shen[b]

[a] *National Key Laboratory for Novel Software Technology, Nanjing University, China.*
[b] *The University of Adelaide, Adelaide, Australia.*

Abstract

Fine-grained image recognition is a challenging computer vision problem, due to the small inter-class variations caused by highly similar subordinate categories, and the large intra-class variations in poses, scales and rotations. In this paper, we prove that selecting useful deep descriptors contributes well to fine-grained image recognition. Specifically, a novel Mask-CNN model without the fully connected layers is proposed. Based on the part annotations, the proposed model consists of a fully convolutional network to both locate the discriminative parts (*e.g.*, head and torso), and more importantly generate weighted object/part masks for selecting useful and meaningful convolutional descriptors. After that, a three-stream Mask-CNN model is built for aggregating the selected object- and part-level descriptors simultaneously. Thanks to discarding the parameter redundant fully connected layers, our Mask-CNN has a small feature dimensionality and efficient inference speed by comparing with other fine-grained approaches. Furthermore, we obtain a new state-of-the-art accuracy on two challenging fine-grained bird species categorization datasets, which validates the effectiveness of both the descriptor selection scheme and the proposed Mask-CNN model.

Keywords: Fine-grained image recognition, deep descriptor selection, part localization.

1. Introduction

Fine-grained recognition tasks such as identifying the species of birds [1, 2], flowers [3, 4] and cars [5], have been popular in applications of computer vision and pattern recognition. Since the categories are all similar to each other, different categories can only be distinguished by slight and subtle differences, which makes fine-grained recognition a challenging problem. Compared to the general

*Corresponding author
Email address: wujx2001@gmail.com (Jianxin Wu)
[1] The first two authors contributed equally to this work.

(a) Part annotations (b) Part rectangles

Figure 1: We generate the rectangles (in Fig. 1b) for the bird's head and torso based on the part annotations (red, blue and yellow dots in Fig. 1a). Other pixels are treated as background. The two yellow part key points (*i.e.*, nape and throat) are included in both head and torso. (Best if viewed in color.)

object recognition tasks, fine-grained recognition benefits more from learning critical parts of the objects, which helps discriminate different subclasses and align objects of the same class [6, 7, 8, 9, 10, 11, 12, 13].

A straightforward way to represent parts is to use the deep convolutional features/descriptors. The convolutional descriptors contain more localized (*i.e.*, parts) information compared to the feature of the fully connected layers (*i.e.*, whole image). In addition, these deep descriptors are known to correspond to mid-level information, *e.g.*, object parts [14]. All the previous part-based fine-grained approaches, *e.g.*, [7, 8, 10, 11], directly used the deep convolutional descriptors and encoded them into a single representation, without evaluating the usefulness of the obtained object/part deep descriptors. By using powerful convolutional neural networks [15], we may not need to select useful dimensions inside feature vectors, as what we do for hand-crafted features [16, 17]. However, since most deep descriptors are not useful or meaningful for fine-grained recognition, it is necessary to select useful deep convolutional descriptors. Recently, selecting deep descriptors sheds its light on the fine-grained image retrieval task [18]. Moreover, it is also beneficial to fine-grained image recognition.

In this paper, by developing a novel deep part detection and descriptor selection scheme, we propose an end-to-end Mask-CNN (M-CNN) model which discards the fully connected layers for fine-grained bird species categorization. We only require the part annotations and image-level labels during the training time. In M-CNN, given the part annotations, we firstly separate them into two point sets. One set corresponds to the head part of the fine-grained bird image, and the other is for the torso. Then, the smallest rectangles that cover each point set are returned as the ground-truth mask, as shown in Fig. 1. The other pixels are background. By treating part localization as a three-class segmentation task, we leverage fully convolutional networks (FCN) [19] to generate weighted masks in the testing time for both localizing parts and selecting useful deep descriptors, *which does not use any annotation during testing*. After getting these two part masks, the segmentation class scores are treated as the automatically learned weights for aggregating descriptors. Meanwhile, we combine these two part masks to form the weighted object mask. Based on these

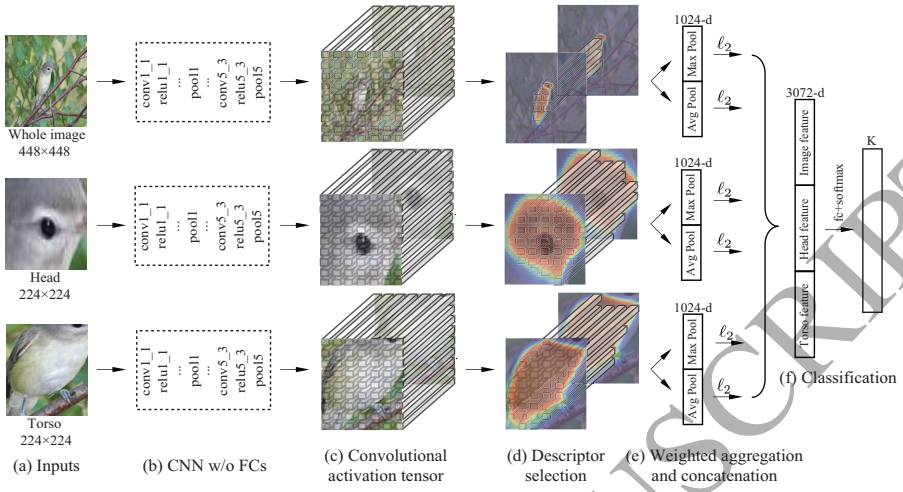


Figure 2: Architecture of the proposed three-stream Mask-CNN model. The three streams correspond to the whole image, head and torso image patches, respectively. In each stream, we employ the learned part/object masks to select the useful deep descriptors, and then aggregate these selected descriptors by weights (presented by different colors in Fig. 2d) to form the final image representation. As shown, thanks to the descriptor selection scheme, a large number of descriptors corresponding to background can be discarded by M-CNN, which is beneficial to fine-grained recognition. (This figure is best viewed in color.)

object/part masks, a three-stream Mask-CNN (image, head, torso) is built for joint training and aggregating the object-level and part-level cues simultaneously. The architecture of the proposed three-stream M-CNN is shown in Fig. 2. In each stream of M-CNN, we discard the fully connected layers. In the last convolutional layer, an input image is represented by multiple deep descriptors. In order to select useful descriptors to keep only those corresponding to the object, the pre-learned object/part masks by FCN are used. After that, the selected descriptors of each stream are both average and max pooled into two 512-d feature vectors, respectively. The standard ℓ_2-normalization is followed. Finally, the feature vectors of these three streams are concatenated, and then a classification (fc+softmax) layer is added for end-to-end joint training.

We validate the proposed three-stream M-CNN on two benchmark fine-grained image recognition datasets, *i.e.*, the Caltech-UCSD Birds (CUB) 200-2011 [1] and *Birdsnap* [20] dataset. On *CUB200-2011*, we achieved 85.7% classification accuracy based on VGG models [21] and 87.3% on Residual Nets [22]. On *Birdsnap*, our proposed M-CNN obtained 77.3% accuracy based on VGG [21] and 80.2% based on Residual Nets [22]. The classification accuracy of our M-CNN is new state-of-the-art on both two fine-grained datasets. Moreover, we also get accurate part localization (cf. Sec. 4.3). The key advantages and major contributions of the proposed M-CNN model are:

- To the best of our knowledge, Mask-CNN is the first model that selects

deep convolutional descriptors for object recognition, especially for fine-grained image recognition.

- We present a novel and efficient part-based three-stream model for fine-grained recognition. By discarding the fully connected layers, the proposed M-CNN is computationally efficient (cf. Table 1 and Table 4). Additionally, comparing with state-of-the-art methods, M-CNN has smaller feature dimensionality. Beyond those, it achieves the highest classification accuracy on *CUB200-2011* and *Birdsnap* among published methods.[2]

- The part localization performance of the proposed model outperforms other part-based fine-grained approaches which requires additional bounding boxes. In particular, M-CNN is 12.76% higher than state-of-the-art for head localization on *CUB200-2011*.

The rest of the paper is organized as follows. Section 2 summarizes related work. The proposed Mask-CNN model including the object/part masks learning procedure and the classification training process is described in Section 3. Detailed performance studies and analyses are conducted in Section 4. Section 5 concludes the paper.

2. Related work

In this section, we first review fine-grained image recognition, and then, give a brief recap about the researches of deep descriptor selection.

2.1. Fine-grained image recognition

Fine-grained recognition is a challenging problem and has recently emerged as a hot topic [3, 23, 5, 24, 25, 1]. During the past few years, a number of effective fine-grained recognition methods have been developed in the literature [7, 26, 8, 27, 28, 10, 11, 29, 12, 13]. We can roughly categorize these methods into three groups. The first group, *e.g.*, [26, 27], attempted to learn a more discriminative feature representation by developing powerful deep models for classifying fine-grained images. The second group aligned the objects in fine-grained images to eliminate pose variations and the influence of camera position, *e.g.*, [30, 31, 8]. The last group focused on part-based representations, because it is widely acknowledged that the subtle difference between fine-grained images mostly resides in the unique properties of object parts.

For the part-based fine-grained recognition methods, [32, 8, 10] used both bounding boxes of the birds and part annotations during training to learn an accurate part localization model. Then, based on these detected parts, different CNNs are fine-tuned using the detected parts separately. To ensure satisfactory localization results, they even used bounding boxes in the testing phase. In

[2] In this comparison, we do not consider methods that use large amounts of external images collected from the web.

contrast, our method only need part annotations for training, and do not need any supervision during testing. Moreover, our three-stream M-CNN is a unified framework for capturing object- and part-level information simultaneously.

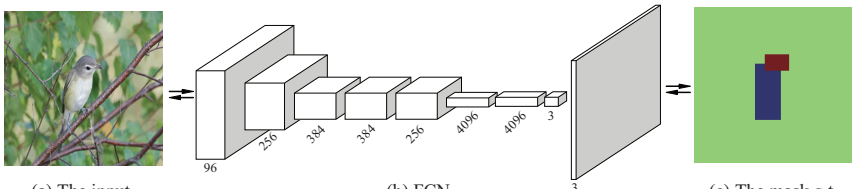
Some other part-based methods considered a weakly supervised setting, in which they categorize fine-grained images with only image-level labels, *e.g.*, [33, 34, 35, 11]. As will be shown by our experiments, classification accuracy of M-CNN is significantly higher than these weakly supervised methods. Meanwhile, M-CNN discards the parameter redundant fully connected layers, which makes it efficient to train/inference. Besides, the dimensionality of image representations in M-CNN is quite low, cf. Table 1. Therefore, M-CNN can be scalable to large-scale fine-grained datasets.

Moreover, these part-based methods, *e.g.*, [33, 34, 10, 35, 11, 36], usually require to firstly produce object/part proposals by selective search [37]. By comparing with that, the proposed M-CNN is more concise, which can accurately localize fine-grained parts *without utilizing bounding boxes and redundant object proposals*.

In addition, there are also fine-grained recognition methods based on segmentation, *e.g.*, [7, 38]. The most significant difference between them and M-CNN is: these methods only use segmentation to localize the whole object [38] or parts [7], while we further select useful deep convolutional descriptors using the masks obtained from segmentation. Among them, the part-stacked CNN model [7] is the most related work to ours. In [7], part-stacked CNN requires both bounding boxes and part annotations in training, and even needed the bounding boxes during testing. Within the image patch cropped using the bounding box, [7] treated the image crop around each of the fifteen part key points as 15 segmentation foreground classes, and used FCN to solve the 16-classes segmentation task. After obtaining the trained FCN, it localized these part point positions in the last convolutional layer. Then, deep activations corresponding to the fifteen parts and the whole object were stacked together. Fully connected layers were used for classification. Comparing with part-stacked CNN, M-CNN only needs to localize two main parts (head and torso), which makes the segmentation problem much easier and more accurate. M-CNN achieves high localization accuracy, as will be shown in Table 3. Meanwhile, as demonstrated in [7], using all the fifteen part activations cannot lead to better classification accuracy. Besides, M-CNN's accuracy on *CUB200-2011* is 2.0% higher than that of [7] using the same baseline network, although we use less annotations in training and do not use any annotation in testing. More detailed empirical comparisons can be found in Sec. 4.2.

2.2. Deep descriptor selection

As aforementioned, in the deep learning scenario, we might no longer need to select useful dimensions inside the learnt deep features. While, useful deep descriptors are necessary to be selected and noisy descriptors should be discarded, especially for fine-grained images. The so called "descriptor" here indicates the d-dimensional component vector of activations in a convolutional layer.



(a) The input (b) FCN (c) The mask g.t.

Figure 3: Demonstration of the mask learning procedure by fully convolutional network (FCN) [19]. (Best viewed if in color.)

In the line of deep descriptor selection for fine-grained images, SCDA [18] (Selective Convolutional Descriptor Aggregation) was proposed recently for dealing with the fine-grained image *retrieval* problem. In SCDA, it employs pre-trained models to first localize the main object in fine-grained images unsupervisedly. Then, based on the results of localization, it treats these deep descriptors corresponding to the object localization as the useful descriptors, and regards the others as background and noises. Thanks to the descriptor selection scheme, SCDA achieves the best retrieval performance in the content-based fine-grained image retrieval task. Comparing with SCDA, our proposed method can not only localize whole objects, but also localize fine-grained parts in a supervised manner, which achieves much more accurate part localization performance. Besides, our M-CNN is *the first work* to demonstrate that deep descriptor selection is beneficial to fine-grained image *recognition*.

3. The Mask-CNN model

In this section, we present the proposed three-stream Mask-CNN (M-CNN) model. Firstly, we adopt a fully convolutional network (FCN) [19] to generate the object/part masks for locating object/parts, and more importantly selecting deep descriptors. Then, based on these masks, the three-stream M-CNN is built for joint training and capturing both object- and part-level information.

3.1. Learning object and part masks

The fully convolutional network (FCN) [19] is designed for pixel-wise labeling. FCN can take an input image with any resolution and produce an output of the same size. In our method, we use FCN to not only localize the object and parts in fine-grained images, but also treat the segmentation predictions as the object and parts masks for the later descriptor selection process.

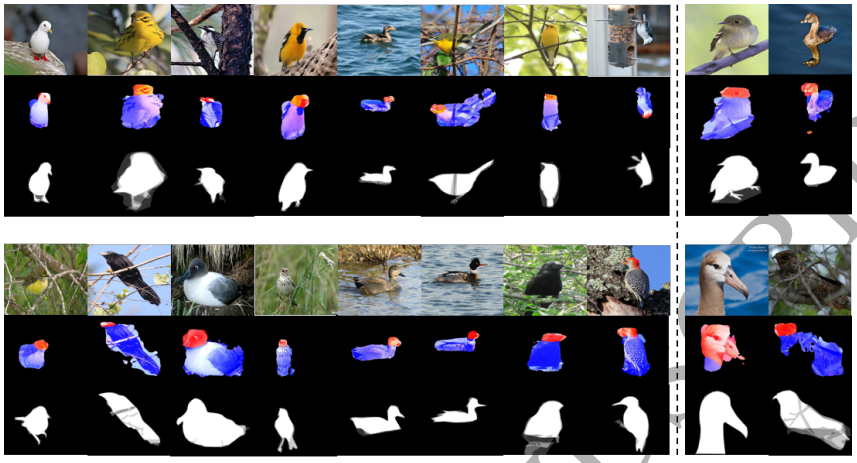
Each fine-grained image in the *CUB200-2011* [1] and *Birdsnap* [20] dataset is equipped with part annotations. *CUB200-2011* has fifteen part key points for each image, and *Birdsnap* has seventeen part key points for each. While, the other fine-grained image datasets (*e.g.*, [23, 5, 4]) have no such part annotations. As shown in Fig. 1, we split these key points into two sets, including the head key points (*i.e.*, the beak, forehead, crown, left eye, right eye, nape and throat for *CUB200-2011*; the beak, forehead, crown, left eye, right eye, left cheek,

right cheek, nape and throat for *Birdsnap*) and torso key points (*i.e.*, the back, breast, belly, left leg, right leg, left wing, nape, right wing, tail and throat for both *CUB200-2011* and *Birdsnap*). Based on the key points, two ground-truth of part masks are generated. One is the *head mask*, which corresponds to the smallest rectangle covering all the head key points. The other is the *torso mask*, which is the smallest rectangle covering all the torso key points. The overlapping part of the two rectangles is regarded as the head mask. As shown in Fig. 1b, the red rectangle is the head mask, and the blue one is for torso. The rest of the image is background. Similar to [39, 40], these bounding-box-like part masks are treated as the segmentation ground-truth. Thus, we model the part mask learning procedure as a three-class segmentation problem. For effective training, all the training and testing fine-grained images remain at their original resolutions. Then, we crop a 384×384 image patch in the middle of the original image as the inputs to FCN. The mask learning network architecture is shown in Fig. 3. In our experiments, we adopted FCN-8s [19] for learning and predicting part masks.

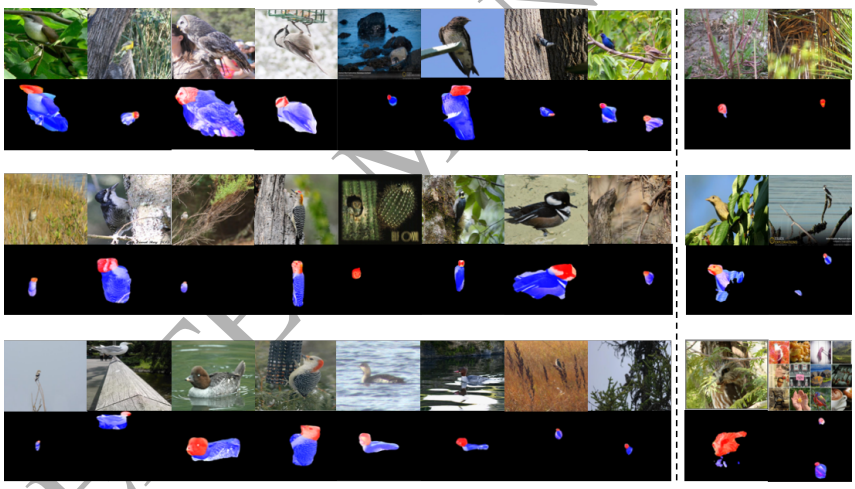
During the FCN inference, without using any annotation, three class heat maps (in the same size as the original input image) are returned for every image. Moreover, the predicted segmentation class scores are regarded as the learned part weights for the later descriptors aggregation process. We randomly choose some qualitative examples of the predicted part masks, and show them in Fig. 4. In these figures, the learned masks are overlaid onto the original images. The head part is highlighted in red, and the torso is in blue. The predicted background pixels are in black. As can be seen from these figures, even though the ground-truth part masks are not very accurate, the learned FCN model is able to return more accurate part masks. Meanwhile, these part masks can also localize the part positions by finding their enclosing rectangles. Moreover, comparing with the segmentation ground-truth (in the third row of Fig. 4), head masks combining with torso masks can be generally able to segment the foreground object well, even though no post-processing (*e.g.*, conditional random fields [41, 42]) is used. Quantitative results of part localization and object segmentation will be reported in Sec. 4.3 and Sec. 4.4, respectively.

Also, there are several failure cases, *e.g.*, the figures shown in the right side of Fig. 4. In some cases, it will treat the branch as the bird's torso. Some ones will also detect the head's and torso's reflections in water. In other cases, due to the scale of the main object or the complicated background, the torso masks can not be intactly predicted.

For the final recognition performance, both part masks, if accurately predicted, will benefit the later deep descriptor selection process and the final fine-grained classification. Therefore, during both the training and testing phases, we will use the predicted masks for both part localization and descriptor selection in M-CNN. We also combine the two masks to form a mask for the whole object, which is called the *object mask*.



(a) Predicted masks on the *CUB200-2011* dataset.



(b) Predicted masks on the *Birdsnap* dataset.

Figure 4: Random samples of successfully predicted part masks (on the left side) and four failure cases (on the right side) from the testing set on the *CUB200-2011* [1] and *Birdsnap* [20] dataset, respectively. The first row of each subfigure contains input fine-grained images. The second row are the part masks predictions. In these figures, we overlay the part mask predicted by FCN (the head highlighted in red and the torso in blue) onto the original images. The pixels predicted as background are in black. The third row in (a) is the corresponding segmentation ground-truth provided in the *CUB200-2011* dataset. The *Birdsnap* dataset does not supply the segmentation ground-truth for its fine-grained images. (The figures are best viewed in color.)

3.2. Training Mask-CNN

After obtaining the object and part masks, we build the three-stream M-CNN for joint training. The overall architecture of the proposed model is presented in Fig. 2. We take the whole image stream as an example to illustrate the pipeline of each stream.

The inputs of the whole image stream are the original images resized to $h \times h$. In our experiments, we report the results for $h = 224$ and $h = 448$, respectively. The input images are fed into a traditional convolutional neural network, but the fully connected layers are discarded. That is to say, the CNN model used in our proposed M-CNN only contains convolutional, ReLU and pooling layers, which greatly brings down the M-CNN model size. Specifically, we use VGG-16 [21] as the baseline model, and the layers before pool_5 are kept (including pool_5). We obtain a $7 \times 7 \times 512$ activation tensor in pool_5 if the input image is 224×224. Therefore, we have 49 deep convolutional descriptors of 512-d, which also correspond to 7×7 spatial positions in the input images. Then, the learned object mask (cf. Sec. 3.1) is firstly resized to 7×7 by the bilinear interpolation, and then used for selecting useful and meaningful deep descriptors.

As illustrated in Fig. 2c and Fig. 2d, the descriptor should be kept by weights when it locates in the object region. If it locates in the background region, that descriptor will be discarded. In our implementation, the mask contains the learned part/object segmentation scores, which is a real matrix whose elements are in the range of $[0, 1]$. Correspondingly, 1 stands for absolutely keeping and 0 is for absolutely discarding. We implement the selection process as an element-wise product operation between the convolutional activation tensor and the mask matrix. Therefore, the descriptors located in the object region will remain by weights, while the other descriptors will become zero vectors. Concretely, if the pixels are predicted as head/torso by FCN, the real values of the mask are kept. Otherwise, if the pixels indicate the regions are background, the value of these background regions in the mask are reset to the zero value. Then, the processed masks are used for selecting descriptors and the rest processing.

For these selected descriptors, in the end-to-end M-CNN learning process, we both average and max pool them into two 512-d feature vectors, respectively. Then, the ℓ_2-normalization is followed for each of them. After that, we concatenate them into an 1024-d feature as the final representation of the whole image stream.

The streams for head and torso have similar processing steps as the whole image one. However, different from the inputs of the whole image stream, we generate the input images of the head and torso streams as follows. After obtaining the two part masks, we use the part masks as the part detectors to localize the head part and torso part in the input images. For each part, we return the smallest rectangle bounding box which contains the part mask regions. Based on the rectangle bounding box, we crop the image patch which acts as the inputs of the part stream. The last two streams of Fig. 2 show the head and torso streams in M-CNN. The inputs of these two streams are all resized into 224×224 in our experiments.

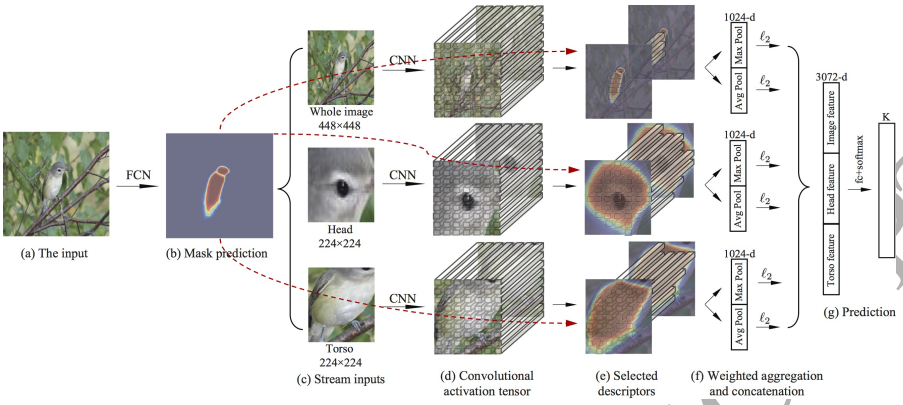


Figure 5: Testing stage of Mask-CNN. (Best viewed if in color.)

In the classification step shown in Fig. 2f, the final 3,072-d image representation is the concatenation of the whole image, the head and the torso features. The last layer of M-CNN is a 200-way classification (fc+softmax) layer for recognition on *CUB200-2011* and a 500-way classification layer on *Birdsnap*, respectively. The three-stream M-CNN is learned end-to-end, with the parameters of three CNNs learned simultaneously. During training M-CNN, the parameters of the learned FCN segmentation network are fixed.

3.3. Testing stage of Mask-CNN

During inference, when facing with a testing image, the learned FCN model firstly returns the corresponding mask predictions for both head and torso. Then, based on the masks, we use them as the part detectors to localize the head part and torso part in the input images. The extracted head and torso image patches are regarded as the inputs for the head and torso streams in Mask-CNN. After obtaining the convolutional descriptors through the convolution layers of three-stream Mask-CNN, the predicted masks are employed again. While, at this time, the masks are utilized for selecting descriptors (cf. Fig. 5(e)). At last, the selected descriptors are aggregated following the strategy in the training stage, and then we can get the predicted label based on the 3,072-d final image representation. The whole testing stage of Mask-CNN is shown in Fig. 5.

4. Experiments

In this section, we firstly describe the experimental settings and implementation details. Then, we report the classification accuracy. The performance of part localization and object segmentation will also be provided. Finally, we present some discussions about the proposed M-CNN model.

4.1. Dataset and implementation details

Following the other published part-based fine-grained methods [7, 8, 10], we perform the empirical evaluation on the widely-used fine-grained benchmark Caltech-UCSD 2011 bird dataset [1] and Birdsnap [20] dataset. The *CUB200-2011* dataset contains 200 bird categories, and each category has roughly 30 training images. Totally, there are 5,994 training images and 5,794 test images in that dataset. For the *Birdsnap* dataset, it contains 500 North American bird species whose images having 47,386 images for training and 2,443 images for test, which is much larger and challenging than *CUB200-2011*. We follow the training and testing splitting included with these two datasets. In the training phase, the fifteen part annotations of each dataset are adopted for generating the part masks' ground-truth, and meanwhile the image-level labels are used for the end-to-end M-CNN joint training. We need no supervision signals (*e.g.*, part annotations or bounding boxes) when testing.

The proposed Mask-CNN model and FCN used for generating masks are implemented using the open-source library MatConvNet [43]. In our experiments, after getting the learned part masks, we firstly generate the image patches of birds' head and torso as described in Sec. 3.2. Then, to facilitate the convergence of three-stream CNNs, each single stream corresponding to the whole image, head and torso is fine-tuned on its input images separately. The CNNs used in each stream is initialized by the popular VGG-16 model [21] pre-trained on ImageNet. The loss function in each stream is the popular used cross-entropy loss function. For fair comparisons with other methods (*e.g.*, [7, 8, 10]), we also implement our three-stream M-CNN model based on the Alex-Net model [15]. In addition, we double the training data by a horizontal flipping for all the three streams. After fine-tuning on each stream, as shown in Fig. 2, the joint training of three-stream M-CNN is performed. Dropout is not used in M-CNN. At the test time, we average the predictions of the image and its flipped copy, and output the class with the highest score as the prediction for a test image. In addition, directly using the softmax predictions results is a slight drop in accuracy compared to logistic regression (LR), which is consistent with the observations in [27]. Therefore, in the following, the reported results of M-CNN are all achieved by one-vs-all logistic regression [44] on the extracted features of three-stream M-CNNs with the default hyper-parameter $C_{\mathrm{LR}} = 1$. Upon acceptance, we will release our source code and trained models, so that all results in the paper can be reproduced. All the experiments are run on a computer with Intel Xeon E5-2660 v3, 64G main memory, and an Nvidia Tesla K40 GPU.

4.2. Comparisons with state-of-the-art methods

In this section, we compare our proposed M-CNN with state-of-the-arts on *CUB200-2011* and *Birdsnap*, respectively.

4.2.1. Results on CUB200-2011

We report the classification accuracy on the *CUB200-2011* dataset of the proposed three-stream M-CNN model, and compare with the baseline methods

and state-of-the-art methods in the literature. The classification results are presented in Table 1. For fair comparison, we only report the results when they do not use part annotations in testing.

At first, the input images of the three streams are all resized to 224×224. The classification accuracy of M-CNN is 84.2%. Following [7, 27], we change the input images of the whole image stream to 448×448 pixels. It improves the classification performance by 1.5%, which achieves the best classification accuracy 85.7% on *CUB200-2011*. Moreover, when using the Residual Net-50 [22] architecture, three-stream M-CNN obtains 87.3% accuracy.

For comparisons of the final classification accuracy on *CUB200-2011*, since there is no previous work in the same experimental setting (*i.e.*, only using the part annotation in training) as ours, we divide the previous work into two kinds of fine-grained methods: the first one are the methods using the part annotations (*e.g.*, [10, 7, 8, 30]), and the second one are the methods using only image-level labels (*e.g.*, [45, 38, 46, 34, 11, 33, 27, 26, 35]).

On one hand, comparing with the methods using the part annotations, part-stacked CNN [7] was one of state-of-the-arts, which is a strong baseline of Mask-CNN. Specifically, because part-stacked CNN used the Alex-Net model [15], we also build another three-stream M-CNN based on Alex-Net. The accuracy of our three-stream M-CNN (Alex-Net) is 78.6%. It is 2.0% higher than that of [7]. Meanwhile, the inference speed of our three-stream M-CNN (Alex-Net) is 33.9 FPS, which is much faster than 20 FPS reported in [7]. Moreover, in the Alex-Net based three-stream M-CNN, the final feature vector is only 1,536-dimensional.

On the other hand, for the methods using only image-level labels, such as PDFS [35], Spatial Transformer CNN [26] and Bilinear [27], they are two outstanding fine-grained methods using only the image-level supervisions. The classification accuracy of our Mask-CNN is 1.2% and 1.6% higher than PDFS and STCNN, respectively. It also validates the effectiveness of our proposed method. Moreover, the image representation of M-CNN has lower feature dimensions than that of Bilinear [27] and PDFS [35].

4.2.2. Results on Birdsnap

The classification accuracy on *Birdsnap* is reported in Table 2. The input images of the whole image stream are of the 448×448 image resolution. The input images of the other streams are of 224×224. Comparing with the previous methods conducted on *Birdsnap*, our proposed M-CNN outperforms them by a large margin. Meanwhile, the small dimensionality bring it scalability and efficiency in large-scale datasets.

4.3. Part localization results

In addition to the qualitative part localization results shown in Sec. 3.1, in this section, we quantitatively assess the localization correctness using the Percentage of Correctly Localized Parts (PCP) metric.

As reported in Table 3, the metrics of *CUB200-2011* are the percentage of parts (*i.e.*, the head and torso) that are correctly localized with a >50% IOU

Table 1: Comparison of classification accuracy on *CUB200-2011* with state-of-the-art methods. Note that, "Model" describes the deep models used in these methods. The inference speeds (frames/sec) of M-CNNs on a K40 GPU are also reported.

Method	Train phase		Test phase		Model	Dim.	Acc.
	BBox	Parts	BBox	Parts			
PB R-CNN with BBox [10]	✓	✓	✓		Alex-Net×3	12,288	76.4%
Part-Stacked CNN [7]	✓	✓	✓		Part-Stacked CNN×1	4,096	76.6%
Deep LAC [8]	✓	✓	✓		Alex-Net×3	12,288	80.3%
PB R-CNN [10]	✓	✓			Alex-Net×3	12,288	73.9%
Pose Normalized CNNs [30]	✓	✓			Alex-Net×3	13,512	75.7%
MixDCNN [45]	✓				GoogLeNet×1	6,144	81.1%
Co-Segmentation [38]	✓				VGG-19×2	126,976	82.0%
Multi-grained [46]	✓				VGG-19×3	12,288	83.0%
Two-Level [34]					VGG-16×1	16,384	77.9%
Weakly supervised FG [11]					VGG-16×1	262,144	79.3%
Constellations [33]					VGG-19×1	208,896	81.0%
Multi-grained [46]					VGG-19×3	12,288	81.7%
Bilinear [27]					VGG-16 and VGG-M	262,144	84.1%
Spatial Transformer CNN [26]					ST-CNN (inception)×4	4,096	84.1%
PDFS [35]					VGG-16×1	69,632	84.5%
Our 3-stream M-CNN (Alex-Net)[1]		✓			Alex-Net (w/o FCs)×3	**1,536**	**78.6%**
Our 3-stream M-CNN (VGG-16)[2]		✓			VGG-16 (w/o FCs)×3	**3,072**	**85.7%**
Our 3-stream M-CNN (ResNet-50)[3]		✓			ResNet-50×3	**12,288**	**87.3%**

[1]**The inference speed of M-CNN (Alex-Net) is 33.9 frames/sec.**
[2]**The inference speed of M-CNN (VGG-16) is 8.3 frames/sec.**
[3]**The inference speed of M-CNN (ResNet) is 11.8 frames/sec.**
(All the speeds here contain both part masks prediction and the final classification.)

Table 2: Comparison of classification accuracy on *Birdsnap* with state-of-the-art methods. Note that, "Model" describes the deep models used in these methods. We do not list the inference speeds on this dataset, because the inference speeds on *Birdsnap* is similar to the speeds on *CUB200-2011*.

Method	Train phase		Test phase		Model	Dim.	Acc.
	BBox	Parts	BBox	Parts			
MixDCNN [45]	✓				GoogLeNet×1	6,144	74.1%
Multi-grained [46]	✓				VGG-19×3	12,288	74.8%
Multi-grained [46]					VGG-19×3	12,288	65.9%
Our 3-stream M-CNN (Alex-Net)		✓			Alex-Net (w/o FCs)×3	**1,536**	**64.8%**
Our 3-stream M-CNN (VGG-16)		✓			VGG-16 (w/o FCs)×3	**3,072**	**77.3%**
Our 3-stream M-CNN (ResNet-50)		✓			ResNet-50×3	**12,288**	**80.2%**

with the ground-truth part bounding boxes as generated in [8, 10]. Comparing the results of PCP for torso, our method (no matter based on VGG-16 or Alex-Net) outperforms part-based R-CNN [10] and strong DPM [32] by a large margin. However, because we do not use any annotation in testing, the localization performance is lower than the one of Deep LAC [8] which used the bounding boxes during testing. In addition, for the head localization task which is more challenging than the torso one, even though our method just uses part annotations in training, the head localization performance (86.76% for VGG-16 based, and 81.22% for Alex-Net based) is still significantly higher than the other methods.

Additionally, the head and torso localization accuracy on *Birdsnap* of our method are 67.40% and 78.87% based on Alex-Net, and 74.51% and 84.45% based on VGG-16, respectively. Since there is no previous results on part local-

Table 3: Comparison of part localization performance on the *CUB200-2011* dataset.

Method	Train phase		Test phase		Head	Torso
	BBox	Parts	BBox	Parts		
Strong DPM [32]	✓	✓	✓		43.49%	75.15%
Part-based R-CNN with BBox [10]	✓	✓	✓		68.19%	79.82%
Deep LAC [8]	✓	✓	✓		74.00%	**96.00%**
Part-based R-CNN [10]	✓	✓			61.42%	70.68%
Ours (Alex-Net based FCN)		✓			81.22%	91.72%
Ours (VGG-16 based FCN)		✓			**86.76%**	91.87%

ization on *Birdsnap* in the literature, we further conducted experiments using the released source codes of Part-based R-CNN [10] for comparisons. When utilizing part annotations and bounding boxes of the *Birdsnap* dataset, the part localization accuracy of Part-based R-CNN are 49.97% and 74.86% for head and torso, respectively. Particularly, the head localization accuracy of our method (no matter based on Alex-Net or VGG-16) is significantly higher than the accuracy of Part-based R-CNN. The observations of *Birdsnap* are consistent with the localization results of *CUB200-2011*.

4.4. Object segmentation performance

Because the *CUB200-2011* dataset also supplies the object segmentation ground-truth, we can directly test the learned object masks on the segmentation metric. The figures in the second row of Fig. 4 show qualitative segmentation results. Our method based on FCN is generally able to segment the foreground object well, but understandably has trouble to segment the birds' finer details, *e.g.*, claws and beak. Since our goal is not to segment objects, we do not perform any refinement as pre-processing or post-processing. We evaluate the segmentation performance quantitatively by the common semantic segmentation metric mean IU (region intersection over union) between the ground truth foreground object and the predicted object masks. It is 74.59% on the testing set. In fact, a better segmentation result will lead to better predicted object/part masks, and also benefit the final classification. To further improve the classification accuracy, some pre-processing methods, *e.g.*, GrabCut [47], are worth trying to obtain better mask ground-truth than the rectangles in Fig. 3c.

4.5. Ablation and diagnostic experiments

In this section, we conduct ablation experiments on both the *CUB200-2011* and *Birdsnap* dataset, and present discussions of the proposed three-stream M-CNN model. The experimental results are all based on M-CNN with the VGG-16 architecture.

4.5.1. Is descriptor selection effective?

In order to clearly validate the effectiveness of the descriptor selection process in M-CNN, we perform two baseline methods which are also based on the proposed three-stream architecture. Different from our M-CNN, these two baseline methods do not contain the descriptor selection part, *i.e.*, the processing shown in Fig. 2d.

Table 4: Comparison with the baseline methods on *CUB200-2011* and *Birdsnap*. For all the models, the inputs of the whole image stream are 224×224 for fair comparisons. The inference speed contains both part masks predictions and the final classification process.

Model	3-stream FCs	3-stream Pooling	The proposed 3-stream M-CNN
Descriptor selection	✗	✗	✓
Accuracy on *CUB200-2011*	81.4%	82.8%	**84.2%**
Accuracy on *Birdsnap*	73.0%	74.3%	**76.0%**
Inference speed	9.2 FPS	**13.0 FPS**	12.9 FPS

The first baseline method employ the traditional fully connected layers to conduct classification for each stream, which is called "3-stream FCs". In "3-stream FCs", we replace the (b) to (e) parts of each stream in Fig. 2 with a CNN containing fully connected layers (*i.e.*, VGG-16 with only fc_8 removed). Thus, the generated feature in the last layer of each stream is a 4,096-d single vector. The rest procedure is also to concatenate the three 4,096-d features into the final one with 12,288-d, and to learn a 200-way (or 500-way) classification (fc+softmax) layer on the 12,288-d image representation.

The second baseline is similar to the proposed M-CNN. The most prominent difference is that it discards the descriptor selection part, *i.e.*, the processing in Fig. 2d. Thus, the convolutional deep descriptors of pool_5 in each stream are directly average and max pooled, and then ℓ_2-normalized, respectively. Therefore, we call it the "3-stream Pooling". The remaining procedures are the same as the proposed M-CNN.

Table 4 presents the comparison of classification accuracy and inference speed on the *CUB200-2011* and *Birdsnap* dataset. Because CNN models with fully connected layers require the inputs of 224×224, the input images of these three compared methods in Table 4 are all 224×224. In that table, the proposed M-CNN achieves the best classification accuracy rate. Due to the missing of descriptor selection, "3-stream Pooling" is about 1.4% lower than M-CNN on *CUB200-2011* and 1.7% on *Birdsnap*. The "3-stream FCs" baseline method has the lowest accuracy. Its lower accuracy might be caused by the fully connected layers, which may have caused overfitting.

In addition, the feature extraction speeds (frames/sec) on a Tesla K40 GPU for these methods using our MatConvNet based implementation are shown on the bottom of Table 4. The speeds are conducted on the *CUB200-2011* dataset, and the *Birdsnap* dataset has the similar inference speeds. In addition, please note that, for these streams models, the speeds are the serial computing speeds. That is to say, a GPU is used for inference one stream by one stream (whole image→head→torso). The inference speed of the proposed M-CNN is almost the same as the "3-stream Pooling" baseline model without selecting descriptor, and is 3.7 FPS faster than the baseline with fully connected layers. Besides, as the input images in the whole image stream are of 224×224, the inference speed (12.9 FPS) is faster than the one whose inputs are 448×448 (8.3 FPS).

For further validating the effectiveness of selecting descriptors, we also change the inputs of the whole image stream in "3-stream Pooling" to 448×448, which can get 84.5% on *CUB200-2011* (76.0% on *Birdsnap*). It is still 1.2% (1.3%)

Table 5: Comparison of M-CNN with different streams on the *CUB200-2011* and *Birdsnap* dataset.

Dataset	Stream			Accuracy
	Image	Head	Torso	
CUB200-2011	✓			80.5%
	✓	✓		84.2%
	✓		✓	82.1%
	✓	✓	✓	**85.7%**
Birdsnap	✓			72.7%
	✓	✓		76.2%
	✓		✓	73.6%
	✓	✓	✓	**77.3%**

lower than the classification accuracy of our three-stream M-CNN (cf. 85.7% in Table 1 and 77.3% in Table 2).

4.5.2. How important are different streams?

We here investigate what different streams contribute to the final recognition performance. Table 5 reports the classification accuracy of M-CNN containing different streams. When it only has the whole image stream, on *CUB200-2011*, the accuracy is 80.5%. By incorporating the head and torso stream, after joint training, the accuracy increases to 85.7% until containing both two part streams. From that table, we can find the head stream could be more important/discriminative than the torso stream. After incorporating the head stream, the original whole image stream can improve 3.7% (80.5%→84.2%). However, incorporating the torso stream, it just increases 1.6% accuracy (80.5%→82.1%). Additionally, comparing with the results of two-stream (*i.e.*, the second and third row) and three-stream (*i.e.*, the last row) in Table 5 separately, we can see that: adding the torso stream improves the accuracy by 1.5% (84.2%→85.7%), while adding the head one can improve it by 3.6% (82.1%→85.7%). Besides, similar observations can be found on *Birdsnap*.

4.5.3. Are different pooling strategies necessary?

In M-CNN, we propose to concatenate both the average- and max-pooled features in each stream as the final representations. In the following, the diagnostic experiments on different pooling strategies are presented. As shown in Table 6, on *CUB200-2011*, the proposed M-CNN (average-pooling+max-pooling, 85.7%) outperforms the ones with only average-pooled features (85.1%) or only max-pooled features (85.4%). For *Birdsnap*, different pooling ensemble can also improve the classification accuracy up to 77.3%. Therefore, different pooling strategies used in M-CNN is necessary for the final classification accuracy.

4.5.4. Can M-CNN share the previous layers between different streams?

Because the early layers of CNNs usually correspond to low-level visual atoms (*e.g.*, orientated edges, bars or blobs) [48], we attempt to combine the first ten layers (from "conv_{1_1} with relu_{1_1}" to "conv_{2_1}, relu_{2_1} with pool_2") in VGG-16 as layer sharing, and jointly train the new three-stream M-CNN. However, the accuracy of the layer sharing M-CNN is only 82.1% on *CUB200-2011*

Table 6: Comparison of the three-stream M-CNN model with different pooling strategies on the *CUB200-2011* and *Birdsnap* dataset.

Accuracy	Pooling		Accuracy
	Ave.-pool	Max-pool	
CUB200-2011	✓		85.1%
		✓	85.4%
	✓	✓	**85.7%**
Birdsnap	✓		77.1%
		✓	76.9%
	✓	✓	**77.3%**

(73.6% on *Birdsnap*), which are significantly worse than 85.7% (77.3%) of the proposed M-CNN. But, the result of layer sharing justifies the separate design of the three-stream M-CNN's architecture.

5. Conclusion

In this paper, we presented the benefits of selecting deep convolutional descriptor in object recognition, especially fine-grained image recognition. By developing the descriptor selection scheme, we proposed a novel end-to-end Mask-CNN (M-CNN) model without the fully connected layers to not only accurately localize object/parts, but also generate weighted object/part masks for selecting deep convolutional descriptors. After aggregating the selected descriptors, the object-level and part-level cues were encoded by the proposed three-stream M-CNN model. Mask-CNN not only achieved a new state-of-the-art bird species classification accuracy on *CUB200-2011* and *Birdsnap*, but also had the lowest dimensional feature representations.

In the future, we plan to solve the part detection problem of M-CNN in the weakly supervised setting, in which we only require the image-level labels. Thus, it will require far less labeling effort to achieve comparable classification accuracy. In addition, another interesting direction is to explore the benefits of descriptor selection for generic object categorization.

Acknowledgement

This research was supported by the National Natural Science Foundation of China under Grant 61772256 and Grant 61422203, the Collaborative Innovation Center of Novel Software Technology and Industrialization. The authors would like to thank Qichang Hu, Jian-Hao Luo for reading the draft, and thank the anonymous reviewers, whose comments have helped improving this paper.

References

[1] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD birds-200-2011 dataset, Tech. Report CNS-TR-2011-001.

[2] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-UCSD birds 200, Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010).

[3] A. Angelova, S. Zhu, Y. Lin, Image segmentation for large-scale subcategory flower recognition, in: Proceedings of Applications of Computer Vision, Clearwater Beach, FL, Jan. 2013, pp. 39–45.

[4] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Proceedings of Indian Conference on Computer Vision, Graphics and Image Processing, Bhubaneswar, India, Dec. 2008, pp. 722–729.

[5] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, in: Proceedings of IEEE International Conference on Computer Vision Workshop on 3D Representation and Recognition, Sydney, Australia, Dec. 2013.

[6] T. Berg, P. Belhumeur, POOF: Part-based one-vs.-one features for fine-grained categorization, face verification and attribute estimation, in: Proceedings of IEEE International Conference on Computer Vision, Portland, Oregon, Jun. 2013, pp. 955–962.

[7] S. Huang, Z. Xu, D. Tao, Y. Zhang, Part-stacked CNN for fine-grained visual categorization, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, Jun. 2016, pp. 1173–1182.

[8] D. Lin, X. Shen, C. Lu, J. Jia, Deep LAC: Deep localization, alignment and classification for fine-grained recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, Jun. 2015, pp. 1666–1674.

[9] S. Maji, G. Shakhnarovich, Part and attribute discovery from relative annotations, International Journal of Computer Vision 108 (1-2) (2014) 82–96.

[10] N. Zhang, J. Donahue, R. Girshick, T. Darrell, Part-based R-CNNs for fine-grained category detection, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), European Conference on Computer Vision, Part I, LNCS 8689, Springer, Switzerland, Zürich, Switzerland, Sept. 2014, pp. 834–849.

[11] Y. Zhang, X.-S. Wei, J. Wu, J. Cai, J. Lu, V.-A. Nguyen, M. N. Do., Weakly supervised fine-grained categorization with part-based image representation, IEEE Transactions on Image Processing 25 (4) (2016) 1713–1725.

[12] G.-S. Xie, X.-Y. Zhang, W. Yang, M. Xu, S. Yan, C.-L. Liu, LG-CNN: From local parts to global discrimination for fine-grained recognition, Pattern Recognition 71 (2017) 118–131.

[13] S. H. Lee, C. S. Chan, S. J. Mayo, P. Remagnino, How deep learning extracts and learns leaf features for plant classification, Pattern Recognition 71 (2017) 1–13.

[14] M. D. Zeiler, G. W. Taylor, R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, in: Proceedings of IEEE International Conference on Computer Vision, Sydney, Australia, Dec. 2013, pp. 2018–2025.

[15] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, Lake Tahoe, NV, Dec. 2012, pp. 1097–1105.

[16] A. Eigenstetter, B. Ommer, Visual recognition using embedded feature selection for curvature self-similarity, in: Advances in Neural Information Processing Systems, Lake Tahoe, NV, Dec. 2012, pp. 377–385.

[17] Y. Zhang, J. Wu, J. Cai, Compact representation for image classification: To choose or to compress?, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, Jun. 2014, pp. 907–914.

[18] X.-S. Wei, J.-H. Luo, J. Wu, Z.-H. Zhou, Selective convolutional descriptor aggregation for fine-grained image retrieval, IEEE Transactions on Image Processing 26 (6) (2017) 2868–2881.

[19] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, Jun. 2015, pp. 3431–3440.

[20] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, P. N. Belhumeur, Birdsnap: Large-scale fine-grained visual categorization of birds, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, Jun. 2014, pp. 2019–2026.

[21] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of International Conference on Learning Representations, San Diego, CA, May. 2015, pp. 1–14.

[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, Jun. 2016, pp. 770–778.

[23] A. Khosla, N. Jayadevaprakash, B. Yao, L. Fei-Fei, Novel dataset for fine-grained image categorization, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop on Fine-Grained Visual Categorization, Colorado Springs, CO, Jun. 2011, pp. 806–813.

[24] O. M. Parkhi, A. Vedaldi, A. Zisserman, C. V. Jawahar, Cats and dogs, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, Jun. 2012, pp. 3498–3505.

[25] E. Rodner, M. Simon, G. Brehm, S. Pietsch, J. W. Wägele, J. Denzler, Fine-grained recognition datasets for biodiversity analysis, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop on Fine-grained Visual Classification, Boston, MA, Jun. 2015.

[26] M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, Spatial transformer networks, in: Advances in Neural Information Processing Systems, Montréal, Canada, Dec. 2015, pp. 2008–2016.

[27] T.-Y. Lin, A. RoyChowdhury, S. Maji, Bilinear CNN models for fine-grained visual recognition, in: Proceedings of IEEE International Conference on Computer Vision, Sandiago, Chile, Dec. 2015, pp. 1449–1457.

[28] S. Gao, I. W.-H. Tsang, Y. Ma, Learning category-specific dictionary and shared dictionary for fine-grained image categorization, IEEE Transactions on Image Processing 23 (2) (2014) 623–634.

[29] H. Yao, S. Zhang, Y. Zhang, J. Li, Q. Tian, Coarse-to-fine description for fine-grained visual categorization, IEEE Transactions on Image Processing 25 (10) (2016) 4858–4872.

[30] S. Branson, G. V. Horn, S. Belongie, P. Perona, Bird species categorization using pose normalized deep convolutional nets, in: British Machine Vision Conference, Nottingham, England, Sept. 2014, pp. 1–14.

[31] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, T. Tuytelaars, Local alignments for fine-grained categorization, International Journal of Computer Vision 111 (2) (2014) 191–212.

[32] H. Azizpour, I. Laptev, Object detection using strongly-supervised deformable part models, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), European Conference on Computer Vision, Part I, LNCS 7572, Springer, Heidelberg, Firenze, Italy, Oct. 2012, pp. 836–849.

[33] M. Simon, E. Rodner, Neural activation constellations: Unsupervised part model discovery with convolutional networks, in: Proceedings of IEEE International Conference on Computer Vision, Sandiago, Chile, Dec. 2015, pp. 1143–1151.

[34] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, Z. Zhang, The application of two-level attention models in deep convolutional neural network for fine-grained image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, Jun. 2015, pp. 842–850.

[35] X. Zhang, H. Xiong, W. Zhou, W. Lin, Q. Tian, Picking deep filter responses for fine-grained image recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, Jun. 2016, pp. 1134–1142.

[36] L. Zhang, Y. Yang, M. Wang, R. Hong, L. Nie, X. Li, Detecting densely distributed graph patterns for fine-grained image categorization, IEEE Transactions on Image Processing 25 (2) (2016) 553–565.

[37] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders, Selective search for object recognition, International Journal of Computer Vision 104 (2) (2013) 154–171.

[38] J. Krause, H. Jin, J. Yang, L. Fei-Fei, Fine-grained recognition without part annotations, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, Jun. 2015, pp. 5546–5555.

[39] J. Dai, K. He, J. Sun, Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation, in: Proceedings of IEEE International Conference on Computer Vision, Sandiago, Chile, Dec. 2015, pp. 1635–1643.

[40] G. Papandreou, L.-C. Chen, K. Murphy, A. Yuille, Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation, in: Proceedings of IEEE International Conference on Computer Vision, Sandiago, Chile, Dec. 2015, pp. 1742–1750.

[41] P. Krähenbühl, V. Koltun, Efficient inference in fully connected CRF with Guassian edge potentials, in: Advances in Neural Information Processing Systems, Granada, Spain, Dec. 2011, pp. 109–117.

[42] L. Ladicky, C. Russell, P. Kohli, P. H. Torr, Associate hierarchical crfs for object class image segmentation, in: Proceedings of IEEE International Conference on Computer Vision, Kyoto, Japan, Sept. 2009, pp. 739–746.

[43] A. Vedaldi, K. Lenc, MatConvNet – Convolutional Neural Networks for MATLAB, in: Proceeding of ACM International Conference on Multimedia, Brisbane, Australia, Oct. 2015, pp. 689–692, `http://www.vlfeat.org/matconvnet/`.

[44] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A library for large linear classification, Journal of Machine Learning Research 9 (2008) 1871–1874.

[45] Z. Y. Ge, A. Bewley, C. McCool, P. Corke, B. Upcroft, C. Sanderson, Fine-grained classification via mixture of deep convolutional neural networks, in: Proceedings of IEEE Winter Applications of Computer Vision, Lake Placid, NY, Mar. 2016, pp. 1–6.

[46] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, Z. Zhang, Multiple granularity descriptors for fine-grained categorization, in: Proceedings of IEEE International Conference on Computer Vision, Sandiago, Chile, Dec. 2015, pp. 2399–2406.

[47] C. Rother, V. Kolmogorov, A. Blake, GrabCut: Interactive foreground extraction using iterated graph cuts, ACM Transactions on Graphics 23 (2004) 309–314.

[48] M. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Neural Networks, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), European Conference on Computer Vision, Part I, LNCS 8689, Springer, Switzerland, Zürich, Switzerland, Sept. 2014, pp. 818–833.

Xiu-Shen Wei received the B.S. degree in computer science and technology in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Nanjing University, China. His research interests are computer vision and machine learning. He received the first prize in the Apparent Personality Analysis competition (in association with ECCV 2016) and the first RunnerUp Award from the Cultural Event Recognition Competition (in association with ICCV 2015) as the Team Director.

Chen-Wei Xie received his BS degree from Southeast University, China, in 2015. He is currently a postgraduate student in the Department of Computer Science and Technology, Nanjing University, China. His research interests include computer vision and machine learning.

Jianxin Wu received his BS and MS degrees in computer science from Nanjing University, and his PhD degree in computer science from the Georgia Institute of Technology. He is currently a professor in the Department of Computer Science and Technology at Nanjing University, China, and is associated with the National Key Laboratory for Novel Software Technology, China. He was an assistant professor in the Nanyang Technological University, Singapore, and has served as an area chair for CVPR 2017, ICCV 2015, senior PC member for AAAI 2017, AAAI 2016 and an associate editor for Pattern Recognition Journal. His research interests are computer vision and machine learning.

Chunhua Shen is a Professor at School of Computer Science, University of Adelaide. He is a Project Leader and Chief Investigator at the Australian Research Council Centre of Excellence for Robotic Vision (ACRV), for which he leads the project on machine learning for robotic vision. Before he moved to Adelaide as a Senior Lecturer, he was with the computer vision program at NICTA (National ICT Australia), Canberra Research Laboratory for about six years. His research interests are in the intersection of computer vision and statistical machine learning. He studied at Nanjing University, at Australian National University, and received his PhD degree from the University of Adelaide. From 2012 to 2016 he held an Australian Research Council Future Fellowship.