SPDA-CNN: Unifying Semantic Part Detection and Abstraction for Fine-grained Recognition

Han Zhang*[1], Tao Xu*[2], Mohamed Elhoseiny[1],
Xiaolei Huang[2], Shaoting Zhang[3], Ahmed Elgammal[1], Dimitris Metaxas[1]

[1]Department of Computer Science, Rutgers University
[2]Department of Computer Science and Engineering, Lehigh University
[3]Department of Computer Science, University of North Carolina at Charlotte

Abstract

Most convolutional neural networks (CNNs) lack mid-level layers that model semantic parts of objects. This limits CNN-based methods from reaching their full potential in detecting and utilizing small semantic parts in recognition. Introducing such mid-level layers can facilitate the extraction of part-specific features which can be utilized for better recognition performance. This is particularly important in the domain of fine-grained recognition.

In this paper, we propose a new CNN architecture that integrates semantic part detection and abstraction (SPDA-CNN) for fine-grained classification. The proposed network has two sub-networks: one for detection and one for recognition. The detection sub-network has a novel top-down proposal method to generate small semantic part candidates for detection. The classification sub-network introduces novel part layers that extract features from parts detected by the detection sub-network, and combine them for recognition. As a result, the proposed architecture provides an end-to-end network that performs detection, localization of multiple semantic parts, and whole object recognition within one framework that shares the computation of convolutional filters. Our method outperforms state-of-the-art methods with a large margin for small parts detection (e.g. our precision of 93.40% vs the best previous precision of 74.00% for detecting the head on CUB-2011). It also compares favorably to the existing state-of-the-art on fine-grained classification, e.g. it achieves 85.14% accuracy on CUB-2011.

1. Introduction

Fine-grained recognition aims to distinguish among subordinate categories, such as identifying product models [31, 26, 30] and discriminating animal and plant species [37, 23]. Compared to generic object recognition, this task is more challenging since the subtle visual differences can be easily overwhelmed by the other factors such as poses and viewpoints. Humans typically refer to the difference in some semantic parts to distinguish subordinate categories. Thus, detecting and fully utilizing object parts is extremely important in fine-grained object recognition.

A majority of fine-grained recognition methods have incorporated part localization. State-of-the-art methods apply convolutional neural networks (CNNs) to detect part regions [41, 28]. However, they do not model or utilize small semantic parts. For example, on the CUB-2011 bird dataset [37], both methods [41, 28] only localized the head and body, *i.e.*, large parts, and they did not utilize other smaller parts such as the tail and wings although these parts can be very useful for recognition [9]. The head and body detection results by these two methods also show that the results for the head are consistently worse than that of the body because of the head's smaller size. To the best of our knowledge, existing CNN-based fine-grained classification methods have not focused on the detection and utilization of small semantic parts.

Traditional CNNs lack mid-level layers that model semantic parts of objects. In order to introduce such layers to facilitate the extraction of part-specific features, several works proposed part-based CNN methods [41, 5, 28, 39]. These methods define and train a separate CNN network for each part. Features extracted from each part are then concatenated into a long vector and used to train a separate classifier (*e.g.*, SVM) for the final classification. This framework has several limitations, however. It makes training and testing a multi-stage process, and makes the sharing of convolutional filters among the separate part networks difficult. Furthermore, it limits the ability of the overall architecture to learn correlations among different parts, which

*Indicates equal contribution. Email: han.zhang@rutgers.edu

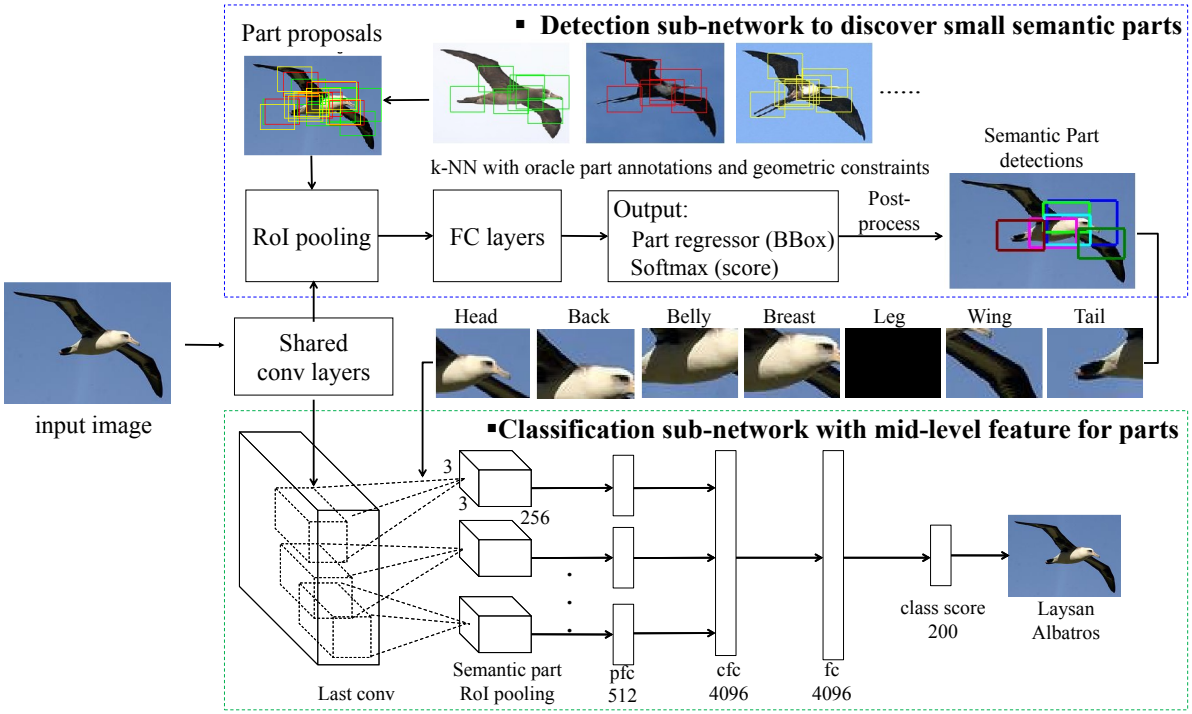

Figure 1. SPDA-CNN: Unifying Semantic Part Detection and Abstraction for fine-grained classification. In the detection sub-network, we propose a novel top-down k nearest neighbor (k-NN) method to generate proposals for small semantic parts. The number of our k-NN proposals is about one order less than the traditional region proposal methods (*e.g.*, selective search [36]). Furthermore our proposals inherit prior geometric constraints from the nearest neighbors. Given k-NN proposals, our part detection network applies Fast RCNN [13] to regress and obtain much more accurate part bounding boxes compared with the directly transfer method [16]. The final part detections are then sent to the part-abstraction and classification sub-network. The invisible/occluded parts (such as leg here) are represented by zeros. To get an abstraction of semantic parts, combine them and learn the correlation among them for recognition, we propose to add a semantic part RoI pooling layer, a part-based fully connected layer (pfc), and a concatenation fully connected layer (cfc) to the traditional CNN framework. By sharing the computation of convolutional filters, the proposed architecture provides an end-to-end network that performs detection, localization of multiple semantic parts, and whole object recognition within one framework.

can be essential to recognition. Therefore, an end-to-end CNN framework with mid-level semantic part abstraction layers is needed, in particular for fine-grained classification.

To tackle these above-mentioned challenges, we propose a new CNN architecture with built-in mid-level part abstraction layers. As shown in Figure 1, the proposed architecture has two sub-networks: a detection sub-network and a part-abstraction and recognition sub-network. The contribution of our paper is threefold: (1) A novel top-down proposal method is designed to generate small semantic part candidates for multiple semantic parts detection. As a result, our detection sub-network outperforms state-of-the-art methods for small parts; *e.g.*, the precision of head is improved from 74.00% to 93.40% on the CUB-2011 dataset. (2) A new type of part-based layers is proposed in the recognition sub-network, which provides an abstraction of small semantic parts, extracts part-based features and combines them for recognition. Our recognition sub-network achieves state-of-the-art performance. (3) We further integrate the part detection and part-based recognition sub-networks into a unified architecture to form an end-to-end system for fine-grained classification; in this way, the sub-networks can easily share the computation of convolutional filters.

2. Related Work

Subordinate classes within a category generally share common appearances with subtle differences at certain parts. Therefore, localizing object parts and extracting discriminative part features play crucial roles in fine-grained image recognition. Some of the pioneering works in this research direction use low-level image features for part localization and part feature abstraction. Among them, DPM [11, 42, 7] and Poselet [4, 10] have been extensively utilized to localize object parts from different poses and viewpoints. Other works [16, 12] transferred part locations to a test image from training samples with the most similar global shapes. Göring *et al.* [16] extracted hand-crafted features from each part for the final classification; this method achieved promising classification results on the Caltech-UCSD birds datasets [37, 38], because all 15 small semantic parts of the bird were used. Since they directly

transferred part regions from training samples to a test image, however, the transferred regions suffer from low overlapping with the ground truth; by running their source code, we found the average overlap between the transferred part regions and the ground truth is only 0.45.

Currently, methods based on CNNs [41, 25, 28, 5, 15, 18] significantly outperform previous works that rely on hand-crafted features for part detection, part abstraction and fine-grained classification. For example, Zhang *et al*. [41] applied the bottom-up selective search method [36] to generate part and object proposals and use RCNN [14] to perform detection. It was difficulty for their selective search method to propose small semantic part regions. So they only utilized two big parts (*i.e*., head and torso). Also because there are no geometric constraints among selective search proposals, they had to provide extra hand-crafted geometric constraints to further filter the detection results. Lin *et al*. [28] directly regressed part bounding box coordinates from CNN features and proposed to use valve linkage function to join part localization, alignment and class prediction in one network for each part. However, this method also only used the head and torso. Other unsupervised methods [24, 25, 39, 34] could generate multiple object parts, but they are not guaranteed to produce small parts with semantic meanings. On the other hand, many of those part-based CNN methods [41, 28, 5] followed the multi-stage CNN-SVM scheme for fine-grained classification, which makes the training process expensive and also restricts the usage of more semantic parts. Although [40] has shown some neurons in CNN might implicitly capture part or attribute information, there is no evidence that part-level features are well modeled in the current architecture.

3. Our Approach

As illustrated in Figure 1, the proposed architecture integrates a detection sub-network and a part-abstraction and recognition sub-network. The two sub-networks share a common set of convolutional layers. In this section we explain the details of these two sub-networks.

3.1. Part Detection Sub-network

3.1.1 Geometrically-constrained Top-down Region Proposals for Small Semantic Parts

Small semantic object parts are hard to detect since they may not have distinct visual features compared to the rest of the object. On the other hand, their rough locations can easily be estimated if we know the global shape of the object and geometric constraints among parts are utilized. However, traditional region proposal methods [36, 6, 1, 43] often focus on bottom-up image cues ignoring geometric constraints, thus fail to generate region candidates for small semantic parts. In this paper, inspired by the recent success of nonparametric part transfer methods [16, 12] for fine-grained recognition, we propose a geometrically-constrained part proposal method similar to the k Nearest Neighbors (k-NN) approach to generate candidate part regions for detection.

The proposed method is a top-down scheme. First, histograms of oriented gradients (HOG) in the bounding box of the object are computed to represent its rough global shape. Then based on HOG features, the k nearest neighbors of the given image are retrieved from the training dataset. All part regions of each neighbor are scaled proportionally according to the size of the given test image. Let $B = [b_{11}, ..., b_{1m}, b_{21}, ..., b_{2m}, ..., b_{k1}, ..., b_{km}]$ denote all the transferred part bounding boxes, where m is the number of parts in each object. These transferred parts inherit the prior information from nearest neighbors, which have oracle part annotations and geometric constraints among parts. To generate part region proposals from those transferred regions, we investigate two types of priors.

1. *Strong prior* is the prior information that inherits both part class label and part geometric constraints from the nearest neighbors. With this type of prior, we generate proposals for the i-th part of the given image using transferred part locations $[b_{1i}, b_{2i}, ..., b_{ki}]$. Thus, the number of proposals for each part is k and the total number for all parts is $N = km$.

2. *Weak prior* is the prior that is not restricted to the prior part class label compared to the strong prior. That is, $[b_{11}, ..., b_{1m}, b_{21}, ..., b_{2m}, ..., b_{k1}, ..., b_{km}]$ are equally shared as proposals for every part of the given image. In this case, the number of proposals for each part is the same as the total number of proposals (*i.e*., $N = km$).

Considering the possibility of invisible or occluded parts, the total number of proposals might be less than N.

Compared with our top-down part region proposal method, the bottom-up methods, (*e.g*., selective search [36]), use no prior information. They can propose regions everywhere in a given image without any part geometric constraints. Moreover, our approach does not generate part regions through low-level texture or color image features, since those features may not be reliable for small semantic parts or part regions without distinct boundaries. In addition, since the values for m and k are usually very small (*e.g*., $m \leq 10$, $k \leq 20$), the number of our part proposals is about one order of magnitude less than that of the traditional region proposals (*e.g*., 200 vs 2000).

3.1.2 Fast RCNN based Part Detection

Given the k-NN part proposals, our detection network (DET-NET) applies Fast RCNN [13] to regress each proposed part region and assigns a part label. As each object has m parts, the DET-NET has $(m+1)$ way output, including m part labels and one background label as 0. Each way of the output contains one regressed bounding box, b, and a confidence score, $s \in [0, 1]$. As in Fast RCNN [13], we

train the part classifier and part regressor jointly by optimizing the multi-task loss L.

$$L\left(s, b, c, b^{gt}\right) = L_{cls}\left(s, c\right) + \lambda\left[c > 0\right] L_{loc}\left(b^{c}, b^{gt}\right) \quad (1)$$

in which $c \in [0, m]$ is the ground truth class for the input part bounding box; $L_{cls}\left(s, c\right)$ is the log loss for the true class; L_{loc} is the loss for part bounding box regression, where b^c is the regressed bounding box for the true class and b^{gt} is the ground truth box for the input part. More details about L_{cls} and L_{loc} can be referred to [13].

We classify all the part region proposals in parallel, and use a simple post-processing strategy to filter the results. We first assume that each object part could have at most one detection in the test image. Thus for each part, only the part bounding box with the highest confident score is chosen, indicated by $\{b^*, s^*\}$. We then remove the detections with confidence scores lower than a threshold, which indicates the corresponding parts are actually invisible (*e.g.*, the leg detection in Figure 1). In this paper, we set the threshold on the confidence score to be the probability of random guess, $1/(m+1)$.

3.2. Part Abstraction and Classification

Our part abstraction and classification sub-network (CLS-NET) introduces a semantic part RoI pooling layer, a part-based fully connected layer (pfc) and a concatenation fully connected layer (cfc) to the traditional CNN architecture to adjust it to be an end-to-end framework for fine-grained classification. The semantic part RoI pooling layer is devoted to extracting features only from the semantic object parts detected by the detection sub-network, and re-organizing them in a pre-defined order. The pfc layer only allows connections inside the same part in order to abstract mid-level part-specific features. A cfc layer is used to combine the pfc layers for all parts to enable an end-to-end training for all parts together in one network. The other convolutional (conv) and fully connected (fc) layers are the same as those in [27]. Figure 1 shows the details of our part abstraction and classification network. We also explain the details of these above-mentioned layers next.

3.2.1 Semantic Part RoI Pooling Layer

In the traditional CNN architecture, the pooling layer is used to increase the translation invariance and reduce the spatial size of the network. So the same pooling operation (*e.g.*, max pooling) is applied everywhere in the feature map. However, this "blind-mind" pooling strategy ignores the fact that not all the features in the feature map are useful for classification. Given that features from semantic parts of an object are more valuable for classification, we propose a part RoI pooling layer which is "clever" enough to conduct pooling just from the semantic parts of the object.

The proposed layer has two operations, pooling and re-organizing. First, based on the results from detection (during testing) or ground truth (during training), the part RoI pooling layer does semantic pooling. 1) Each part region is divided into $H \times W$ (*e.g.*, 3×3) sub-windows and then max-pooling is applied to each sub-window. A similar strategy was used in methods [13, 19]. 2) Features that do not lie within the semantic parts of the object are just discarded.

Then the pooled features from different parts are re-organized in a pre-defined order (*e.g.*, head, belly, back, ...). This process can also be viewed as part alignment, which is useful for fine-grained classification.

Note that this is different from the RoI pooling in [13], because region proposals in the RoI pooling do not have an order and are evaluated separately in later steps. Their RoI pooling is just a way to reduce computational cost by sharing the convolutional filters. In contrast, our semantic part RoI pooling layer conducts feature selection and re-ordering, which are useful for the final classification.

3.2.2 Part-based Fully Connected Layer

Considering that the performance of fine-grained recognition heavily relies on the features in object parts, we propose to directly add a part-based fully connected layer (pfc) in CNN to model mid-level part information for fine-grained classification. Each node in the pfc layer is only allowed to connect nodes which are from the same part of the object.

$$\mathbf{y}_i = f\left(\mathbf{W}_i \mathbf{x}_i\right), i = 1, 2, ...m \quad (2)$$

where $\mathbf{x}_i$ are the input features in part i, $\mathbf{y}_i$ are the output features of part i in the pfc layer, $\mathbf{W}_i$ are the weight parameters set for part i. Note that $\mathbf{W}_i$ are unique for each part to enforce the network to learn part specific features.

Compared to the fully connected layer, the main advantage of this pfc layer is that it cuts the redundant interactions of nodes in different parts and focuses on modeling the part features in the mid-level. It bridges the gap between low-level image features and high-level holistic information. Moreover, the number of parameters in this layer is much smaller than that of the fully connected layer given the same size input, which is also a desirable property in a large neural network.

Note that our pfc layer is different from other works [17, 20, 35], where the local convolutional filters are utilized to specified local regions. First, the convolutional filters in their works are applied to the same spatial location rather than the same parts, thus they are less applicable to objects with different poses since parts are not necessarily at the same location in different images. Second, for each part, we still use the same convolutional filters to capture the low-level image features. In our case, only the mid-level pfc layer discriminates the variation among different parts. To the best of our knowledge, we are the first to propose adding a pfc layer in CNN for mid-level part abstraction.

3.2.3 Concatenation Fully Connected Layer

Note that most previous part-based CNN approaches [41, 5, 28] train a separate CNN network for each part and concatenate the CNN features extracted for each part and then train a SVM on this concatenated feature vector. Here we

propose to use a concatenation fully connected (cfc) layer to build an integrated network dealing with different parts for fine-grained classification. This allows the propagation of classification error to all the parts and, hence, adjusting the part weights during training. The nodes in this layer connect to all the nodes in pfc layers. Thus this layer models the interactions among the nodes in different parts.

$$\mathbf{y} = f\left(\sum_{i=1}^{m} \mathbf{W}_i \mathbf{x}_i\right) \tag{3}$$

where $\mathbf{x}_i$ are the input features from part i, $\mathbf{W}_i$ are the weight parameters connecting with part i, $\mathbf{y}$ are the output features in this layer. During the training procedure, the connection weights $\mathbf{W}_i$ are adjusted to assign relative importance to different parts.

Compared with previous works' CNN-SVM scheme, our network can be trained and tested end-to-end for different parts in one stage. No extra storage is needed for feature caching in our network. Further, while the CNN-SVM scheme ignores the fact that different parts contribute differently in the classification, in our network, the cfc layer learns the relative importance of different parts for the recognition task.

3.3. Unifying Two Sub-networks

So far we have discussed the structure in our detection sub-network and classification sub-network. These two sub-networks can be trained and tested independently for the corresponding tasks. However, we want to build a unified network instead of having two separate networks. One additional motivation is that in one unified network, the convolution computation can be shared thus reduce significantly the computational cost. Some other recent works [8, 32, 33] have explored the same idea in object detection and semantic segmentation.

To unify the two sub-networks, we follow a similar idea from [32], using alternating optimization. Our 3-step training algorithm is as follows: First, the detection sub-network (DET-NET) and classification sub-network (CLS-NET) are trained for the corresponding task, respectively. Initialized with the ImageNet pre-trained model, these two sub-networks are fine-tuned end-to-end independently. For training the CLS-NET, the oracle part annotations instead of part detection results are used. At this point, these two sub-networks still have different conv layers . Second, we use the first n conv layers of CLS-NET to replace the corresponding layers in DET-NET, and then fine-tune all the other unique layers in DET-NET. Here n is a hyper-parameter, which plays a trade-off between accuracy and efficiency for the unified network. In the last step, using the part detections from DET-NET, we fine-tune all the other layers in CLS-NET except the shared conv layers. Therefore, these two sub-networks will have the same conv layers and thus form a unified network.

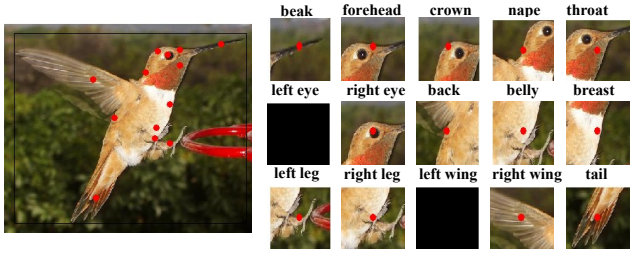

Figure 2. Parts illustration (Left: bird in the $W \times H$ bounding box with part centers marked by circles; Right: $\frac{1}{4}W \times \frac{1}{4}H$ region for each part. Black regions represent invisible parts.)

4. Experiments

Datasets: we evaluate our method on the well-known fine-grained benchmark birds dataset, CUB-2011 [37]. It has 200 bird species, with high degrees of similarity among some categories. Each category contains about 60 images with oracle object bounding boxes. Just as several previous works [16, 28], we will use these bounding boxes for both training and testing. Each image also provides the oracle annotations for 15 part centers. Similar to [16], we set each part region to be the size of $\frac{1}{4}W \times \frac{1}{4}H$ where W and H indicates the width and height of the object bounding box, respectively. As Figure 2 shows, regions of beak, forehead, crown, nape, throat, left and right eyes are highly overlapping. Thus, to avoid the duplicate usage of those regions, we define the union of all these seven part regions to be a grouped part, called head. Also the pair of legs and the pair of wings are symmetrical, so we assign an identical label to each pair. Consequently, we have seven parts for the bird in the order of head, back, belly, breast, leg, wing and tail.

Implementation Details: Our network is built on the open-source package Caffe [22]. CaffeNet (a variant of AlexNet [27]) is by default used for the initialization of both detection and classification sub-networks. The aspect ratio of the input image is kept unchanged and then either the shortest length is scaled to 600 or the longest length is scaled to 800. In the classification network, for each part, the part RoI pooling size is 3×3 and the number of nodes in the pfc layer is 512 for each part. The number of nodes in the cfc layer is kept as 4096. Out of the 5 conv layers, the two sub-networks share the first 3 conv layers in order to achieve the best trade-off between accuracy and efficiency. Random flip, crop and rotation are added as data augmentation for training the network.

4.1. Part Detection Results on CUB-2011

To evaluate our part detection sub-network for small semantic object part detection, we first investigate the hyper-parameters of our k-NN proposal method and then compare our detection results with the state-of-the-art works [28, 41, 2]. For all experiments, part detection is considered correct if it has at least 0.5 overlap with ground truth.

Hyper-parameter k. Table 2 lists detection results of our k-NN proposal method with different k values. It indi-

Parts	Head	Back	Belly	Breast	Leg	Wing	Tail	mAP
MCG [1]	90.58%	43.36%	34.23%	34.43%	53.44%	51.72%	51.25%	51.29%
Edge box [43]	90.54%	35.66%	48.61%	50.08%	66.28%	53.03%	43.28%	55.35%
selective search [36]	90.80%	56.07%	50.98%	51.79%	66.26%	62.09%	63.87%	63.12%
Ours	90.87%	75.88%	63.16%	67.46%	79.69%	64.79%	67.17%	72.72%

Table 1. Comparison of our k-NN proposal and bottom-up proposal methods for small semantic parts detection by average precision.

k	1	5	10	15	20
mAP	52.83%	70.96%	71.45%	72.70%	72.72%

Table 2. Comparison of our k-NN proposal method with different k values by the mean average precision (mAP) of all seven parts.

cates that we can improve the overall performance for all parts by increasing the k value up to 20. From $k = 15$ to $k = 20$, the incremental value becomes very small. So we set 20 as the default k value of our k-NN proposal method for all other experiments.

Strong prior or weak prior. Compared to the strong prior, our k-NN proposal method with weak prior gives slightly higher recall, because it is not restricted to the prior part class label inherited from the nearest neighbors, and more candidate regions are proposed for each part. The mean average precision (mAP) of the weak prior improves that of the strong prior from 71.79% to 72.72%, see Table 2. So we will use the weak prior in all our other experiments by default.

Comparison with bottom-up proposal methods. To evaluate the effectiveness of our top-down proposal method for small semantic parts detection, we compare it with several well known bottom-up methods [36, 43, 1] with the same metric. As shown in Table 1, our k-NN proposal method achieves 72.72% mAP that significantly outperforms all baseline methods. For example, our method gives a 9.6% higher mAP than our best baseline, the selective search method [36]. Results for each part further indicate that our method is much more accurate for proposing small semantic part regions (such as "back" and "leg"), compared with all baseline methods, *e.g.*, we achieves a 19.48% higher average precision than the selective search method for the part "back". Figure 3 shows example detection results of our method and the best baseline (selective search [36]). It indicates that part detections from our k-NN proposals have more accurate locations and more precise shapes than the detections from the selective search method. It qualitatively demonstrates that our k-NN proposal method plays a crucial role for small semantic parts detection. With respect to efficiency, the selective search method proposes an average number of 1270 regions for each image while our 20-NN proposal method only generates fewer than 180 proposals. Note that, to have faster speed and fewer false positives it is extremely important to have fewer proposals. In conclusion, our top-down k-NN proposal method is more efficient and more effective than bottom-up methods for small semantic parts detection.

Comparison with other state-of-the-art methods. Table 3 shows the comparison result of our part detection network (DET-NET) and the previous works [28, 41, 2] by Percentage of Correctly Localized Parts (PCP). By using the exactly same 2-part annotations as all the baseline methods, our part detection network achieves the best overall performance. Especially, for the relatively smaller part, head, our DET-NET outperforms the previous best method (LAC) by 19.4% PCP. We believe that our DET-NET achieves the significant improvement on small semantic part detection for two reasons. One is that our k-NN proposals inherit priors from the nearest neighbors in the training data, so that many promising small semantic part candidates are proposed. The other reason is that the fast RCNN integrated in our DET-NET performs the region regression to calculate more accurate part locations. In conclusion, our final part detection result outperforms all the previous works with a large margin on the CUB-2011 dataset [37].

Methods	Head	Body
Strong DPM [41, 2]	43.49%	75.15%
Selective search [41, 36]	68.19%	79.82%
LAC [28]	74.00%	96.00%
Ours	93.40%	94.93%

Table 3. Comparison with previous works by Percentage of Correctly Localized Parts (PCP) on CUB-2011. (To fairly compare, we use exactly the same 2-part annotations for all methods.)

4.2. Classification Results on CUB-2011

In this section, we evaluate the effectiveness of our part-abstraction and classification sub-network (CLS-NET), especially the proposed part-based fully connected (pfc) layer. A set of experiments are conducted to decide how many pfc layers and how many parts are the best to use in our CLS-NET. Here we use the oracle part annotations to avoid any influence from the part detection sub-network.

The experimental results are shown in Table 4. Compared with other alternative settings (Row 1-4) under exactly the same condition, the 7-part CLS-NET with 1pfc (Row 5) performs the best. We can draw several insights from the comparison results. First, it is very important to build part layers (*e.g.*, pfc layer) in the CNN framework to abstract and concatenate multiple parts; and it is important to use more semantic parts (67.02% vs 77.08% vs 79.46%). Second, one pfc layer is sufficient for part feature abstraction on this dataset (79.10 vs 79.46%). Last but not the least, although the whole object has been shown useful in previous fine-grained classification [16, 28, 41], it is not needed anymore when more small semantic parts of the object are

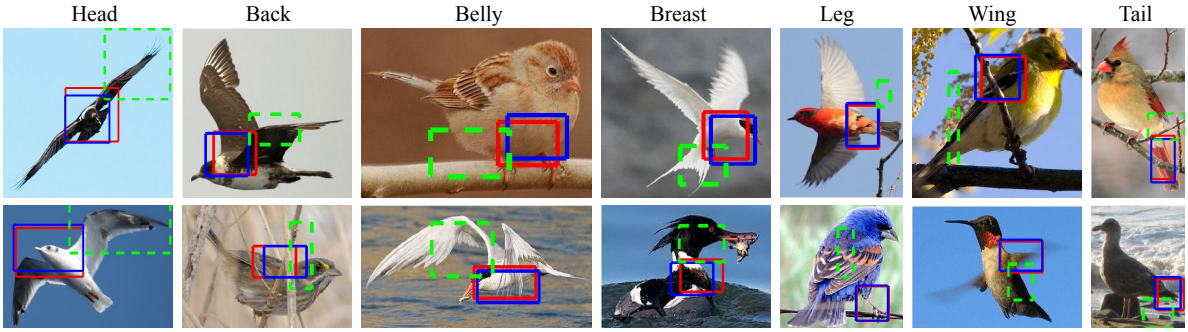

Figure 3. Detections failed by selective search (green boxes) but succeeded by our k-NN proposal method (blue). Red are ground truth.

Row	CLS-NET	Acc(%)
1	Object only (no pfc)	67.02
2	2-part with 1pfc	77.08
3	7-part with 2pfc	79.10
4	7-part+object with 1pfc	78.17
5	7-part with 1pfc	79.46
6	Ensemble of Row 1-5	81.96
7	7-part with 1pfc, VGGNet	84.69
8	Ensemble, VGGNet	85.71

Table 4. Comparison of different settings of our CLS-NET on CUB-2011 with oracle part annotations. *Ensemble, VGGNet* indicates the ensemble of 7-part with 1pfc and 7-part with 2pfc.

integrated in our framework (78.17% vs 79.46%).

We can further improve our best accuracy on CaffeNet to 81.96% using the ensemble of all five models with different settings. To test the generalization of our method, the results of models initialized by VGGNet are listed in Table 4. The final accuracy is boosted to 85.71%.

In conclusion, our 7-part CLS-NET with 1pfc layer is the best setting for CaffeNet on the CUB-2011. We will use it as the default setting in all other experiments.

4.3. Classification Results of Our Unified Network

This subsection gives the overall performance of our SPDA-CNN by feeding the seven semantic parts detected by the detection sub-network to the part-abstraction and classification sub-network. We compare the proposed SPDA-CNN with state-of-the-art previous works on CUB-2011 and CUB-2010, respectively.

CUB-2011: By directly using the model trained with oracle part annotations to classify test images using parts detected by the DET-NET, we achieves 78.15% accuracy (Table 5) which is only 1.31% lower than the accuracy of classifying test images using oracle part annotations (Row 5 in Table 4). After fine-tuning the model by training with part detections, as shown in Table 5, the gap becomes even smaller. This shows that our SPDA-CNN is robust to the part detection results. Moreover, our DET-NET is able to provide very good part detection results, as demonstrated in subsection 4.1. The comparison results in Table 5 show that our method performs much better than previous part-based methods, including fully supervised methods [3, 16, 7, 42, 41, 28] and methods without part annotations [34, 25]. Compared with state-of-the-arts [29, 21], our method also achieves slightly better performance. Moreover, with more supervision our method can explicitly detect small semantic parts and learn part-specific features.

Net	Train	Test	Methods	Acc(%)
n/a	BBox +Parts	BBox	Berg *et al.* [3]	56.89
			Göring *et al.* [16]	57.84
			Chai *et al.* [7]	59.40
			Zhang *et al.* [42]	64.96
Caffe	BBox +Parts	BBox	Zhang *et al.* [41]	76.37
			Lin *et al.* [28]	80.26
			Ours	78.15
			Ours+ft	78.93
			Our ensemble+ft	**81.01**
VGG	n/a	n/a	Simon *et al.* [34]	81.01
VGG	BBox	BBox	Krause *et al.* [25]	82.80
VGG	n/a	n/a	Lin *et al.* [29]	84.10
STN	n/a	n/a	Jaderberg *et al.* [21]	84.10
VGG	BBox +Parts	BBox	Ours+ft	84.55
			Our ensemble+ft	**85.14**

Table 5. Comparison with state-of-the-art methods on CUB-2011. *+ft* means fine-tuning the model using the part detections of training images; *Our ensemble* models for CaffeNet and VGGNet are same models shown in Table 4;

Methods	Accuracy
Göring *et al.* [16]	35.94%
Chai *et al.* [7]	47.30%
Lin *et al.* [28]	65.25%
Ours	**66.14%**

Table 6. Comparison with state-of-the-art on CUB-2010.

CUB2010: We also evaluate the generalization of our method on CUB-2010 [38], which does not provide oracle part annotations. We use the same part detection sub-network (DET-NET) trained on CUB-2011, but re-train the classification sub-network (CLS-NET) by only using data from CUB-2010. Part detections from DET-NET are used in both training and testing procedures of CLS-NET. The comparison results of our method with the previous works [16, 7, 28] in Table 6 illustrate that we achieve the state-of-the-art accuracy on CUB-2010, indicating the proposed method can be well generalized to other datasets.

features	spotted	malar	crested	masked	pattern	eyebrow	eyering	plain	eyeline	striped	capped	mean
head	0.81	0.64	0.83	0.72	0.71	0.75	0.63	0.74	0.67	0.77	0.71	0.73
object	0.75	0.63	0.75	0.70	0.67	0.73	0.62	0.71	0.64	0.75	0.67	0.69
other parts	0.70	0.60	0.69	0.62	0.61	0.69	0.58	0.66	0.60	0.72	0.64	0.65
head+object	0.81	0.62	0.81	0.70	0.70	0.73	0.62	0.73	0.65	0.76	0.68	0.71
others+object	0.73	0.61	0.76	0.67	0.65	0.70	0.59	0.69	0.62	0.72	0.65	0.67

Table 7. Area Under the Curve (AUC) for head attribute prediction using different part features in the pfc layer. Object means the whole object and other parts means all the other 6 parts except head.

4.4. Discussion

By analyzing our part detection and part-based classification results, we have an interesting observation. That is, our classification performance based on the 7-part detections is nearly as good as the performance based on the ground truth part annotations, however, the mAP of our 7-part detections is only 72.72% at the default 0.5 overlap threshold. As shown in Figure 4, decreasing the overlap threshold will increase the PCP for every part. For example, by decreasing the threshold to 0.2, we can achieve a 88.24% overall precision (i.e., object PCP) at 94.07% recall. The mAP for all parts is increased from 72.72% to 88.75%. In particular, we can achieve much higher PCPs for the parts whose exact regions are ambiguous. Thus we conjecture that it might be safer to have a smaller overlap threshold for the part detection task, compared with general object detection tasks. As shown in Figure 5, although some detections have less than 0.5 overlaps with the ground truth, they are visually correct. To be consistent with previous works [28, 41, 2], we use 0.5 as the default overlap threshold for all our experiments. However, this observation indicates that our detection results are actually more accurate than what is shown by its mAP at the 0.5 overlap threshold. This might be the reason why our classification performance based on the part detections is nearly as good as the performance based on the ground truth annotations.

To investigate whether the added part-based fully connected (pfc) layer actually learns part-specific features, we have done another experiment. We directly use features in the pfc layer to predict attributes in the parts. As shown in Table 7, the features from head are consistently better in predicting head attributes compared to features from the whole object or from the other parts. Also, adding features of the whole body does not improve the result, which proves the features in the pfc layer are part-specific and perform better for part-related tasks. This experiment also shows attribute prediction can be one potential application of the proposed network and we will investigate it in the future.

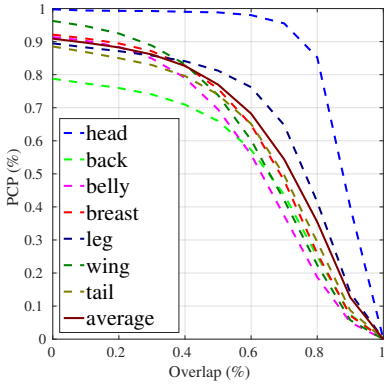

Figure 4. PCP of part detection network at different overlaps

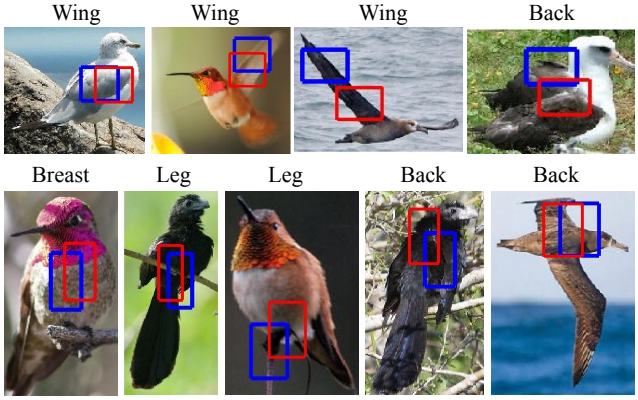

Figure 5. Detections (blue) that have less than 0.5 overlaps with the ground truth annotations (red), but are visually correct.

5. Conclusions

We have presented an end-to-end network (SPDA-CNN) that performs detection, multiple part localization, and recognition within one framework for fine-grained classification. The proposed network has two sub-networks for detection and recognition, respectively. The detection sub-network outperforms previous state-of-the-art methods with a large margin for small semantic parts detection, because of the proposed top-down part region proposal method. The classification sub-network introduces novel part layers which can learn discriminative part-specific features. This part-specific learning representation opens the door for a deeper understanding of fine-grained categories beyond just recognizing the class label.

6. Acknowledgements

This research is partially funded by National Science Foundation (NSF) under NSF-USA award #1069258 and NSF-USA award #1409683, and the research contract HHSN276201000693P from the Lister Hill Center for Biomedical Communications (LHNCBC) and the National Library of Medicine (NLM).

References

[1] P. A. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marqués, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 3, 6

[2] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*, 2012. 5, 6, 8

[3] T. Berg and P. N. Belhumeur. POOF: part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013. 7

[4] L. D. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 2

[5] S. Branson, G. Van Horn, P. Perona, and S. Belongie. Improved bird species recognition using pose normalized deep convolutional nets. In *BMVC*, 2014. 1, 3, 4

[6] J. Carreira and C. Sminchisescu. CPMC: automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 2012. 3

[7] Y. Chai, V. S. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013. 2, 7

[8] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 5

[9] J. Deng, J. Krause, and F. Li. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, 2013. 1

[10] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011. 2

[11] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010. 2

[12] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013. 2, 3

[13] R. B. Girshick. Fast R-CNN. In *ICCV*, 2015. 2, 3, 4

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

[15] G. Gkioxari, R. Girshick, and J. Malik. Actions and attributes from wholes and parts. In *ICCV*, 2015. 3

[16] C. Göring, E. Rodner, A. Freytag, and J. Denzler. Nonparametric part transfer for fine-grained recognition. In *CVPR*, 2014. 2, 3, 5, 6, 7

[17] K. Gregor and Y. LeCun. Emergence of complex-like cells in a temporal product network with local receptive fields. *CoRR*, abs/1006.0448, 2010. 4

[18] B. Hariharan, P. A. Arbeláez, R. B. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 3

[19] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 2015. 4

[20] G. B. Huang, H. Lee, and E. G. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *CVPR*, 2012. 4

[21] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 7

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[23] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR Workshop*, 2011. 1

[24] J. Krause, T. Gebru, J. Deng, L. Li, and F. Li. Learning features and parts for fine-grained recognition. In *ICPR*, 2014. 3

[25] J. Krause, H. Jin, J. Yang, and F. Li. Fine-grained recognition without part annotations. In *CVPR*, 2015. 3, 7

[26] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV Workshop*, 2013. 1

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 4, 5

[28] D. Lin, X. Shen, C. Lu, and J. Jia. Deep LAC: Deep localization, alignment and classification for fine-grained recognition. In *CVPR*, 2015. 1, 3, 4, 5, 6, 7, 8

[29] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 2015. 7

[30] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis. Jointly optimizing 3d model fitting and fine-grained classification. In *ECCV*, 2014. 1

[31] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 1

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 5

[33] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 5

[34] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *ICCV*, 2015. 3, 7

[35] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 4

[36] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013. 2, 3, 6

[37] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. 1, 2, 5, 6

[38] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, 2010. 2, 7

[39] 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 *CVPR*, 2015. 1, 3

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

[41] N. Zhang, J. Donahue, R. B. Girshick, and T. Darrell. Part-based R-CNNs for fine-grained category detection. In *ECCV*, 2014. 1, 3, 4, 5, 6, 7, 8

[42] N. Zhang, R. Farrell, F. N. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013. 2, 7

[43] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 3, 6