Part-Stacked CNN for Fine-Grained Visual Categorization

Shaoli Huang*[1], Zhe Xu*[1,2], Dacheng Tao[1], and Ya Zhang[2]

[1]Centre for Quantum Computation & Intelligent Systems and Faculty of Engineering and Information Technology, University of Technology Sydney, Australia

[2]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China

{shaoli.huang@student.,dacheng.tao@}uts.edu.au, {xz3030,ya_zhang}@sjtu.edu.cn

Abstract

In the context of fine-grained visual categorization, the ability to interpret models as human-understandable visual manuals is sometimes as important as achieving high classification accuracy. In this paper, we propose a novel Part-Stacked CNN architecture that explicitly explains the fine-grained recognition process by modeling subtle differences from object parts. Based on manually-labeled strong part annotations, the proposed architecture consists of a fully convolutional network to locate multiple object parts and a two-stream classification network that encodes object-level and part-level cues simultaneously. By adopting a set of sharing strategies between the computation of multiple object parts, the proposed architecture is very efficient running at 20 *frames/sec during inference. Experimental results on the CUB-200-2011 dataset reveal the effectiveness of the proposed architecture, from multiple perspectives of classification accuracy, model interpretability, and efficiency. Being able to provide interpretable recognition results in realtime, the proposed method is believed to be effective in practical applications.*

1. Introduction

Fine-grained visual categorization aims to distinguish objects at the subordinate level, *e.g.*, different species of birds [47, 44, 4], pets [17, 30], flowers [29, 1] and cars [38, 26]. It is a highly challenging task due to the small inter-class variance caused by highly similar subordinate categories, and the large intra-class variance by nuisance factors such as pose, viewpoint and occlusion. Inspiringly, huge progress has been made over the last few years [43, 4, 42, 18, 49], making fine-grained recognition techniques a large step closer to practical use in various applications, such as wildlife observation and surveillance systems.

Whilst numerous attempts have been made to boost the classification accuracy of fine-grained visual categorization [10, 9, 6, 22, 46], we argue that another important aspect of the problem has yet been severely overlooked, *i.e.*, the ability to generate a human-understandable "manual" on how to distinguish fine-grained categories in detail. For example, volunteers for ecological protection may certainly benefit from an algorithm that could not only classify bird species accurately, but also provide brief instructions on how to distinguish a category from its most similar subspecies - *e.g.*, a salient difference between a *Ringed-billed gull* and a *California gull* lies in the pattern on their beaks (Figure 1) - with some intuitive illustration examples. Existing fine-grained recognition methods that aim to provide a visual field guide mostly follow the routine of "part-based one-vs-one features" (POOFs) [2, 3, 4] or employ human-in-the-loop methods [20, 7, 41]. Since the data size has been increasing drastically, a method that simultaneously implements and interprets fine-grained visual categorization using the latest deep learning methods [19] is therefore highly advocated.

It is widely acknowledged that the subtle difference between fine-grained categories mostly resides in the unique

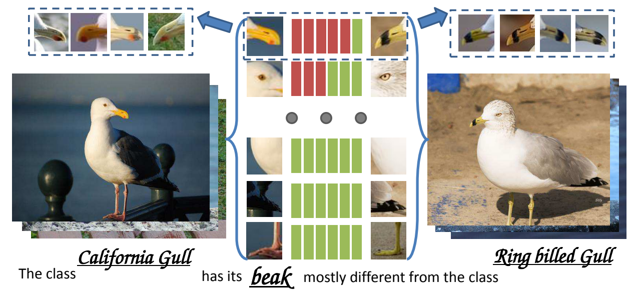

Figure 1. Overview of the proposed approach. We propose to classify fine-grained categories by modeling the subtle difference from specific object parts. Beyond classification results, the proposed PS-CNN architecture also offers human-understandable instructions on how to classify highly similar object categories explicitly.

*These authors contributed equally to this work.

properties of object parts [32, 2, 9, 27, 51, 53]. Therefore, a practical solution to interpret classification results as human-understandable manuals is to discover classification criteria from object parts. Some of existing fine-grained datasets have provided detailed part annotations including part landmarks and attributes [44, 26]. However, they are usually associated with a large number of object parts, which poses heavy computational burden for both part detection and classification. From this perspective, one would like to seek a method that follows the object-part-aware strategy to provide interpretable predicting criteria, while requiring minimum computational effort to deal with a possibly large number of parts.

In this paper, we propose a new part-based CNN architecture for fine-grained visual categorization that models multiple object parts in a unified framework with high efficiency. Similar with previous fine-grained recognition approaches, the proposed method consists of a localization module to detect object parts ("where pathway") and a classification module to classify fine-grained categories at the subordinate level ("what pathway"). In particular, we employ a fully convolutional network (FCN) to perform object part localization. The inferred part locations are fed into the classification network, in which a two-stream architecture is proposed to analyze images in both object-level (bounding boxes) and part-level (part landmarks). The computation of multiple parts is first conducted via a shared feature extraction route, then separated directly on feature maps through a part crop layer, concatenated, and then fed into a shallower network to perform object classification. Except for categorical predictions, the proposed method also generates interpretable classification instructions based on object parts. Since the proposed architecture employs a sharing strategy that stacks the computation of multiple parts together, we call it *Part-Stacked CNN* (PS-CNN).

The contributions of this paper include: 1) we present a novel and efficient part-based CNN architecture for fine-grained recognition; 2) our architecture adopts an FCN to localize object parts, which has seldom been studied before in the context of object recognition; 3) our classification network follows a two-stream structure that captures both object-level and part-level information, in which a new share-and-divide strategy is presented on the computation of multiple object parts. As a result, the proposed architecture is very efficient, with a capacity of 20 frames/sec[1] on a Tesla K80 to classify images at test time using 15 object parts; 4) The proposed method provides effective **model interpretation** for fine-grained object recognition, while being able to run in **real-time**. This is a much preferred property for practical applications, such as surveillance systems. The effectiveness of the proposed method is demonstrated through systematic experiments on the Caltech-UCSD Birds-200-2011 [44] dataset, in which we achieved 76% classification accuracy. We also present practical examples of human-understanding manuals generated by the proposed method for the task of fine-grained visual categorization.

The rest of the paper is organized as follows. Section 2 summarizes related works. The proposed architecture including the localization network and the classification network is described in Section 3. Detailed performance studies and analysis are conducted in Section 4. Section 5 concludes the paper and proposes discussions on the application scenarios of the proposed PS-CNN.

[1] For reference, a single CaffeNet runs at 50 frames/sec under the same experimental setting.

2. Related Work

Fine-Grained Visual Categorization. A number of methods have been developed to classify object categories at the subordinate level. Recently, the best performing methods mostly sought for improvement brought by the following three aspects: more discriminative features including deep CNNs for better visual representation [5, 33, 19, 39, 37], explicit alignment approaches to eliminate pose displacements [6, 14], and part-based methods to study the impact of object parts [2, 52, 27, 51, 15, 55]. Another line of research explored human-in-the-loop methods [8, 10, 45] to identify the most discriminative regions for classifying fine-grained categories. Although such methods provided direct references of how people perform fine-grained recognition in real life, they were impossible to scale for large systems due to the need of human interactions at test time.

Current state-of-the-art methods for fine-grained recognition are part-based R-CNN by Zhang *et al*. [51] and Bilinear CNN by Lin *et al*. [22], which both employed a two-stage pipeline of part detection and part-based object classification. The main idea of the proposed PS-CNN is largely inherited from [51], who first detected the location of two object parts and then trained an individual CNN based on the unique properties of each part. Compared to part-based R-CNN, the proposed method is far more efficient in both detection and classification phrases. As a result, we are able to employ much more object parts than that of [51], while still being significantly faster at test time.

On the other hand, Lin *et al*. [22] argued that manually defined parts were sub-optimal for the task of object recognition, and thus proposed a bilinear model consisting of two streams whose roles were interchangeable as detectors or features. Although this design enjoyed the data-driven nature that could possibly lead to optimal classification performance, it also made the resultant model hard to interpret. On the contrary, our method tries to balance the need of both both classification accuracy and model interpretability in fine-grained recognition systems.

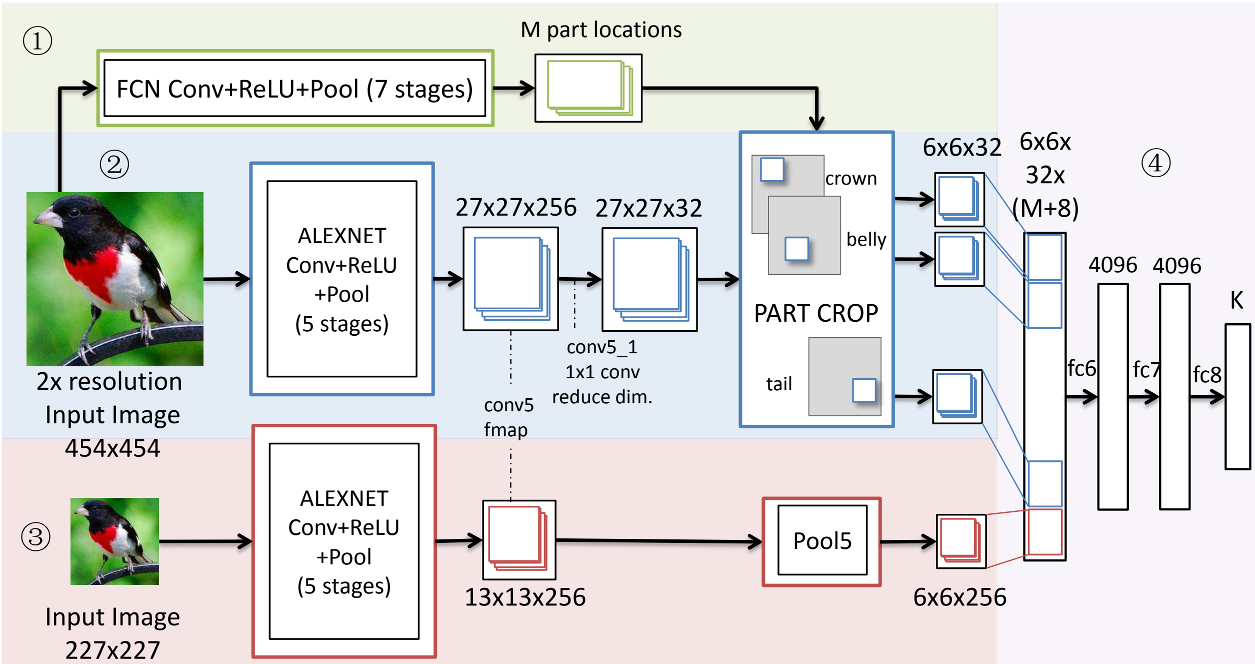

Figure 2. Network architecture of the proposed Part-Stacked CNN model. The model consists of: 1) a fully convolutional network for part landmark localization; 2) a part stream where multiple parts share the same feature extraction procedure, while being separated by a novel part crop layer given detected part locations; 3) an object stream with lower spatial-resolution input images to capture bounding-box level supervision; and 4) three fully connected layers to achieve the final classification results based on a concatenated feature map containing information from all parts and the bounding box.

Fully Convolutional Networks. Fully convolutional network (FCN) is a fast and effective approach to produce dense prediction with convolutional networks. Successful examples can be found on tasks including sliding window detection [34], semantic segmentation [23], and human pose estimation [40].

3. Part-Stacked CNN

We present the model architecture of the proposed Part-Stacked CNN in this section. In accordance with the common framework for fine-grained recognition, the proposed architecture is decomposed into a *Localization Network* (Section 3.1) and a *Classification Network* (Section 3.2). We adopt CaffeNet [16], a slightly modified version of the standard seven-layer AlexNet [19] architecture, as the basic structure of the network; deeper networks could potentially lead to better recognition accuracy, but may also result in lower efficiency.

A unique design in our architecture is that the message transferring operation from the localization network to the classification network, *i.e*. using detected part locations to perform part-based classification, is conducted directly on the *conv5* output feature maps within the process of data forwarding. It is a significant difference compared to the standard two-stage pipeline of part-based R-CNN [51] that consecutively localizes object parts and then trains part-specific CNNs on the detected regions. Based on this design, a set of sharing schemes are performed to make the proposed PS-CNN fairly efficient for both learning and inference. Figure 2 illustrates the overall network architecture.

3.1. Localization Network

The first stage of the proposed architecture is a localization network that aims to detect the location of object parts. We employ the simplest form of part landmark annotations, *i.e*. a 2D key point is annotated at the center of each object part. Assume that M - the number of object parts labeled in the dataset, is sufficient large to offer a complete set of object parts on which fine-grained categories are usually different from each other. Motivated by recent progress of human pose estimation [23] and semantic segmentation [40], we adopt a fully convolutional network (FCN) [28] to generate dense output feature maps for locating object parts.

Fully convolutional network. A fully convolutional network is achieved by replacing the parameter-rich fully connected layers in standard CNN architectures by convolutional layers with kernels in spatial size of 1×1. Given an input RGB image, the output of a fully convolutional network is a *feature map* in reduced dimension compared to

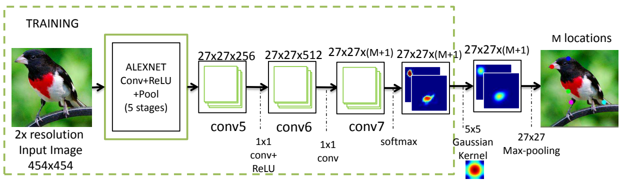

Figure 3. Demonstration of the localization network. Training process is denoted inside the dashed box. For inference, a Gaussian kernel is then introduced to remove noise. The results are M 2D part locations in the 27×27 *conv5* feature map.

the input. The computation of each unit in the feature map only corresponds to pixels inside a region with fixed size in the input image, which is called its *receptive field*. FCN is preferred in our framework due to the following three reasons: 1) feature maps generated by FCN can be directly utilized as the part locating results in the classification network, which will be detailed in Section 3.2; 2) results of multiple object parts can be obtained simultaneously using an FCN; 3) FCN is very efficient in both learning and inference.

Learning. We model the part localization process as a multi-class classification problem on dense output spatial positions. In particular, suppose the output of the last convolutional layer in the FCN is in the size of $h \times w \times d$, where h and w are spatial dimensions and d is the number of channels. We set $d = M + 1$. Here M is the number of object parts and 1 denotes for an additional channel to model the background. To generate corresponding ground-truth labels in the form of feature maps, units indexed by $h \times w$ spatial positions are labeled by their nearest object part; units that are not close to any of the labeled parts (with an overlap < 0.5 with respect to receptive field) are labeled as background.

A practical problem here is to determine the model depth and the size of input images for training the FCN. Generally speaking, layers at later stages carry more discriminative power and thus are more likely to generate promising localization results; however, their receptive fields are also much larger than those of previous layers. For example, the receptive field of *conv5* layer in CaffeNet has a size of 163×163 compared to the 227×227 input image, which is too large to model an object part. We propose a simple trick to deal with this problem, *i.e.*, upsampling the input images so that the fixed-size receptive fields denoting object parts become relatively smaller compared to the whole object, while still being able to use layers at later stages to guarantee enough discriminative power.

The localization network in the proposed PS-CNN is illustrated in Figure 3. The input of the FCN is a bounding-box-cropped RGB image, warped and resized into a fixed size of 454×454. The structure of the first five layers is identical to those in CaffeNet, which leads to a $27 \times 27 \times 256$ output after *conv5* layer. Afterwards, we further introduce a 1×1 convolutional layer with 512 output channels as *conv6*, and another 1×1 convolutional layer with $M + 1$ outputs termed *conv7* to perform classification. By adopting a spatial preserving softmax that normalizes predictions at each spatial location of the feature map, the final loss function is a sum of softmax loss at all 27×27 positions:

$$L = -\sum_{h=1}^{27} \sum_{w=1}^{27} \log \sigma(h, w, \hat{c}), \tag{1}$$

where

$$\sigma(h, w, \hat{c}) = \frac{\exp(f_{conv7}(h, w, \hat{c}))}{\sum_{c=0}^{M} \exp(f_{conv7}(h, w, c))}.$$

Here, $\hat{c} \in [0, 1, ..., M]$ is the part label of the patch at location (h, w), where the label 0 denotes background. $f_{conv7}(h, w, c)$ stands for the output of *conv7* layer at spatial position (h, w) and channel c.

Inference. The inference process starts from the output of the learned FCN, *i.e.*, $(M+1)$ part-specific heat maps in the size of 27×27, in which we introduce a Gaussian kernel $\mathcal{G}$ to remove isolated noise in the feature maps. The final output of the localization network are M locations in the 27×27 *conv5* feature map, each of which is computed as the location with the maximum response for one object part.

Meanwhile, considering that object parts may be missing in some images due to varied poses and occlusion, we set a threshold μ that if the maximum response of a part is below μ, we simply discard this part's channel in the classification network for this image. Let $g(h, w, c) = \sigma(h, w, c) * \mathcal{G}$, the inferred part locations are given as:

$$(h_c^*, w_c^*) = \begin{cases} \text{argmax}_{h,w}\, g(h, w, c) & \text{if } g(h_c^*, w_c^*, c) > \mu, \\ (-1, -1) & \text{otherwise.} \end{cases} \tag{2}$$

3.2. Classification network

The second stage of the proposed PS-CNN is a classification network with the inferred part locations given as an input. It follows a two-stream architecture with a *Part Stream* and a *Object Stream* to capture semantics from multiple levels. A sub-network consisting of three fully connected layers is then performed as an object classifier, as shown in Figure 2.

Part stream. The part stream acts as the core of the proposed PS-CNN architecture. To capture object-part-dependent differences between fine-grained categories, one can train a set of part CNNs, each one of which conducts classification on a part separately, as proposed by Zhang *et al*. [51]. Although such method worked well for [51] who only employed two object parts, we argue that it is not applicable when the number of object parts is much larger in our case, because of the high time and space complexity.

In PS-CNN, we introduce two strategies to improve the efficiency of the part stream. The first one is model parameter sharing. Specifically, model parameters of the first five convolutional layers are shared among all object parts, which can be regarded as a generic part-level feature extractor. This strategy leads to less parameters in the proposed architecture and thus reduces the risk of overfitting.

Other than model parameter sharing, we also conduct a computational sharing strategy. The goal is to make sure that the feature extraction procedure of all parts only requires one pass through the convolutional layers. Analogous to the localization network, the input images of the part stream are in doubled resolution 454×454 so that the respective receptive fields are not too large to model object parts; forwarding the network to *conv5* layer generates output feature maps of size 27×27. By far, the computation of all object parts is completely shared.

After performing the shared feature extraction procedure, the computation of each object part is then partitioned through a *part crop layer* to model part-specific classification cues. For each part, the part crop layer extracts a local neighborhood region centered at the detected part location. Features outside the cropped region are simply dropped. In practice, we crop 6×6 neighborhood regions out of the 27×27 *conv5* feature maps to match the output size of the object stream. The resultant receptive fields for the cropped feature maps has a width of 243, given the receptive field size of *conv5* layers and the respective stride.

Object stream. The object stream utilizes bounding-box-level supervision to capture object-level semantics for fine-grained recognition. It follows the general architecture of CaffeNet, in which the input of the network is a 227×227 RGB image and the output of *pool5* layer are 6×6 feature maps.

We find the design of the two-stream architecture in PS-CNN analogous to the famous Deformable Part-based Models [12], in which object-level features are captured through a root filter in a coarser scale, while detailed part-level information is modeled by several part filters at a finer scale. We find it critical to measure visual cues from multiple semantic levels in an object recognition algorithm.

Dimension reduction and fully connected layers. The aforementioned two-stream architecture generates an individual feature map for each object part and bounding box. When conducting classification, they serve as an overcomplete set of CNN features from multiple scales. Following the standard CaffeNet architecture, we employ a DNN including three fully connected layers as object classifiers. The first fully connected layer *fc6* now becomes a part concatenation layer whose input is generated by stacking the output feature maps of the part stream and the object stream together. However, such a concatenating process requires $M + 1$ times more model parameters than the original *fc6* layer in CaffeNet, which leads to a huge memory cost.

To reduce model parameters, we introduce a 1×1 convolutional layer termed *conv5_1* in the part stream that projects the 256 dimensional *conv5* output to 32-d. It is identical to a low-rank projection of the model output and thus can be initialized through standard PCA. Nevertheless, in our experiments, we find that directly initializing the weights of the additional convolution by PCA in practice worsens the performance. To enable domain-specific fine-tuning from pre-trained CNN model weights, we train an auxiliary CNN to initialize the weights for the additional convolutional layer.

Let $X^c \in \mathbb{R}^{N \times M \times 6 \times 6}$ be the c^{th} 6×6 cropped region around the center point (h_c^*, w_c^*) from *conv5_1* feature maps $X \in \mathbb{R}^{N \times M \times 27 \times 27}$, where (h_c^*, w_c^*) is the predicted location for part c and N is the number of output feature maps. The output of part concatenation layer *fc6* can be formulated as:

$$f_{out}(X) = \sigma(\sum_{c=1}^{M} (W^c)^T X^c), \tag{3}$$

where W^c is the model parameters for part c in *fc6* layer, and σ is an activation function.

We conduct the standard gradient descent method to train the classification network. The most complicated part for computing gradients lies in the dimension reduction layer due to the impact of part cropping. Specifically, the gradient of each cropped part feature map (in 6×6 spatial resolution) is projected back to the original size of *conv5* (27×27 feature maps) according to the respective part location and then summed up. Note that the proposed PS-CNN is implemented as a two stage framework, *i.e.* after training the FCN, weights of the localization network are fixed when training the classification network.

4. Experiments

We present experimental results and analysis of the proposed method in this section. Specifically, we will evaluate the performance through four different aspects: localization accuracy, classification accuracy, inference efficiency, and model interpretation.

4.1. Dataset and implementation details

Experiments are conducted on the widely used fine-grained classification benchmark the Caltech-UCSD Bird-

part	throat	beak	crown	forehead	right eye	nape	left eye	back
APK	0.908	0.894	0.894	0.885	0.861	0.857	0.850	0.807
part	breast	belly	right leg	tail	left leg	right wing	left wing	overall
APK	0.799	0.794	0.775	0.760	0.750	0.678	0.670	0.866

Table 1. *APK* for each object part in the CUB-200-2011 test set in descending order.

Figure 4. Typical localization results on CUB-200-2011 test set. We show 6 of the 15 detected parts here. They are: beak (red), belly (green), crown (blue), right eye (yellow), right leg (magenta), tail (cyan). Better viewed in color.

s dataset (CUB-200-2011) [44]. The dataset contains 200 bird categories with roughly 30 training images per category. In the training phase we adopt strong supervision available in the dataset, *i.e.* we employ 2D key point part annotations of altogether $M = 15$ object parts together with image-level labels and object bounding boxes.

The proposed Part-Stacked CNN architecture is implemented using the open-source package Caffe [16]. Specifically, bounding-box cropped input images are warped to a fixed size of 512×512, randomly cropped into 454×454, and then fed into the localization network and the part stream in the classification network as input. We employ a pooling layer in the object stream that downsamples the 454×454 input to 227×227 to guarantee synchronization between the two streams in the classification network.

4.2. Localization results

As the localization results in our method are directly delivered to the classification network at feature-map-level, we do not intend to achieve accurate keypoint localization at pixel-level but instead focus on a rougher correctness measure. The localization correctness is quantitatively assessed using *APK* (Average Precision of Key points) [50]. Following [24], we consider a key point to be correctly predicted if the prediction lies within a Euclidean distance of α times the maximum of the bounding box width and height compared to the ground truth. We set $\alpha = 0.1$ in all the analysis below.

The adopted FCN architecture in PS-CNN achieves a reasonably inspiring 86.6% *APK* on the test set of CUB-200-2011 for 15 object parts. Specifically, the additional $1\times$ 1 convolutional layer and the employed Gaussian smoothing kernel delivers 1.5% and 2% improvements over the results using standard five convolutional layers in AlexNet, respectively.

BBox only	+2 part	+4 part	+8 part	+15 part
69.08	73.72	74.84	76.63	76.41

Table 2. The effect of increasing the number of object parts on the classification accuracy.

Furthermore, we present per part *APK*s in Table 1. An interesting phenomenon here is that parts residing near the head of the birds tend to be located more accurately. It turns out that the birds' head has relatively more stable structure with less deformations and lower probability to be occluded. On the contrary, parts that are highly deformable such as wings and legs get lower *APK* values. Figure 4 shows typical localization results of the proposed method.

4.3. Classification results

We begin the analysis of classification results by a study on the discriminative power of each object part. Each time we select one object part as the input and discard the computation of all other parts. Different parts reveal significantly different classification results. The most discriminative part *crown* itself achieves a quite impressive accuracy of 57%, while the lowest accuracy is only 10% for part *beak*. Therefore, to obtain better classification results, it may be beneficial to find a rational combination or order of object parts instead of directly ran the experiments on all parts altogether.

We therefore introduce a strategy that incrementally adds object parts to the whole framework and iteratively trains the model. Specifically, starting from a model trained on bounding-box supervision only, which is also the baseline of the proposed method, we iteratively insert object parts into the framework and re-finetune the PS-CNN model. The number of parts inserted in each iteration increases exponentially, *i.e.*, in the i^{th} iteration, 2^i parts are selected and inserted. When starting from an initialized model with relatively high performance, introducing a new object part into the framework does not require to run a brand new classification procedure based on this specific part alone; ideally only the classification of highly confusing categories that may be distinguished through the new part will be impacted and amended. As a result, this procedure overcomes the drawback raised by the existence of object parts with lower discriminative power. In our implementation, the ordering of part inclusion is determined by its discriminative power measured by the classification accuracy using each part only (see Supplementary for details). Table 2 reveals that as the number of object parts increases from 0 to 8, the classification accuracy improves gradually and then becomes saturated. Further increasing the part number does not lead to a better accuracy; however, it does provide more resources for performing explicit model interpretation.

Table 3 shows the performance comparison between PS-CNN and existing fine-grained recognition methods. Since the CNN architecture has a large impact on the recognition performance, for fair comparison, we only compare results reported on the standard seven-layer architecture. Deeper models could surely lead to better accuracy, but also result in less efficiency. The complete PS-CNN model with a bounding-box and 15 object parts achieves 76% accuracy, which is comparable with part-based R-CNN [51], while being slightly lower than several most recent state-of-the-art methods [22, 21, 35] due to the effectiveness-efficiency tradeoff. In particular, our model is over two orders of magnitude faster than [51], requiring only 0.05 seconds to perform end-to-end classification on a test image. This number is quite inspiring, especially considering the number of parts used in the proposed method. The efficiency makes it possible for the proposed method to be conducted in real-time, leading to potential applications in video domain.

Method	Train Anno.	Test Anno.	Acc.
Constellation [36]	n/a	n/a	68.5
Attention [48]	n/a	n/a	69.7
Bilinear-CNN [22]	n/a	n/a	74.2
Weak FGVC [54]	n/a	n/a	75.0
CNNaug [31]	BBox	BBox	61.8
Alignment [13]	BBox	BBox	67.0
No parts [18]	BBox	BBox	74.9
Bilinear-CNN [22]	BBox	BBox	**80.4**
Part R-CNN [51]	BBox+Parts	n/a	73.9
PoseNorm CNN [6]	BBox+Parts	n/a	75.7
POOF [2]	BBox+Parts	BBox	56.8
DPD+DeCAF[11]	BBox+Parts	BBox	65.0
Deep LAC [21]	BBox+Parts	BBox	80.2
Multi-proposal [35]	BBox+Parts	BBox	80.3
Part R-CNN [51]	BBox+Parts	BBox	76.4
PS-CNN (this paper)	BBox+Parts	BBox	76.6

Table 3. Comparison with state-of-the-art methods on the CUB-200-2011 dataset. To conduct fair comparisons, for all the methods using deep features, we report their results on the standard seven-layer architecture (mostly *ALexNet* except *VGG-m* for [22]) if possible. Note that our method achieves comparable results with state-of-the-art while running in real-time.

4.4. Model interpretation

One of the most prominent features of the proposed Part-Stacked CNN (PS-CNN) method is that it can produce human-understandable interpretation manuals for fine-grained recognition. Here we detail the algorithm we use to perform interpretation using the proposed method.

Different from [2] who directly conducted one-on-one classification on object parts, the interpretation process of the proposed method is conducted in a relatively indirect way. Considering that using each object part by itself cannot lead to convincing classification results, we perform the analysis for interpretation on a combination of bounding box supervision and each single object part. The analysis is performed in two ways: a "one-versus-rest" comparison for denoting the most discriminative part to classify a subcategory from all other classes, and a "one-versus-one" comparison to find out the classification criteria of a subcategory with its most similar classes.

- The *"one-versus-rest"* manual for an object category k. For every part p, we compute the summation of prediction scores of the category's positive samples. The most discriminative part is then captured as the one with the largest accumulated score:

$$p_k^* = \underset{p}{\operatorname{argmax}} \sum_{i, y_i = k} S_{ip}^{(p)}. \tag{4}$$

- The *"one-versus-one"* manual obtained by computing as the part which results in the largest difference of prediction scores on two categories k and l. We first pick up the respective two rows in the score matrix S, and re-normalize it using the binary classification criterion as S'. Afterwards, the most discriminative part is given as:

$$p_{k \to l}^* = \underset{p}{\operatorname{argmax}} \Big(\sum_{i, y_i = k} S_{ip}^{\prime(p)} + \sum_{j, y_j = l} S_{jp}^{\prime(p)} \Big) \tag{5}$$

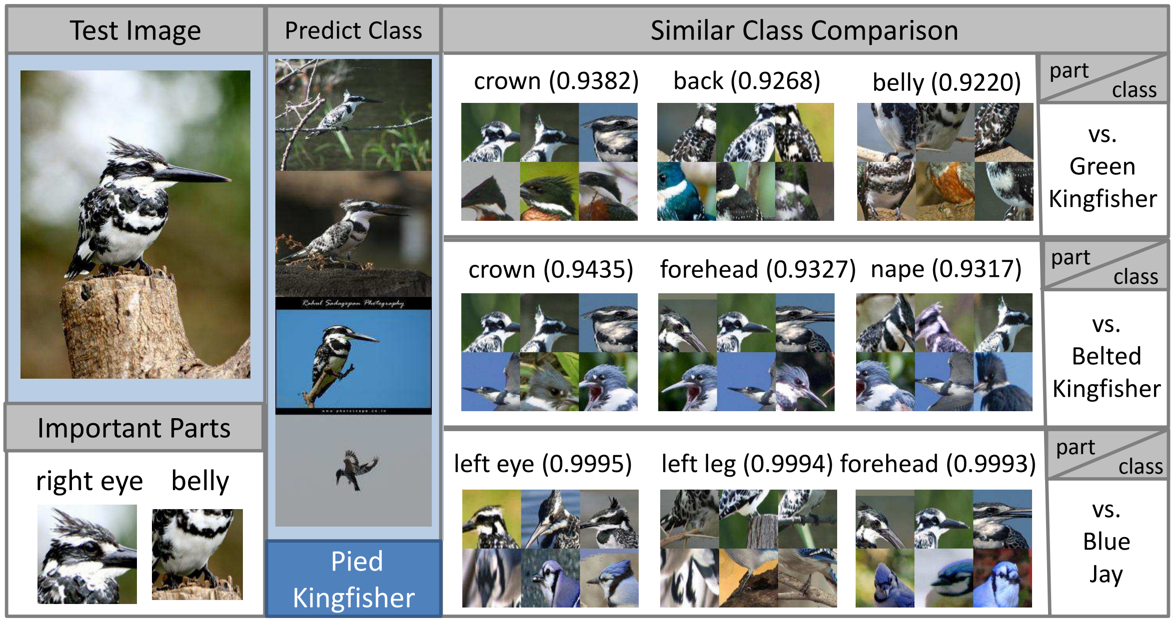

Figure 5. Example of the prediction manual generated by the proposed approach. Given a test image, the system reports its predicted class label with some typical exemplar images. Part-based comparison criteria between the predicted class and its most similar classes are shown in the right part of the image. The number in brackets shows the confidence of classifying two categories by introducing a specific part. We present top three object parts for each pair of comparison. For each of the parts, three part-center-cropped patches are shown for the predicted class (upper rows) and the compared class (lower rows) respectively.

The model interpretation routine is demonstrated in Figure 5. When a test image is presented, the proposed method first conducts object classification through the PS-CNN architecture. The predicted category is presented by a set of images in the dataset that are closest to the test image according to *conv5_1* outputs. Except for classification results, the proposed method also presents classification criteria for distinguishing the predicted category from its most similar neighbor classes based on object parts. Again we use the output of *conv5_1* layer but after performing part cropping to retrieve nearest neighbor part patches of the input test image. The procedure described above provides an intuitive visual guide for distinguishing fine-grained categories.

5. Conclusion

In this paper, we proposed a novel model for fine-grained recognition called Part-Stacked CNN. The model exploited detailed part-level supervision, in which object parts were first located by a fully convolutional network, following by a two-stream classification network that explicitly captured object-level and part-level information. Experiments on the CUB-200-2011 dataset revealed the effectiveness and efficiency of PS-CNN, especially the impact of introducing object parts on fine-grained visual categorization tasks. Meanwhile, we have presented human-understandable interpretations of the proposed method, which can be used as a visual field guide for studying fine-grained categorization.

We have discussed the application of the proposed Part-Stacked CNN on fine-grained visual categorization with strong supervision. In fact, PS-CNN can be easily generalized for varied applications. Examples include:

1) Discarding the requirement of strong supervision. Instead of introducing manually-labeled part annotations for generating human-understandable visual guides, one can also exploit unsupervised part discover methods [18] to define object parts automatically, which requires far less human labeling effort.

2) Attribute learning. The application scenario of PS-CNN is not restricted to FGVC. For instance, performance of online shopping [25] could definitely benefit from clothing attribute analysis from local parts provided by PS-CNN.

3) Context-based CNN. The role of local “parts” in PS-CNN is interchangeable with global contexts, especially for objects that are small in size and have no obvious object parts, such as volleyballs or tennis balls.

Acknowledgements

The work is partially supported by Australian Research Council Projects DP-140102164, FT-130101457, and LE-140100061, the High Technology Research and Development Program of China 2015AA015801, NSFC 61221001, STCSM 12DZ2272600, and the 111 Project B07022.

References

[1] A. Angelova, S. Zhu, and Y. Lin. Image segmentation for large-scale subcategory flower recognition. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 39–45. IEEE, 2013. 1

[2] T. Berg and P. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 955–962, 2013. 1, 2, 7

[3] T. Berg and P. N. Belhumeur. How do you tell a blackbird from a crow? In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 9–16. IEEE, 2013. 1

[4] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Computer Vision and Pattern Recognition (CVPR), 2014*, pages 2019–2026. IEEE, 2014. 1

[5] L. Bo, X. Ren, and D. Fox. Kernel descriptors for visual recognition. In *Advances in neural information processing systems*, pages 244–252, 2010. 2

[6] S. Branson, G. Van Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952*, 2014. 1, 2, 7

[7] S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie. The ignorant led by the blind: A hybrid human–machine vision system for fine-grained categorization. *International Journal of Computer Vision*, 108(1-2):3–29, 2014. 1

[8] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *Computer Vision–ECCV 2010*, pages 438–451. Springer, 2010. 2

[9] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 321–328, 2013. 1, 2

[10] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 580–587. IEEE, 2013. 1, 2

[11] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 7

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010. 5

[13] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV 2013*, pages 1713–1720. IEEE, 2013. 7

[14] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Local alignments for fine-grained categorization. *International Journal of Computer Vision*, 111(2):191–212, 2015. 2

[15] G. Gkioxari, R. Girshick, and J. Malik. Actions and attributes from wholes and parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2470–2478, 2015. 2

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. 3, 6

[17] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, 2011. 1

[18] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5546–5555, 2015. 1, 7, 8

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 2, 3

[20] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision–ECCV 2012*, pages 502–516. Springer, 2012. 1

[21] D. Lin, X. Shen, C. Lu, and J. Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1666–1674, 2015. 7

[22] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015. 1, 2, 7

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 3

[24] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014. 6

[25] K. M. Hadi, H. Xufeng, L. Svetlana, B. Alexander, and B. Tamara. Where to buy it: Matching street clothing photos in online shops. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, 2015. 8

[26] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1, 2

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

[28] O. Matan, C. J. Burges, Y. Le Cun, and J. S. Denker. Multi-digit recognition using a space displacement neural network. 1995. 3

[29] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008. 1

[30] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3498–3505. IEEE, 2012. 1

[31] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014. 7

[32] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive psychology*, 8(3):382–439, 1976. 2

[33] J. Sánchez, F. Perronnin, and Z. Akata. Fisher vectors for fine-grained visual categorization. In *FGVC Workshop in IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011. 2

[34] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 3

[35] K. J. Shih, A. Mallya, S. Singh, and D. Hoiem. Part localization using multi-proposal consensus for fine-grained categorization. *arXiv preprint arXiv:1507.06332*, 2015. 7

[36] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1143–1151, 2015. 7

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[38] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little, B. Schiele, and D. Koller. Fine-grained categorization for 3d scene understanding. *International Journal of Robotics Research*, 30(13):1543–1552, 2011. 1

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 2

[40] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2014. 3

[41] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015. 1

[42] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko, D. Weiss, et al. Understanding objects in detail with fine-grained attributes. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3622–3629. IEEE, 2014. 1

[43] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2524–2531. IEEE, 2011. 1

[44] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1, 2, 6

[45] C. Wah, G. Van Horn, S. Branson, S. Maji, P. Perona, and S. Belongie. Similarity comparisons for interactive fine-grained categorization. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 859–866. IEEE, 2014. 2

[46] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang. Multiple granularity descriptors for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2399–2406, 2015. 1

[47] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. 2010. 1

[48] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 842–850, 2015. 7

[49] Z. Xu, S. Huang, Y. Zhang, and D. Tao. Augmenting strong supervision using web data for fine-grained categorization. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, 2015. 1

[50] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2878–2890, 2013. 6

[51] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014. 2, 3, 5, 7

[52] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1637–1644. IEEE, 2014. 2

[53] X. Zhang, H. Xiong, W. Zhou, and Q. Tian. Fused one-vs-all mid-level features for fine-grained visual categorization. In *Proceedings of the ACM International Conference on Multimedia*, pages 287–296. ACM, 2014. 2

[54] Y. Zhang, X.-s. Wei, J. Wu, J. Cai, J. Lu, V.-A. Nguyen, and M. N. Do. Weakly supervised fine-grained image categorization. *arXiv preprint arXiv:1504.04943*, 2015. 7

[55] J. Zhu, X. Chen, and A. L. Yuille. Deepm: A deep part-based model for object detection and semantic part localization. *arXiv preprint arXiv:1511.07131*, 2015. 2