Deep LSAC for Fine-Grained Recognition

Di Lin, *Member, IEEE*, Yi Wang, *Member, IEEE*, Lingyu Liang, *Member, IEEE*, Ping Li, *Member, IEEE*, and C. L. Philip Chen, *Fellow, IEEE*

***Abstract*— Fine-grained recognition emphasizes the identification of subtle differences among object categories given objects that appear in different shapes and poses. These variances should be reduced for reliable recognition. We propose a fine-grained recognition system that incorporates localization, segmentation, alignment, and classification in a unified deep neural network. The input to the classification module includes functions that enable backward-propagation (BP) in constructing the solver. Our major contribution is to propose a valve linkage function (VLF) for BP chaining and form our deep localization, segmentation, alignment, and classification (LSAC) system. The VLF can adaptively compromise errors of classification and alignment when training the LSAC model. It in turn helps to update the localization and segmentation. We evaluate our framework on two widely used fine-grained object data sets. The performance confirms the effectiveness of our LSAC system.**

***Index Terms*— Convolutional neural network (CNN), fine-grained recognition, object detection, pose alignment, semantic segmentation.**

I. Introduction

FINE-GRAINED object recognition aims to identify subcategory object classes. It is used to find subtle differences among animals [1]–[3], product brands [4], [5], and architectural styles [6]. Recognition systems [7]–[18] make use of deep convolutional neural networks (CNNs) [19]–[22] and in general perform well.

CNN's flexibility provides fine-grained recognition, while having much room to improve. A key challenge is that discriminative patterns (e.g., a bird head in bird species recognition) can appear in different locations and with rotation and scaling in the collected images. Although the research of [23], [24] has shown that CNN features are reasonably robust in scale and rotation variation, it is better to directly capture changes of poses to increase the recognition accuracy [7], [8], [25]–[29]. Existing solutions [7], [8], [10], [30] perform localization and alignment to reduce pose variance. This procedure is illustrated in Fig. 1 using solid arrows where parts are localized, aligned according to templates, and then fed into the classification neural network. Because all steps are processed independently and consecutively, any error arising during localization can influence alignment and classification.

Previously, we proposed a feedback-control framework [31] to backpropagate alignment and classification errors to localization to optimally update all states in iterations. This process is highlighted with dashed arrows in Fig. 1, which improves the fine-grained classification performance in our experiments. We further investigate the multistage cascade for fine-grained recognition. Initially, the localization component provides box-shaped areas. The bounding box inevitably includes irrelevant background. While such background may provide context that facilitates classification, such as a sea background might strengthen the confidence of a gull's presence, it may harm the downstream alignment process.

In contrast to our previous work, in this article, we use the segmentation subnetwork to provide the pixel-wise objectness map and resolve the above problem. Here, the motivation is to use the pixel-wise objectness map to eliminate the adverse effect of the background reasonably well, consequently improving alignment and classification. Furthermore,

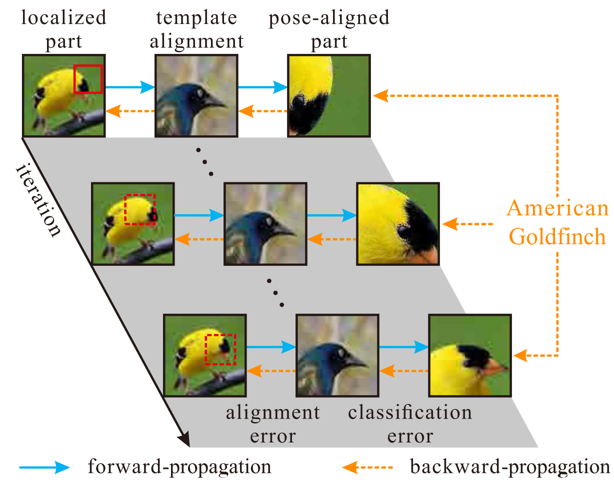

Fig. 1. One-way procedure from localization to template alignment makes each module rely on results from the previous one. BP highlighted by a dashed arrow makes it possible to refine the localization according to the classification and alignment results in the training phase, forming a bidirectional refinement process.

Manuscript received August 7, 2019; revised June 27, 2020; accepted September 18, 2020. Date of publication October 13, 2020; date of current version January 5, 2022. This work was supported in part by NSFC under Grant 61702338, Grant 61872151, and Grant 61701312; in part by the Natural Science Foundation of Guangdong Province under Grant 2019A1515011045, Grant 2020A1515011128, and Grant 2019A1515010847; in part by the Fundamental Research Funds for the Central Universities under Grant 2019MS023, in part by the National Key Research and Development Program of China under Grant 2019YFB1703600, and in part by The Hong Kong Polytechnic University under Grant P0030419 and Grant P0030929. *(Corresponding author: Ping Li.)*

Di Lin is with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China (e-mail: ande.lin1988@gmail.com).

Yi Wang is with the School of Biomedical Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: onewang@szu.edu.cn).

Lingyu Liang is with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: eelyliang@scut.edu.cn).

Ping Li is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: p.li@polyu.edu.hk).

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, also with the Navigation College, Dalian Maritime University, Dalian 116026, China, and also with the Faculty of Science and Technology, University of Macau, Macau 999078, China (e-mail: philip.chen@ieee.org).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3027603

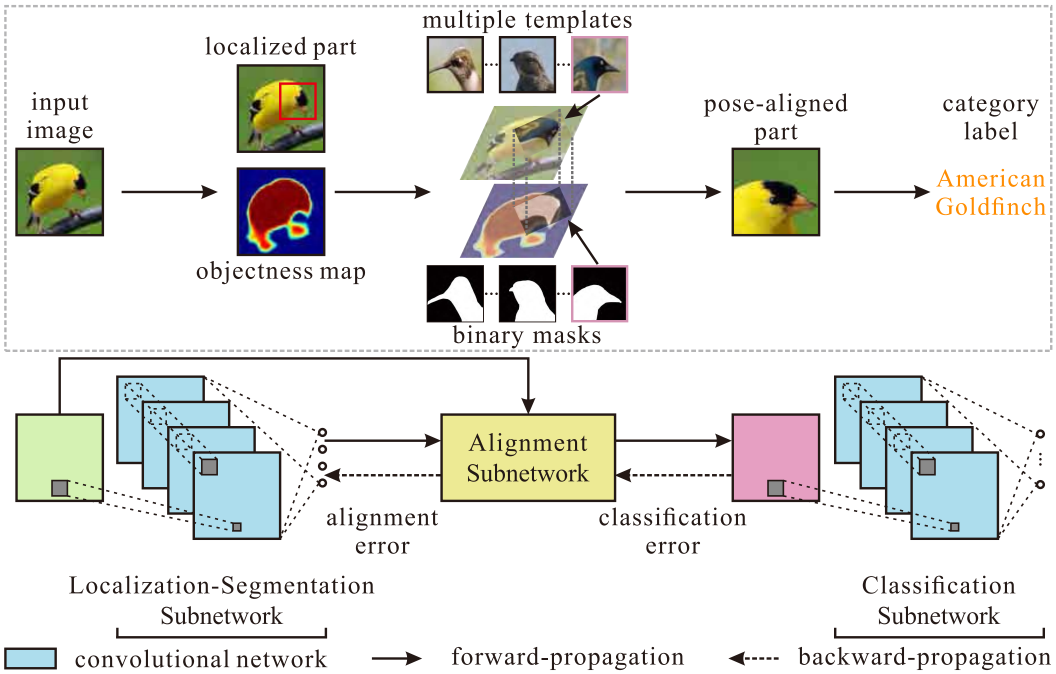

Fig. 2. Deep LSAC consists of the LSAC subnetworks. With the help of VLF, the alignment subnetwork outputs pose-aligned part images for classification in the FP stage, while the classification and alignment errors are propagated back to the localization in the BP stage.

the pixel-wise objectness map represents the accurate shapes of the object parts. Thus, we harness the useful shape information to guide the network learning. The whole framework is constructed as one deep neural network including localization, segmentation, alignment, and classification (LSAC) tasks; see the architecture in Fig. 2.

The difficulty of forming the unified neural network stems from the special requirement of the classification input. As shown in Fig. 1, the input to classification is an image after alignment. It cannot achieve the backward-propagation (BP) chain during training because the derivative of a constant, which is the aligned region, is zero. It means that the up-stream subnetworks miss the opportunity to update the network parameters. To avail an end-to-end network with all modules, we make three contributions to satisfy the conditions: 1) the input to the classification network is an aligned image; and 2) the gradients from the classification network can be back-propagated to better update the upstream networks.

A. Valve Linkage Function

We propose a differentiable valve linkage function (VLF) to form the alignment subnetwork. In our deep network, VLF plays a pivotal part in connecting the localization, segmentation, and classification modules of our deep LSAC network. VLF allows the pose-aligned part image to be directly passed into the classification network. In BP, VLF plays as a function containing necessary parameters for updating the localization and segmentation subnetworks. If alignment is good enough in the forward-propagation (FP) stage, VLF correspondingly guarantees an accurate classification. Otherwise, errors propagated from classification finely tune the localization and segmentation modules to improve the reliability of alignment. These effects cause the whole network to reach a stable state.

B. Localization-Segmentation Subnetwork

We use a localization-segmentation subnetwork to simultaneously regress a part-level location and object-level confidence map. As shown in Fig. 2, localization and segmentation share base CNN parameters. Different from methods [31]–[33] using separate segmentation and part-localization modules, our approach captures a relatively stable relationship between a fine-grained object (e.g., a bird) and part regions (e.g., bird heads and torsos). In addition, good localization and segmentation information forms a shape prior to guide part alignment. Some cluttered background can be filtered out thanks to the object-segmentation mask.

C. Multitemplate Alignment Subnetwork

Finally, we contribute an alignment subnetwork based on the localization and segmentation results. This subnetwork is equipped with a multitemplate selection to reduce the pose variance for the classification task. As illustrated in Fig. 2, the alignment subnetwork selects the best template and the associated binary mask, which are used for matching the shape of the part in the image. This is done by minimizing the cost of matching the shapes of the template and the part. Here, we use the segmentation result (i.e., the objectness map) to remove the background image region, thus eliminating the effect of the irrelevant background on matching the shapes. It allows our joint modeling LSAC perform better than other methods that neglect the rich shape information of object templates.

We summarize the major contributions of this article as:

1) We propose VLF to connect the LSAC components as a unified network. During the learning stage, VLF can reasonably control the BP signal to better coordinate the training of all network components.
2) We present a new feature of the localization-segmentation component by sharing CNN parameters. This component effectively learns the shape information of the object to improve the alignment of object parts.

3) We use a flexible strategy of multitemplate selection to reduce the pose variance of the object parts. This strategy remarkably improves the classification accuracy.

In Section II, we revisit related work on fine-grained recognition. In Sections III–V, we elaborate on our deep LSAC model and its implementation details. In Section VI, we study our deep LSAC model by conducting ablation studies and comparison experiments. We provide our conclusions in Section VII.

II. Related Work

Unlike generic object classification [34], [35], subordinate categories (e.g., gull and sparrow) are the recognition targets in fine-grained tasks. We review recognition systems to discover the subtle distinctions among subordinate categories.

A. Holistic Representation

Early work focused on constructing discriminative whole-image representation [1]–[3]. Fine-grained data sets were proposed in [1] and [2]. The methods are based on the traditional pipeline of global feature construction and classification [36]–[38]. In [3], localization was exploited to find object parts for identification of dog breeds. This module was designed for dog faces only. Other parts (e.g., torso and tail) were not considered.

B. Pose Normalization

To recognize subtle part differences, later work used localization and alignment, which extract semantic parts from visually similar regions to reduce their variance, yielding pose-normalized representation.

Farrell *et al.* [39] used part templates to obtain a location. The templates were predefined (e.g., a bird head and torso) to use part annotations during training. The relationship among parts was not modeled. Zhang *et al.* [40] adapted the deformable part model [41], [42] to extract part regions and features as image representation. Similarly, Yao *et al.* [43] and Yang *et al.* [44] learned part models to localize important parts for fine-grained objects. The models were trained in data-driven ways to mine discriminative parts.

Fine-grained recognition benefits from the correspondence offered by localization. However, the localization operation does not consider rotation [39], [40], [43], [44]. Therefore, alignment is invoked as a complement. Gavves *et al.* [45] aligned the whole object to accommodate possibly large variation of poses. The alignment was accomplished by matching histograms of gradients (HOGs) [46] and fitting elliptical shapes. Berg and Belhumeur [47] proposed a part-based one-vs-one feature (POOF), where objects were divided into different groups for alignment. Xie *et al.* [48] performed hierarchical alignment, where small parts were aligned independently and assembled later for further processing. The methods of [47], [48] showed flexibility to handle pose variance.

Segmentation and localization were unified in [33], [49], and [9]. In these studies, the hand-designed prior plays an important role in modeling the relationship between objects and parts, which simultaneously benefits object-level segmentation and part-level localization. There have been several works [27], [50], [51] that depended on the deep neural networks to localize and segment the object parts. Simon *et al.* [50] employed the activation values of the convolutional feature maps to localize the discriminative part regions, where they extracted useful information to assist the classification task. Zheng *et al.* [27] built the hierarchical network architecture to capture the local and global relationship between object parts. Furthermore, Zheng *et al.* [51] employed the nonlocal attention model to localize the object, which is associated with the parts. The nonlocal attention model provides the pixel-level mask to lessen the distractions of the background.

Although localization, segmentation, and alignment were used to produce pose-normalized representation for fine-grained recognition [33], [39], [40], [47]–[49], [52], they worked without feedback. We introduce the end-to-end training of all modules and improve the classification performance.

C. Fine-Grained Recognition With CNN

Deep CNN has been used to achieve fine-grained recognition. It has been successfully applied to fundamental recognition tasks, such as image classification [19], [53], object detection [54]–[58], and semantic segmentation [59], [60]. CNN provides transferable knowledge by means of pretraining on large-scale image data [34], benefiting various vision tasks. Research into the use of a pretrained CNN [19], [20] in fine-grained recognition has produced compelling results.

CNN has been applied to part representation learning [7], [8], [40]. Zhang *et al.* [8] used selective search [61] for part proposals. Branson *et al.* [7] studied higher order geometric warping to align parts. Fine-tuned CNN models [19], [20] extracted representation on parts [7], [8]. Apart from learning representation, methods [10], [30] applied CNN to learn explicit part-based models for localization. Part localization and classification were integrated to allow end-to-end training. The localized parts in [10], [30], [62] increase semantic information of mid-level CNN features. In [9] and [63]–[67], an implicit part model is applied to eliminate the requirement of part annotations. Krause *et al.* [9] resorted to cosegmentation [68], [69] to discover semantic parts. Simon *et al.* [64], Lin *et al.* [63], and Kong *et al.* [66] selected meaningful network activations that capture the part information. He *et al.* [28], Zheng *et al.*, [27] and Yan *et al.* [29] employ the deep attention network to detect object parts. Krause *et al.* [67] investigated the use of abundant cost-effective data to replace expensive part annotations. They made remarkable improvements over fine-grained recognition. Jaderberg *et al.* [65] proposed a network to account for the spatial transformation for flexible localization of parts.

As a vital component to reduce the pose variance, alignment must be jointly modeled with other CNN-based components, for which the aforementioned methods have not considered yet. Furthermore, the existing methods perform the LSAC on high-level convolutional feature maps. Due to the stacked down-sample operations, the high-level feature maps are lack of the visual details, which are important for fine-grained recognition. In our framework, the alignment takes part in the end-to-end training of CNN and helps update the whole framework. Compared to previous works, our LSAC model enables the joint update of the low- and high-level information. It learns the high-level features for localization and segmentation. Then the localization and segmentation results assist the alignment manipulation on the image, selecting the useful low-level image information to refine the high-level features for better classification.

TABLE I
LIST OF SYMBOLS WITH DESCRIPTIONS

symbol	description	symbol	description
$\boldsymbol{c} \in \mathbb{R}^2$	coordinate in image plane	$\boldsymbol{R} \in \mathbb{R}^{h\times w\times 3}$	RGB part region
$\boldsymbol{c}^r \in \mathbb{R}^2$	center of a regressed bounding box	$\boldsymbol{t} \in \mathbb{R}^{h\times w\times 3}$	part template
$\boldsymbol{I} \in \mathbb{R}^{h\times w\times 3}$	RGB image	$\boldsymbol{t_m} \in \mathbb{R}^{h\times w}$	binary mask of a part template
$\boldsymbol{L} \in \mathbb{R}^4$	predicted bounding box	$y \in \mathbb{R}$	category label of an image
$\boldsymbol{L^{gt}} \in \mathbb{R}^4$	ground-truth bounding box	$\alpha \in \mathbb{R}$	scaling factor
$o \in \mathbb{R}$	predicted label of a pixel	$\theta \in \mathbb{R}$	rotation degree
$o^{gt} \in \mathbb{R}$	ground-truth label of a pixel	$\boldsymbol{W}_c$	parameters for classification
$\boldsymbol{O} \in \mathbb{R}^{h\times w}$	objectness map	$\boldsymbol{W}_l$	parameters for localization
$\boldsymbol{p} \in \mathbb{R}^{256}$	distribution of gray-scale values	$\boldsymbol{W}_s$	parameters for segmentation
$P \in [0,1]$	probability	$\boldsymbol{W}_{ls}$	parameters for localization and segmentation
$D(\boldsymbol{c},\boldsymbol{L}) \in \mathbb{R}$	distance energy	$E_c(\boldsymbol{W}_c;\boldsymbol{I}^*,y^{gt}) \in \mathbb{R}$	classification loss
$E_a(\boldsymbol{c},\theta,\alpha,\boldsymbol{t};\boldsymbol{I},\boldsymbol{L},\boldsymbol{O}) \in \mathbb{R}$	alignment energy	$E_l(\boldsymbol{W}_l;\boldsymbol{I},\boldsymbol{L}^{gt}) \in \mathbb{R}$	localization loss
$E_s(\boldsymbol{W}_s;\boldsymbol{I},o^{gt}) \in \mathbb{R}$	segmentation loss	$E_{ls}(\boldsymbol{W}_{ls};\boldsymbol{I},\boldsymbol{L}^{gt},o^{gt}) \in \mathbb{R}$	localization-segmentation loss
$f_c(\boldsymbol{W}_c;\boldsymbol{I}^*) \in \mathbb{R}$	classification result	$f_l(\boldsymbol{W}_l;\boldsymbol{I}) \in \mathbb{R}^4$	localization result
$F(\boldsymbol{O},\boldsymbol{t}_m) \in \mathbb{R}$	objectness confidence	$J(\boldsymbol{W}_c,\boldsymbol{W}_{ls};\boldsymbol{I},\boldsymbol{L}^{gt},y^{gt},o^{gt}) \in \mathbb{R}$	joint loss of the LSAC network
$S(\boldsymbol{R}_i,\boldsymbol{R}_j) \in \mathbb{R}$	pose similarity of two part regions	$V(\boldsymbol{L},\boldsymbol{O};\boldsymbol{I},\boldsymbol{L}_f,\boldsymbol{O}_f) \in \mathbb{R}^{h\times w\times 3}$	valve linkage function

III. DEEP LSAC MODEL

To recognize fine-grained classes, we learn deep LSAC models for distinct and meaningful parts. Features extracted from parts are used in classical classifiers, such as support vector machine (SVM). The main framework consists of four major tasks: 1) the localization module that provides part positions; 2) the segmentation module that yields the pixel-wise object region; 3) the template alignment that offsets translation, scaling, and rotation for pose-aligned parts; and 4) the final classification module, which takes its inputs from the previous task.

As mentioned above, the way to connect these four modules in a unified deep neural network is nontrivial. Below, we describe localization and segmentation, which share the same set of CNN parameters. Then we detail our alignment subnetwork, where FP and BP stages are implemented. For the convenient reference, we summarize the critical notations with their indications in Table I.

A. Localization-Segmentation Subnetwork

1) Localization: The localization module outputs the coordinates for the top-left and bottom-right bounding-box corners denoted as (x_1, y_1) and (x_2, y_2), given an input natural image for fine-grained recognition. In the training phase, we regress bounding boxes of part regions. Ground-truth bounding boxes are generated with part annotation. We unify input image resolution and construct a localization module based on CNN. The localization network includes the learnable parameters $\boldsymbol{W}_l$ and a predicted bounding-box $\boldsymbol{L}$.

Given the input image $\boldsymbol{I} \in \mathbb{R}^{h\times w\times 3}$ and the bounding-box $\boldsymbol{L} = (x_1, y_1, x_2, y_2)$, we express the localization subnetwork as

$$\boldsymbol{L} = f_l(\boldsymbol{W}_l; \boldsymbol{I}). \tag{1}$$

During training, the ground-truth location of parts $\boldsymbol{L}^{gt}$ is used. The location objective function is given by

$$E_l\big(\boldsymbol{W}_l; \boldsymbol{I}, \boldsymbol{L}^{gt}\big) = \frac{1}{2}||f_l(\boldsymbol{W}_l; \boldsymbol{I}) - \boldsymbol{L}^{gt}||^2. \tag{2}$$

We minimize it over $\boldsymbol{W}_l$.

2) Segmentation: The segmentation module produces pixel-wise scores for foreground objects and the background by preforming two-class regression.

We let $\boldsymbol{W}_s$ denote the learnable network parameters. Given a pixel at location $\boldsymbol{c}_i \in \mathbb{R}^2$, the segmentation network outputs the probability $P(o_i|\boldsymbol{c}_i, \boldsymbol{W}_s)$ that predicts the pixel to have either a foreground $(o_i = 1)$ or background $(o_i = 0)$ label. We are given N_{pixel} pixels, and each pixel at location $\boldsymbol{c}_i$ is given the ground-truth label $o_i^{gt} \in \{0, 1\}$. During training, the parameters are learned by minimizing the following objective function with network parameters $\boldsymbol{W}_s$:

$$E_s\big(\boldsymbol{W}_s; \boldsymbol{I}, o^{gt}\big) = -\frac{1}{N_{\text{pixel}}}\sum_i \log P\big(o_i = o_i^{gt}|\boldsymbol{c}_i, \boldsymbol{W}_s\big). \tag{3}$$

In our scenario, we use segmentation to generate an objectness map $\boldsymbol{O} \in \mathbb{R}^{h\times w}$, where

$$\boldsymbol{O}(\boldsymbol{c}_i) = P(o_i = 1|\boldsymbol{c}_i, \boldsymbol{W}_s) \tag{4}$$

predicts the probability of the pixel at location $\boldsymbol{c}_i$ to have a foreground label. A high probability means that the pixel is located inside an object region. The objectness map is an object-sensitive cue. It reduces the chance that the alignment is applied to the background. Along with the localization result, the objectness map can be used to refine the alignment. We provide more details in Section III-C below.

3) Parameter Sharing: Our localization and segmentation components account for two tasks–localization provides part regions, and segmentation regresses the whole object. Appearances of objects and part regions are generally stable in fine-grained tasks. The context can be used to benefit both tasks. Thus, we integrate part localization and object segmentation to capture their underlying correlation.

We tackle the joint localization-segmentation task by sharing parameters of CNN. We place the regression output of localization and segmentation at the end of the network. That is, the localization and segmentation share the base layers, that is, conv1_1 to $fc7$. The feature generated from the base layers is embedded by an extra fully connected layer to the part coordinates. The base feature is processed by convolutional manipulation to yield pixel-wise scores.

We let $\boldsymbol{W}_{ls}$ denote the parameters of the localization-segmentation subnetwork. With (2) and (3), we formulate the training objective of localization-segmentation as

$$E_{ls}\left(\boldsymbol{W}_{ls}; \boldsymbol{I}, \boldsymbol{L}^{gt}, o^{gt}\right) = \frac{1}{2}||f_l(\boldsymbol{W}_{ls}; \boldsymbol{I}) - \boldsymbol{L}^{gt}||^2 - \frac{1}{N_{\text{pixel}}}\sum_i \log P\left(o_i = o_i^{gt}|\boldsymbol{c}_i, \boldsymbol{W}_{ls}\right) \tag{5}$$

for which an implicit weight of 1 is used to balance between the localization and segmentation losses. We will show next that joint localization-segmentation yields superior performance compared to performing them independently.

B. Classification Subnetwork

The classification subnetwork is the last module shown in Fig. 2. Our classification takes the pose-aligned part image as input, denoted as $\boldsymbol{I}^* \in \mathbb{R}^{h\times w\times 3}$, and generates the category label. This classification CNN [19] is expressed as

$$y = f_c\left(\boldsymbol{W}_c; \boldsymbol{I}^*\right) \tag{6}$$

where $\boldsymbol{W}_c$ is the weight parameter set in this subnetwork. The output is the category label $y \in \mathbb{R}$.

During training, the ground-truth label y^{gt} is provided. We use the probability $P(y|\boldsymbol{I}^*, \boldsymbol{W}_c)$ to predict the pose-aligned part image $\boldsymbol{I}^*$ to have the category y. The predicted category y should be consistent with y^{gt}. We enforce a penalty on y by following [19], which is denoted as:

$$E_c\left(\boldsymbol{W}_c; \boldsymbol{I}^*, y^{gt}\right) = -\log P\left(y = y^{gt}|\boldsymbol{I}^*, \boldsymbol{W}_c\right). \tag{7}$$

Our major contribution in this system is the construction of the alignment subnetwork, which is detailed below together with the formulation of $\boldsymbol{I}^*$ in (6).

C. Alignment Subnetwork

The alignment subnetwork receives part location $\boldsymbol{L}$ (i.e., the bounding box) from the localization module and the objectness map $\boldsymbol{O}$ (i.e., the pixel-wise foreground probabilities) from the segmentation module. It then performs template alignment [70] and feeds a pose-aligned part image to classification, as shown in Fig. 2. Our alignment offsets translation, scaling, and rotation for pose-aligned part region generation, which is important for accurate classification. Apart from pose aligning, this subnetwork plays a crucial role in bridging the BP stage of LSAC model, which helps utilize the classification and alignment results to refine the localization-segmentation subnetwork.

We propose a new VLF as the output of the alignment subnetwork to accomplish the above goals. In what follows, we present our alignment part and then detail our VLF in line with the FP and BP stages of the LSAC model.

Template Alignment: We rectify the localized part regions, making their poses close to the templates. We define a function to evaluate the pose similarity between a pair of uniform-size part regions $\boldsymbol{R}_i, \boldsymbol{R}_j \in \mathbb{R}^{h\times w\times 3}$. To reduce illumination variance, we normalize the pixel values of each part region. We quantize the range of pixel values into 256 bins, and, respectively, compute the distributions, i.e., $\boldsymbol{p}_i, \boldsymbol{p}_j \in \mathbb{R}^{256}$, of gray-scale values of the part regions $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$. The normalization of the gray-scale values and the calculation of the distribution follow the construction of the normalized color histogram. As $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$ have the equal sizes, every two pixels having the same location of $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$ form a pair of gray-scale values. Using all the tuples, we calculate the

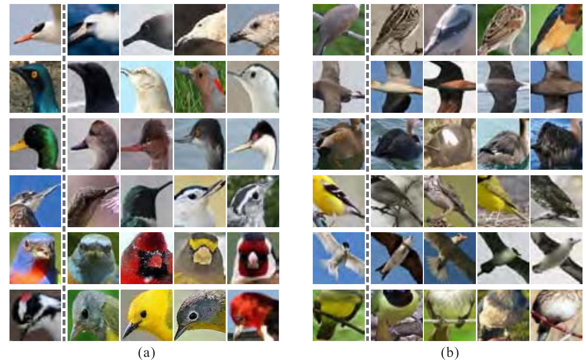

Fig. 3. Examples of (a) bird heads and (b) torsos. The alignment templates selected by the clustering algorithm are presented in the first columns of (a) and (b).

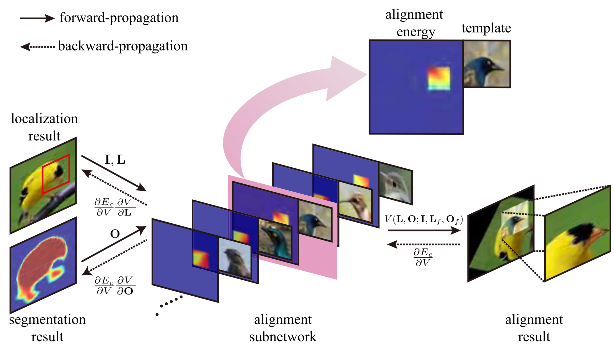

Fig. 4. Alignment subnetwork selects pose-aligned parts for classification. The solid arrows indicate the inputs passed forward. The dash arrows indicate the back-propagated gradients. In FP stage, the alignment subnetwork takes input as an image and the associated localization-segmentation results. In BP stage, it computes the gradients with respect to the localization-segmentation results.

joint distribution of gray-scale values of $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$, which is denoted as $\boldsymbol{p}_{ij} \in \mathbb{R}^{256\times 256}$. With the distributions $\boldsymbol{p}_i$, $\boldsymbol{p}_j$, and $\boldsymbol{p}_{ij}$, we define the similarity function as

$$S\left(\boldsymbol{R}_i, \boldsymbol{R}_j\right) = \sum_{m=1}^{256}\sum_{n=1}^{256} \boldsymbol{p}_{ij}(m,n)\log\left(\frac{\boldsymbol{p}_{ij}(m,n)}{\boldsymbol{p}_i(m)\boldsymbol{p}_j(n)}\right). \tag{8}$$

We note that this similarity function is based on mutual information [70]. A large value means similar poses between $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$.

To minimize large pose variation, we generate a template set for alignment. For each pair in N training part images, we calculate the similarity using (8) and finally form a similarity matrix $\boldsymbol{S}_t \in \mathbb{R}^{N\times N}$. $\boldsymbol{S}_t$ is then processed with spectral clustering [71] to split the N images into K clusters. For each cluster, we select the part region closest to the cluster center as template. To include mirrored poses, we flip each template. Eventually, we obtain a template set $\mathfrak{T}$. Examples of bird heads and torsos are shown in Fig. 3.

Fig. 4 shows the pipeline of alignment. Given an input image $\boldsymbol{I}$, we assume that the pose-aligned part region has center location $\boldsymbol{c}$, rotated by θ degree and scaled by factor α. To compare it with a template $\boldsymbol{t}$, we extract the region from image $\boldsymbol{I}$, denoted as $\boldsymbol{I}(\boldsymbol{c}, \theta, \alpha)$. Apart from applying the similarity function (8), we make the regressed-part bounding box $\boldsymbol{L}$, generated by the localization module, and a regularization is then defined as

$$D(\boldsymbol{c}, \boldsymbol{L}) = \exp\left(-\frac{\|\boldsymbol{c} - \boldsymbol{c}^r(\boldsymbol{L})\|^2}{2\sigma^2}\right) \tag{9}$$

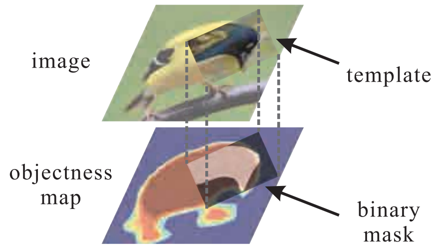

Fig. 5. Illustration to measure the foreground confidence of aligned parts. We register the binary mask to the objectness map with zero padding.

where σ is a constant set to 15 empirically. $\boldsymbol{c}^r(\boldsymbol{L}) = (x_1 + x_2/2, y_1 + y_2/2)$ represents the center of the bounding box $\boldsymbol{L}$. Using (9), we adjust the aligning center $\boldsymbol{c}$ according to the regressed center $\boldsymbol{c}^r(\boldsymbol{L})$ of parts. This makes alignment more reliable.

The similarity measurement of (8) is defined in terms of the distributions of pixel values, which offers little critical shape information of the object. By knowing the shape of the foreground object, the influence of a cluttered background can be reduced when we align the part region with the template. To this end, we measure the objectness confidence of the aligned part, which is covered by the template. Fig. 5 illustrates this process. We are given the binary mask $\boldsymbol{t}_m$ along with the template $\boldsymbol{t}$. We use the ground-truth data to generate the binary mask $\boldsymbol{t}_m$. $\boldsymbol{t}_m(\boldsymbol{c}_i) \in \{0, 1\}$ indicates that the pixel $\boldsymbol{c}_i$ belongs to the background or foreground. Given a pixel $\boldsymbol{c}_i$, we denote $\boldsymbol{O}_f(\boldsymbol{c}_i)$ and $\boldsymbol{O}_b(\boldsymbol{c}_i)$ as foreground and background scores, which are computed by the objectness probability as

$$\boldsymbol{O}_f(\boldsymbol{c}_i) = -\log(1 - \boldsymbol{O}(\boldsymbol{c}_i)), \quad \boldsymbol{O}_b(\boldsymbol{c}_i) = -\log \boldsymbol{O}(\boldsymbol{c}_i). \tag{10}$$

A high foreground/background score means the pixel is located inside the foreground/background region. Assume $\boldsymbol{t}_m$ has a total of N_{pixel} pixels, including N_f foreground and N_b background pixels. We define the objectness confidence as

$$F(\boldsymbol{O}, \boldsymbol{t}_m) = \frac{1}{N_f} \sum_{i=1}^{N_{\text{pixel}}} \boldsymbol{O}_f(\boldsymbol{c}_i)\boldsymbol{t}_m(\boldsymbol{c}_i) + \frac{1}{N_b} \sum_{i=1}^{N_{\text{pixel}}} \boldsymbol{O}_b(\boldsymbol{c}_i)(1 - \boldsymbol{t}_m(\boldsymbol{c}_i)). \tag{11}$$

We note that the binary mask $\boldsymbol{t}_m$ is registered to the image with zero padding. The confidence output by (11) encourages a part region with a high foreground probability to be located in the foreground region of the template, while suppressing the background region of the template that overlaps with the part region. With the guidance of objectness confidence, the part region can be better aligned with the template that has a similar shape.

Using (8), (9), and (11), we formulate the alignment process as finding the values of $\boldsymbol{c}$, θ, α, and $\boldsymbol{t}$ that maximize

$$\begin{aligned} &E_a(\boldsymbol{c}, \theta, \alpha, \boldsymbol{t}; \boldsymbol{I}, \boldsymbol{L}, \boldsymbol{O}) \\ &\quad = \lambda_a S(\boldsymbol{I}(\boldsymbol{c}, \theta, \alpha), \boldsymbol{t}) + \lambda_d D(\boldsymbol{c}, \boldsymbol{L}) + \lambda_s F(\boldsymbol{O}, \boldsymbol{t}_m) \\ &\quad\quad \boldsymbol{c} \in [x_1, x_2] \times [y_1, y_2], \quad \theta \in \Theta, \ \alpha \in \mathfrak{A}, \ \boldsymbol{t} \in \mathfrak{T} \end{aligned} \tag{12}$$

where λ_a, λ_d, and λ_s are constants set to 1, 0.001, and 0.0003, respectively. Θ, $\mathfrak{A}$, and $\mathfrak{T}$ define the ranges of parameters. The ranges of parameters defined in (12) are used by searching in the FP stage of the network to determine the parameters $\boldsymbol{c}$, θ, α, and $\boldsymbol{t}$. A large alignment energy output by (12) indicates reliable alignment. Maximizing the alignment energy is achieved by searching the quantized parameter space.

IV. Valve Linkage Function

We now detail the VLF, which is important to link the subnetworks and make them work as a whole in the training phase.

We denote the pose-aligned part as $\boldsymbol{I}(\boldsymbol{c}^*, \theta^*, \alpha^*) \in \mathbb{R}^{h \times w \times 3}$. In the FP stage, the outputs of the localization-segmentation subnetwork are represented as $\boldsymbol{L}_f \in \mathbb{R}^4$ and $\boldsymbol{O}_f \in \mathbb{R}^{h \times w}$. Our VLF is defined as

$$\begin{aligned} &V\big(\boldsymbol{L}, \boldsymbol{O}; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f\big) \\ &\quad = \frac{E_a(\boldsymbol{c}^*, \theta^*, \alpha^*, \boldsymbol{t}^*; \boldsymbol{I}, \boldsymbol{L}, \boldsymbol{O})}{E_a\big(\boldsymbol{c}^*, \theta^*, \alpha^*, \boldsymbol{t}^*; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f\big)} \boldsymbol{I}\big(\boldsymbol{c}^*, \theta^*, \alpha^*\big) \end{aligned} \tag{13}$$

where

$$\begin{aligned} &\big\{\boldsymbol{c}^*, \theta^*, \alpha^*, \boldsymbol{t}^*\big\} \\ &\quad = \arg\max_{\boldsymbol{c}, \theta, \alpha, \boldsymbol{t}} E_a\big(\boldsymbol{c}, \theta, \alpha, \boldsymbol{t}; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f\big) \\ &\quad\quad \text{s.t. } \boldsymbol{c} \in [x_1, x_2] \times [y_1, y_2], \quad \theta \in \Theta, \ \alpha \in \mathfrak{A}, \boldsymbol{t} \in \mathfrak{T}. \end{aligned} \tag{14}$$

The VLF comprises three critical terms: 1) the function of alignment energy $E_a(\boldsymbol{c}^*, \theta^*, \alpha^*, \boldsymbol{t}^*; \boldsymbol{I}, \boldsymbol{L}, \boldsymbol{O})$; 2) the alignment energy $E_a(\boldsymbol{c}^*, \theta^*, \alpha^*, \boldsymbol{t}^*; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f) \in \mathbb{R}$ that is computed by the forward-propagated part localization $\boldsymbol{L}_f$ and objectness map $\boldsymbol{O}_f$; and 3) the pose-aligned part $\boldsymbol{I}(\boldsymbol{c}^*, \theta^*, \alpha^*)$.

Our VLF satisfies the end-to-end training of the classification and localization-segmentation subnetwork. This is because our VLF allows the pose-aligned part to be directly fed into the classification subnetwork in FP, while passing the back-propagated gradients from the classification subnetwork to update the localization-segmentation subnetwork in BP. The role of VLF in FP and BP is discussed below.

A. FP Stage

In the FP stage, the alignment subnetwork receives part location $\boldsymbol{L}_f$ and objectness score map $\boldsymbol{O}_f$. As in (13), the function of alignment energy and the forward-propagated energy are in a ratio form, which always results in a factor of 1 after the FP stage. It makes the output of VLF

$$V\big(\boldsymbol{L}_f, \boldsymbol{O}_f; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f\big) = \boldsymbol{I}\big(\boldsymbol{c}^*, \theta^*, \alpha^*\big) \tag{15}$$

which is exactly the pose-aligned part. We note the ratio form allows the pose-aligned part to be the input of the classification subnetwork without changing the pixel values by a nonidentity transformation.

B. BP Stage

In (13), the VLF preserves the function of alignment energy, for which the variables of part location and objectness score map can be regarded as inputs. It enables updating of the signal of classification to be passed to the localization-segmentation subnetwork using the chain rule, as shown in Fig. 4.

In the BP stage, the output of the alignment subnetwork $V(\boldsymbol{L}, \boldsymbol{O}; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f)$ becomes a function of $\boldsymbol{L}$ and $\boldsymbol{O}$. Therefore, the objective function of LSAC is formulated as

$$\begin{aligned} &J\big(\boldsymbol{W}_c, \boldsymbol{W}_{ls}; \boldsymbol{I}, \boldsymbol{L}^{gt}, y^{gt}, o^{gt}\big) \\ &\quad = E_c\big(\boldsymbol{W}_c; V\big(\boldsymbol{L}, \boldsymbol{O}; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f\big), y^{gt}\big) + E_{ls}\big(\boldsymbol{W}_{ls}; \boldsymbol{I}, \boldsymbol{L}^{gt}, o^{gt}\big). \end{aligned} \tag{16}$$

This training objective balances the loss E_c for the classification subnetwork and E_{ls} for the localization-segmentation subnetwork, where $\boldsymbol{W}_c$ and $\boldsymbol{W}_{ls}$ are the network parameters to be determined. The losses output by E_c and E_{ls} are balanced by a factor of 1 during training. We minimize this objective function for updating localization-segmentation and classification subnetworks during training.

To update the classification subnetwork, we compute the gradients of J with respect to $\boldsymbol{W}_c$. To update the localization-segmentation subnetwork, gradients with respect to $\boldsymbol{W}_{ls}$ are computed as

$$\nabla_{\boldsymbol{W}_{ls}} J = \frac{\partial E_{ls}}{\partial \boldsymbol{W}_{ls}} + \frac{\partial E_c}{\partial \boldsymbol{W}_{ls}} \tag{17}$$

where E_{ls} and E_c, respectively, denote the training objectives of the localization-segmentation and classification subnetworks. The term $\partial E_{ls}/\partial \boldsymbol{W}_{ls}$ represents the BP stage within localization-segmentation.

The second term of (17) can be expanded as

$$\frac{\partial E_c}{\partial \boldsymbol{W}_{ls}} = \frac{\partial E_c}{\partial V}\left(\frac{\partial V}{\partial \boldsymbol{L}}\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W}_{ls}} + \frac{\partial V}{\partial \boldsymbol{O}}\frac{\partial \boldsymbol{O}}{\partial \boldsymbol{W}_{ls}}\right) \tag{18}$$

where V is short for $V(\boldsymbol{L}, \boldsymbol{O}; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f)$. As shown in Fig. 4, $\partial E_c/\partial V$ passes useful information in the BP stages within classification. It is computed by back-propagated the gradient to the input image of classification subnetwork. It is noted that the alignment subnetwork does not have learnable parameters. Thus, we directly back-propagate the gradient $\partial E_c/\partial V$ to update the localization-segmentation subnetwork.

In 18, the gradients $\partial \boldsymbol{L}/\partial \boldsymbol{W}_{ls}$ and $\partial \boldsymbol{O}/\partial \boldsymbol{W}_{ls}$ are used to update the localization-segmentation network following conventional box [54], [55], [72] and pixel-wise mask [59], [60] regression. With the chain rule, the valve function V connects the classification and localization-segmentation subnetwork in the BP stage. As illustrated in Fig. 4, this connection is represented by $\partial V/\partial \boldsymbol{L}$ and $\partial V/\partial \boldsymbol{O}$ in (18). With it, the update of the localization-segmentation subnetwork is sensitive to the back-propagated signal of the classification subnetwork.

Furthermore, the signal communicated between the classification and localization-segmentation subnetworks can be adaptively tuned by the VLF. In the BP stage, the valve function V can be rewritten as

$$\begin{aligned} &V(\boldsymbol{L}, \boldsymbol{O}; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f) \\ &\quad = \frac{1}{e} E_a(\boldsymbol{c}^*, \theta^*, \alpha^*, \boldsymbol{t}^*; \boldsymbol{I}, \boldsymbol{L}, \boldsymbol{O}) \boldsymbol{I}(\boldsymbol{c}^*, \theta^*, \alpha^*) \end{aligned} \tag{19}$$

where $e = E_a(\boldsymbol{c}^*, \theta^*, \alpha^*, \boldsymbol{t}^*; \boldsymbol{I}, \boldsymbol{L}_f, \boldsymbol{O}_f)$ is the alignment energy computed in the FP stage. This forward-propagated alignment energy is applied in the adaptive update for localization and segmentation.

C. Adaptive Update for Localization

We show that VLF provides information from the classification subnetwork and adaptively control the update in terms of the localization task.

As in (19), the forward-propagated alignment energy is regarded as a constant in the BP stage. With this energy, the linkage part $\partial V/\partial \boldsymbol{L}$ can be expressed as

$$\frac{\partial V}{\partial \boldsymbol{L}} = \frac{1}{e} \boldsymbol{I}(\boldsymbol{c}^*, \theta^*, \alpha^*) \frac{\partial E_a}{\partial \boldsymbol{L}}. \tag{20}$$

And the term $\partial E_a/\partial \boldsymbol{L}$ is extended to

$$\frac{\partial E_a}{\partial \boldsymbol{L}} = -\frac{\lambda_d}{2\sigma^2} \exp\left(-\frac{\|\boldsymbol{c} - \boldsymbol{c}^r(\boldsymbol{L})\|^2}{2\sigma^2}\right) \frac{\partial \|\boldsymbol{c} - \boldsymbol{c}^r(\boldsymbol{L})\|^2}{\partial \boldsymbol{L}} \tag{21}$$

where $\boldsymbol{c} = (c_x, c_y)$ and we have

$$\begin{aligned} &\frac{\partial \|\boldsymbol{c} - \boldsymbol{c}^r(\boldsymbol{L})\|^2}{\partial \boldsymbol{L}} \\ &= (2c_x - x_1 - x_2, 2c_y - y_1 - y_2, 2c_x - x_1 - x_2, 2c_y - y_1 - y_2). \end{aligned} \tag{22}$$

Here, the factor $1/e$ can be deemed to be a valve controlling the influence from classification. As described in Section III-C, a larger alignment energy e corresponds to better alignment in the FP stage. In the BP stage, $1/e$ is used to reweight the update signal $\partial E_c/\partial V$ from the classification network. It functions as a compromise between classification and alignment errors.

In this case, a large e means good alignment in the BP stage, for which information from the classification subnetwork is automatically reduced given a small $1/e$. In contrast, if e is small, current alignment becomes less reliable. Thus more classification information is automatically introduced by the large $1/e$ to guide $\boldsymbol{W}_{ls}$ update for localization. Simply put, one can understand $1/e$ as a dynamic learning rate in the BP stage. It is adaptive to matching performance.

D. Adaptive Update for Segmentation

Similar to (20), the linkage part for segmentation output $\partial V/\partial \boldsymbol{O}$ in (18) can be written as

$$\frac{\partial V}{\partial \boldsymbol{O}} = \frac{1}{e} \boldsymbol{I}(\boldsymbol{c}^*, \theta^*, \alpha^*) \frac{\partial E_a}{\partial \boldsymbol{O}}. \tag{23}$$

The partial derivative $\partial E_a/\partial \boldsymbol{O}$ can be expressed by element-wise expansion as

$$\frac{\partial E_a}{\partial \boldsymbol{O}(\boldsymbol{c}_i)} = \frac{\lambda_s \boldsymbol{t}_m(\boldsymbol{c}_i)}{(1 - \boldsymbol{O}(\boldsymbol{c}_i))N_f} - \frac{\lambda_s(1 - \boldsymbol{t}_m(\boldsymbol{c}_i))}{\boldsymbol{O}(\boldsymbol{c}_i)N_b}. \tag{24}$$

Besides the adaptive factor $1/e$, the update of segmentation is also guided by the template $\boldsymbol{t}_m$, as formulated in (24). With the definition of (24), the template where $\boldsymbol{t}_m(\boldsymbol{c}_i) = 1$ allows the signal $\partial E_a/\partial \boldsymbol{O}(\boldsymbol{c}_i) = \lambda_s/(1 - \boldsymbol{O}(\boldsymbol{c}_i))N_f$ to supervise the segmentation. On the other hand, the signal becomes $\partial E_a/\partial \boldsymbol{O}(\boldsymbol{c}_i) = -\lambda_s/\boldsymbol{O}(\boldsymbol{c}_i)N_b$ where $\boldsymbol{t}_m(\boldsymbol{c}_i) = 0$. It means that the control signals can be flexibly switched according to the foreground/background region of the template. Using the template to guide BP shares the spirit with the methods of [73], [74]. As the template mask matched with the part region becomes available, the network is supervised not only by the object region that reduces global segmentation error, but also by the template shape information that rectifies object boundaries. Fig. 6 shows the alignment results from the final LSAC model. With the shape information provided by the segmentation subnetwork, we reduce the distraction of the background pixels and improve the quality of alignment. The comparison in Fig. 7 demonstrates that including additional shape information improves the segmentation results.

With this kind of auto-adjustment mechanism in our VLF connecting classification and alignment, the localization-segmentation subnetwork can be refined in the BP stage. We verify this design in experiments.

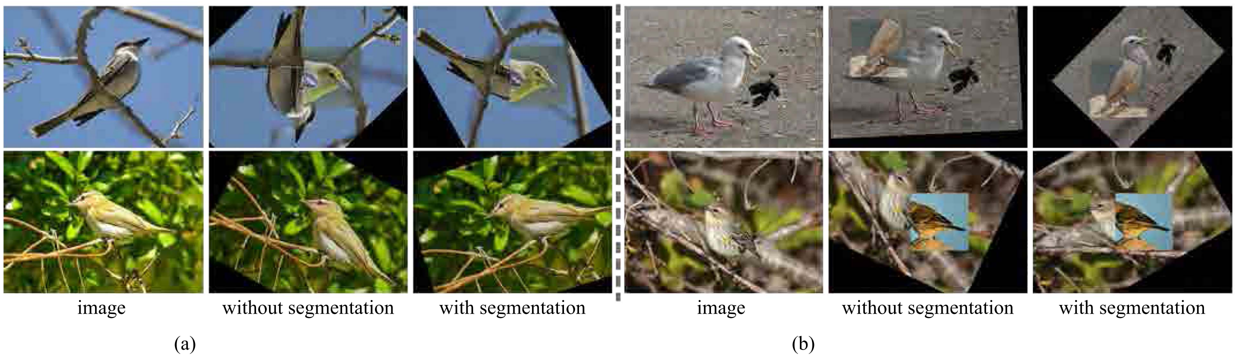

Fig. 6. Images and alignment of the head parts (a) and the torso parts (b). We show the alignment results without/with using the segmentation subnetwork.

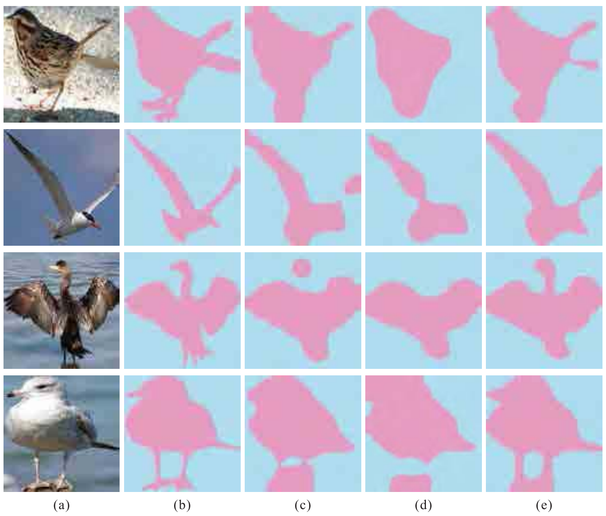

Fig. 7. (a) Input image. (b) Segmentation ground-truth. The results are achieved by (c) segmentation branch without parameter sharing, (d) segmentation branch without VLF, and (e) segmentation branch of LSAC.

V. Implementation Details

A. CNN Construction

In implementation, we modify the Caffe platform [75] for CNN construction. Bird heads and torsos are considered as semantic parts. We train two deep LSACs for them, respectively. All CNN models are fine-tuned based on Visual Geometry Group (VGG)-16 [20]. In the localization-segmentation subnetwork, all input images are resized to 224×224. We remove the original 1000-way fully connected layer. The network is transformed to a fully convolutional network (FCN) following [59]. The fully connected layers, that is, $fc6$ and $fc7$, are now convolutional layers. This module outputs a structure that comprises 4-way regression for part-bounding boxes and a pixel-wise probability map for foreground/background labels. The pretrained model on ImageNet is used to initialize our localization-segmentation subnetwork. The classification subnetwork takes input images each with size 224×224. The first fully connected layer $fc6$ is extracted to form a 4096D feature. Then we follow the CNN-SVM scheme [76] to train an SVM classifier on our CNN feature.

B. Template Alignment

For alignment, in template selection, all 5994 part annotations for head or torso in the training set of the Caltech-University of California San Diego (UCSD) Bird-200-2011 data set [1] are used. The 5994 parts are cropped and resized to 224×224. Using spectral clustering, we obtain the 5994-part split into 30 clusters. From each cluster, we select the part region closest to the cluster center and its mirrored version as two templates. This process eventually forms 60-template $\mathfrak{T}$. The rotation degree θ is an integer. Its range is $\Theta = [-60, 60]$ with an interval of 10°. We search the scale α within $\mathfrak{A}$. As all the input images and templates are resized to 224×224, a part in an image is smaller than any template. We must scale up an input image to match the sizes of a part and a template. To this end, we set $\mathfrak{A} = \{1.5, 2.7, 4.0, 7.7, 15.0\}$ for the head, and $\mathfrak{A} = \{1.2, 1.4, 2.0, 2.5, 3.0\}$ for the torso.

TABLE II

Average Time Consumptions of Subnetworks. The Time Consumption Is Reported in Terms of Millisecond (ms)

methods	localization-segmentation	alignment	classification
training stage	6.7	3.8	4.5
testing stage	3.6	1.3	3.1

The searching spaces in terms of the template, rotation degree, and scales are tuned by respecting the performance on the validation set, which comprises 1000 images randomly selected from the training set. By broadening the searching spaces, we find negligible improvement but extra computational overhead. Thus, we continue to use the searching spaces throughout all experiments. We remark that the results of the pose-similarity function [see (8)] can be precomputed and stored. We use Nvidia TITAN X Pascal graphics card with 3840 cores and 12 GB memory for acceleration. It takes about 5 s/image to accomplish the computation of pose similarity over all possible locations, templates, scales, and rotation degrees. Thus, the pose similarity can be quickly looked up in FP stage. By using VGG-16 architecture, our implementation enables 15 ms/image training and 8 ms/image testing time. In Table II, we report the average training and testing times for an image in different network components (i.e., LSAC subnetworks). By sharing the network parameters of the localization and segmentation subnetworks, we save the GPU memory for storing the parameters, leading to a more compact network. Compared to using the independent localization and segmentation subnetworks, we save 26% of the training time for achieving the converged network parameters.

VI. Experiments

We evaluate our method on three data sets: 1) Caltech-UCSD Bird (CUB)-200-2011 [1]; 2) CUB-200-2010 [2]; and 3) Stanford Cars-196 [5]. The CUB-200-2011 data set is more

TABLE III

OBJECT SEGMENTATION ACCURACY (%) ON THE CUB-200-2011 DATA SET USING DIFFERENT METHODS. THE ABBREVIATIONS "BG" AND "FG" REFER TO THE BACKGROUND AND FOREGROUND, RESPECTIVELY

segmentation	parameters		VLF	
	w/o sharing	w/ sharing	w/o VLF	w/ VLF
bg	82.8	**85.4**	80.7	**85.4**
fg	81.0	**83.5**	79.0	**83.5**
mIoU	81.9	**84.5**	79.9	**84.5**

widely used for analysis. We thus conduct our major evaluation on that data set, while using the other two data sets for extensive comparison with state-of-the-art methods.

A. Caltech-UCSD Bird-200-2011 Data Set

We evaluate our method on the CUB-200-2011 data set [1]. This data set contains 11 788 images of birds, divided into 200 subordinate categories. Each image contains the species along with the bird bounding box. Annotation of bird parts is provided.

During training and testing, we use bounding boxes of the data set to simplify classification, similar to previous work [8], [33], [40], [45], [47], [77]. Our experiments follow the training/testing split fixed in [1]. We define two kinds of semantic templates, that is, "head" and "torso", as in [8], [7], and [31]. Because there is no such annotation, we follow the method of [8], [10], [31] to obtain corresponding rectangles covering annotated parts within bird heads and torsos.

1) Analysis of Parameter Sharing: Our localization and segmentation share convolutional parameters to capture object-part relationships. To investigate its efficacy, we build these two components with independent CNNs, and compare their performance with the sharing-parameter counterpart.

We evaluate the part localization performance based on percentage of correctly localized parts (PCP) [8], which is computed on the top-ranked part prediction and regards parts with $\geq$0.5 overlap with ground-truth as correct. The independent localization results are 93.2 and 94.3 for the head and torso parts. By parameter sharing with segmentation, we obtain better results 95.0 and 97.0.

In Table III, we analyze parameter sharing in terms of segmentation performance. Following [35], we use intersection-over-union (IoU) scores to evaluate the segmentation performance. A mean IoU (mIoU) score is also computed to evaluate the overall segmentation accuracy. The comparison shows that parameter sharing generally improves segmentation on background and foreground regions. The visual difference between Fig. 7(c) and (e) is clear.

2) Analysis of VLF on Localization-Segmentation: To further understand the importance of the localization-segmentation module using our VLF, we move this subnetwork out of the joint LSAC model and compare it with our overall LSAC.

The comparison on localization accuracy is shown in Fig. 8. We test the performance in terms of PCP with overlap $\geq$ 0.5, 0.6, 0.7, and 0.8. In all configurations, the localization branch (LOC) alone performs worse than applying the whole LSAC model. The segmentation subnetwork also suffers from performance degradation, as shown in Table III. Fig. 7(d) shows the segmentation results. The issue is that the localization-segmentation subnetwork does not receive

TABLE IV

COMPARISON WITH STATE-OF-THE-ART IN TERMS OF PART LOCALIZATION ACCURACY (%) ON PART OVERLAP $\geq$ 0.5 WITH GROUND-TRUTH ON THE CUB-200-2011 DATA SET

CNN model	methods	head	torso
w/o CNN	Zhang et al. [8]	68.2	79.8
	Zhang et al. [78]	63.8	89.1
AlexNet	Lin et al. [31]	74.0	96.0
	Shih et al. [79]	88.9	94.3
VGGNet	Lin et al. [31]	90.0	96.2
	Zhang et al. [10]	93.4	94.9
	ours	**95.0**	**97.0**
ResNet	Zhang et al. [80]	95.2	97.2
	Yang et al. [81]	95.6	97.5
	ours	**96.7**	**98.6**

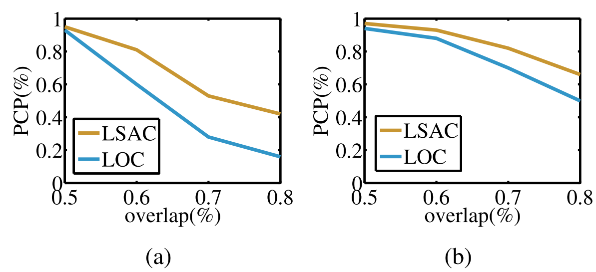

Fig. 8. PCPs of (a) bird head and (b) bird torso under different overlap rates. "LSAC" and "LOC" refer to the LSAC model and localization branch, respectively.

feedback from alignment and classification while our LSAC updates all of them in each iteration.

3) Comparison With Other Localization Methods: To evaluate part localization, we make comparisons with other methods in Table IV. Previous methods [10], [31], [32], [41], [61], [79] localize heads and torsos. We use VGG-16 architecture. With the same experimental setup, we show the comparisons in Table IV.

For the head and torso parts, our results are 95.0 and 97.0 compared to the previous best results of 93.4 [10] and 96.2 [31]. Fig. 9 shows a few examples, where (a) and (b) involve predicted bounding boxes of bird heads and torsos. Compared to our previous localization, alignment and classification (LAC) model [31], all-part localization is improved by our new LSAC. In particular, head localization, which is challenging due to small regions, is improved notably from 90.0 to 95.0. The performance gap suggests the importance of our localization-segmentation subnetwork that captures object-part relationships, which is beneficial to bounding box regression.

Next, we use ResNet-50 [21] as the backbone architecture of LSAC, yielding better localization accuracies. We compare our approach to other localization methods [80], [81], which also use ResNet-50. We employ the released implementations of these methods to output the image representations, which are used for localizing the head and torso parts. As shown in Table IV, our approach outperforms these methods.

4) Comparison to Other Segmentation Methods: Our LSAC model includes segmentation. An alternative is to train a baseline FCN [59] for object segmentation. Besides CNN-based solutions, the interactive object segmentation tool GrabCut [69] and cosegmentation method [82] can also be used. We report the segmentation accuracy of these methods in Table V.

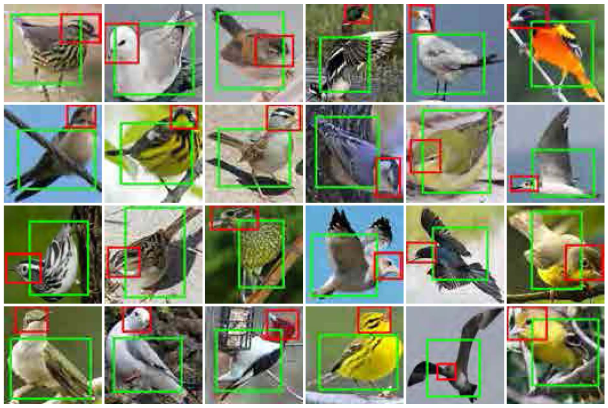

Fig. 9. Localization examples of bird heads (in red rectangles) and torsos (in green rectangles).

TABLE V

COMPARISON WITH OTHER SEGMENTATION METHODS IN TERMS OF OBJECT SEGMENTATION (%) ON THE CUB-200-2011 DATA SET

CNN model	methods	bg	fg	mIoU
w/o CNN	Rother et al. [69]	61.6	64.3	63.0
	Joulin et al. [82]	66.3	68.1	67.2
VGGNet	Simonyan et al. [20]	79.6	78.0	78.8
	ours	**85.4**	**83.5**	**84.5**
ResNet	Zhang et al. [80]	85.9	84.7	85.3
	Yang et al. [81]	86.5	85.5	86.0
	ours	**88.3**	**87.0**	**87.7**

As shown in Table V, the baseline FCN yields a mean IoU of 78.8 compared to our LSAC score of 84.5. The performance degradation stems from the fact that the baseline FCN does not benefit from parameter sharing. Also, this single FCN is isolated from the end-to-end training of multiple tasks that are included in our LSAC model. The non-CNN methods, i.e., GrabCut [69] and cosegmentation method [82], perform less accurately since they depend on low-level image representation that loses semantic object information. Fig. 10 shows examples.

In Table V, we compare our approach to the latest methods [80], [81] that are based on ResNet-50. With a stronger backbone model, the latest methods provide more powerful image representations for the segmentation of objects. They outperform the previous methods which are based on AlexNet or VGGNet. Our approach also benefits from ResNet-50 and outperforms the competitive methods, in terms of the segmentation accuracies.

5) Sensitivities to Alignment Hyperparameters: The alignment energy function (see 9 and 12) has hyperparameters $\{\lambda_a, \sigma, \lambda_d, \lambda_s\}$ that control the importance of different subnetworks. To examine the effect of individual subnetworks on the alignment result, which is eventually related to the classification accuracy, we compare different hyperparameters and report classification accuracies in Fig. 11. To simplify the comparison, we investigate each hyperparameter separately, while fixing other parameters according to the baseline setting $\{1, 15, 0.001, 0.0003\}$.

At first, we compare different values of λ_a [see Fig. 11(a)]. By setting λ_a to 0, we remove the similarity function between the template and object part, yielding a large performance drop. This case is very similar to using $\lambda_s = 0$ [see Fig. 11(d)] that disables the objectness confidence, which also largely

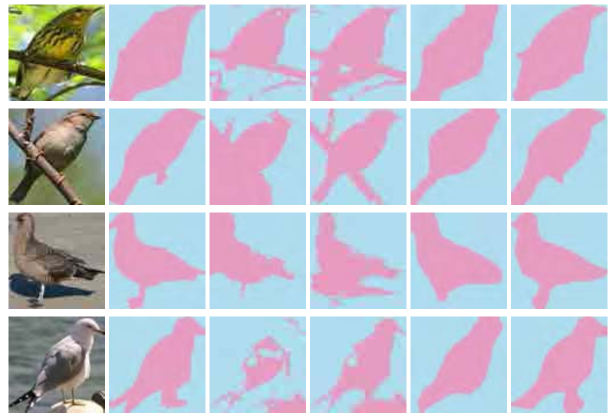

Fig. 10. (a) Input image. (b) Segmentation ground-truth. The results are achieved by (c) GrabCut, (d) cosegmentation, (e) baseline FCN, and (f) segmentation branch of LSAC.

TABLE VI

CLASSIFICATION ACCURACY (%) OF SEMANTIC PARTS, I.E., HEAD AND TORSO, ON THE CUB-200-2011 DATA SET. WE, RESPECTIVELY, BLOCK LOCALIZATION AND ALIGNMENT SUBNETWORKS TO EVALUATE PERFORMANCE

methods	w/o loc-seg	w/o VLF	w/o VLF-BP	w/o seg	full model
head	76.3	77.1	78.2	76.6	**79.5**
torso	**76.3**	52.2	58.5	54.7	63.3

degrades the classification performance. It is noted that the similarities and objectness confidences guide the alignment according to texture and shape information, respectively. Without them, the alignment process is unreliable for the classification subnetwork. We find that too large λ_a or λ_s reduces the classification accuracy, because the alignment subnetwork may become oversensitive to the problematic similarities and objectness confidences.

Next, we study the localization regularization by controlling hyperparameters σ and λ_d. With larger σ [see Fig. 11(b)], the alignment process becomes less sensitive to localization results. This is similar to use smaller λ_d that reduces the impact of localization results. But larger λ_d reduces the effect of similarity function and objectness confidence on alignment, leading to the degradation of classification accuracy.

6) Subnetwork Combination Analysis: Our experimental results confirm that LSAC with all three subnetworks is powerful in part localization and object segmentation. We also evaluate the performance in fine-grained classification and experiment with removing one or two components in the following five cases.

First, we remove the localization-segmentation subnetwork by validating the classification accuracy on images. The results are listed in the first row of Table VI. Without this module, the whole-image classification accuracy is only 76.3.

Second, we remove the alignment subnetwork. The localization-segmentation subnetwork is used to propose part hypotheses for classification. In this case, the localization-segmentation and classification modules are trained independently in BP stages. The results in the second row of Table VI indicate that lack of message propagation in alignment is not recommended.

Third, we use VLF in the alignment subnetwork to output pose-aligned parts for classification in the FP stage. But VLF is disabled in the BP stage to prevent classification and alignment

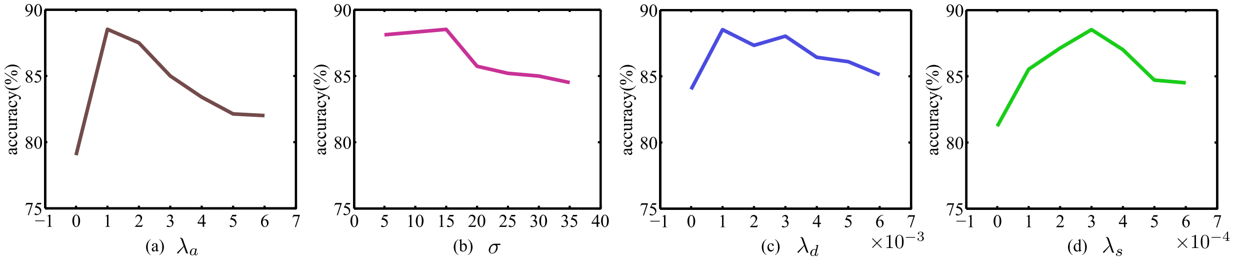

Fig. 11. Sensitivity to alignment hyperparameters. Classification accuracies are evaluated on the CUB-200-2011 data set.

errors from being propagated to localization and segmentation. In this case, we boost the accuracy to 78.2 on bird heads (the third row of Table VI). The alignment subnetwork, along with FP and BP, is thus necessary.

Fourth, we enable VLF during forward and backward stages. But the segmentation branch is removed so that the framework degrades to our previous LAC [31] model, which consists of the localization, alignment, and classification modules. Without the segmentation branch, only localization results are insufficient as reported in the fourth row of Table IV. Unsurprisingly, this model configuration leads to performance degradation in both head and torso classifications.

By replacing the whole image with the torso part, we find a significant performance gap of 24.1 (76.3 vs. 52.2) with respect to classification accuracy. The similar observations were reported in [7] and [79]. The high PCP (97) for torso localization in Table IV demonstrates that the mis-localization causes little performance drop. We conclude that the bird torso is not that distinct for bird species identification, compared to the whole image that includes the discriminative head part. Our alignment subnetwork can improve the classification accuracy using the torso part. After adding alignment (with VLF), the performance gain is about 11.1 (63.3 vs. 52.2). This improvement shows that the better feature of the torso part is produced. The reliability of the torso part is important. It can be combined with the head part and the whole image to benefit the eventual classification.

7) Overall Comparison: Our final classification accuracy compared with state-of-the-art methods is presented in Table VII. The CNN models that are used by all the compared methods are included in the first column of Table VII. All results are accomplished under the setting that the bounding box for the entire bird is given in training and testing. Part annotation is available only during training. In our system, we feed each image into the two trained networks to extract features of the head and torso.

Table VII shows that using the head and torso features achieve 79.5 and 63.3 accuracy. We concatenate the two feature vectors to form a combined representation with 83.7 accuracy. We finally tune the CNN model based on the whole image using the pretrained model [20]. The sixth layer is extracted for training an SVM classifier, obtaining 76.3 accuracy. After concatenating the features of head, torso, and the whole image, our accuracy increases to 88.5. The methods of [10], [31] also consider head and torso parts, and combine CNN features of the whole image. Our improvement on accuracy is mainly due to the reliable localization, segmentation, and alignment in the VLF-enabled LSAC. We compare LSAC to the recent approaches [29], [51], [81], [84], which also select the semantic object parts for fine-grained recondition. For a fair comparison, we use ResNet-50 as the backbone

TABLE VII

COMPARISON WITH STATE-OF-THE-ART ON THE CUB-200-2011 DATA SET

CNN models	methods	accuracy
w/o CNN	Lee et al. [77]	41.0
	Berg et al. [47]	56.9
	Goering et al. [83]	57.8
	Chai et al. [33]	59.4
	Gravves et al. [45]	62.7
AlexNet	Zhang et al. [40]	65.0
	Zhang et al. [8]	76.4
	Lin et al. [31]	80.3
VGGNet	Lin et al. [31]	81.4
	Lin et al. [63]	84.1
	Zhang et al. [10]	85.1
	Jaderberg et al. [65]	86.3
	ours (head)	79.5
	ours (torso)	63.3
	ours (head+torso)	83.7
	ours (whole image)	76.3
	ours (head+torso+whole image)	**88.5**
ResNet	Sun et al. [84]	87.8
	Yang et al. [81]	88.0
	Zheng et al. [51]	88.1
	Yan et al. [29]	88.6
	ours (head+torso+whole image)	**89.3**

architecture of LSAC. Again, we concatenate the features of head, torso and the whole image, yielding better classification accuracy than other methods.

B. Caltech-UCSD Bird-200-2010 Data Set

The CUB-200-2010 data set [2] provides 6033 images from 200 bird categories. It offers no part annotation and contains less training/testing data. It thus can verify whether our LSAC, which is trained on the CUB-200-2011 data set, can be generalized to this data set.

The classification results are shown in Table VIII. The network is trained with the data from the CUB-200-2011 data set. The classification subnetwork is updated on this data set after getting the pose-aligned part images.

Our whole-image classification accuracy (in the "w/o localization-segmentation" row) is 63.7. Through the localization-segmentation subnetwork, the classification accuracy of bird heads is 67.3. In this case, the performance gain is 3.6 (67.3 vs. 63.7). The gain increases to 6.5 after incorporating alignment. The best torso recognition accuracy of 49.1 is achieved by adding localization, segmentation, and alignment.

TABLE VIII

CLASSIFICATION ACCURACY (%) OF SEMANTIC PARTS ON THE CUB-200-2010 DATA SET

methods	head	torso
w/o localization-segmentation	63.7	**63.7**
w/ localization-segmentation	67.3	44.5
full model	**70.2**	49.1

TABLE IX

COMPARISON WITH STATE-OF-THE-ART ON THE CUB-200-2010 DATA SET

CNN models	methods	accuracy
w/o CNN	Yang et al. [44]	28.2
	Angelova et al. [49]	30.2
	Deng et al. [87]	32.8
	Goering et al. [83]	35.9
	Farrell et al. [39]	37.1
	Chai et al. [33]	47.3
AlexNet	Lin et al. [31]	65.3
VGGNet	Zhang et al. [10]	66.1
	Lin et al. [31]	66.5
	Jaderberg et al. [65]	74.6
	ours (head)	70.2
	ours (torso)	49.1
	ours (head+torso)	74.9
	ours (whole image)	63.7
	ours (head+torso+whole image)	**77.5**
ResNet	Mitsuhara et al. [85]	72.1
	Kong et al. [66]	77.1
	Chen et al. [86]	77.8
	ours (head+torso+whole image)	**79.0**

In the final experiment, we compare the classification accuracy with other methods. The results are listed in Table IX. State-of-the-art results are 66.1 by the method of [10], and 66.5 is achieved by the improved LAC model [31]. Both methods used VGG-16. Our bird-head representation obtains 70.2 accuracy. The combined head and torso representation reaches 74.9.

Similar to previous experiments, we consider the whole image. After combining all features, our classification performance is boosted to 77.5. Our representation outperforms previous best result by a remarkable margin of 11. We believe better performance can be achieved if the localization-segmentation and alignment subnetworks are adapted with part annotation, which is however not available in this data set. Based on a deeper backbone architecture (i.e., ResNet-50), LSAC outperforms the recent methods [66], [85], [86] on the CUB-200-2010 data set.

C. Stanford Cars-196 Data Set

Besides bird categorization, our LSAC model can be transferred to the fine-grained recognition on other object types. In this section, we use the Stanford Cars-196 data set [5] as an evaluation benchmark. This data set contains 16 185 images from 196 car classes, and is prepared for fine-grained recognition tasks. There are 8144 training images and 8041 testing images. Unlike the CUB-200-2011 data set [1], the Stanford Cars-196 data set [5] does not provide the object masks. To avail our LSAC model on this data set, we additionally provide the binary masks of all the cars in the 16 185 images. Fig. 12 shows the examples of our mask annotations. We again follow the implementation details as we have described in Section V, adjusting the scales to $\mathfrak{A} = \{1.1, 1.5, 1.7\}$.

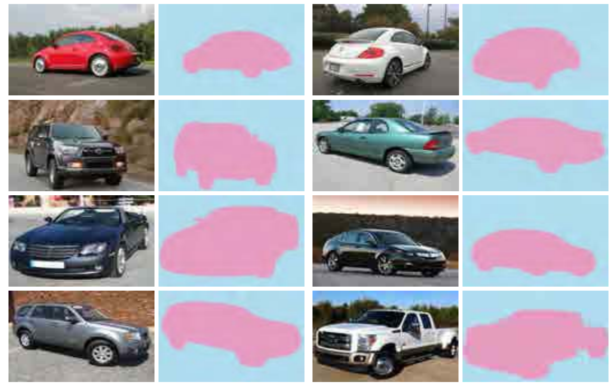

Fig. 12. Examples of annotated masks for Stanford Cars-196 data set.

TABLE X

COMPARISON WITH STATE-OF-THE-ART ON THE STANFORD CARS-196 DATA SET

CNN models	methods	accuracy
w/o CNN	Chai et al. [33]	78.0
AlexNet	Girshick et al. [89]	73.5
	Lin et al. [31]	82.4
VGGNet	Girshick et al. [88]	88.4
	Lin et al. [63]	91.3
	Lin et al. [31]	92.0
	Krause et al. [9]	92.6
	Jaderberg et al. [65]	93.8
	ours	**96.3**
ResNet	Chen et al. [86]	96.1
	Wang et al. [89]	96.3
	Yang et al. [81]	96.5
	ours	**97.8**

In Table X, we compare the classification accuracy of LSAC with those of other methods. When applying the LSAC model for car categorization, we perform the localization, segmentation, and alignment on the car with no subdivided part. Similarly, the compared methods also take their input as the whole car. Using VGG architecture, the previous best result of 92.6 is reported in [9]. Using the same VGG architecture to construct our LSAC model, we achieve better performance than the compared methods. Our result of 96.3 on the Stanford Cars-196 data set demonstrates that LSAC provides accurate classifications of cars. Using the powerful ResNet-50 as the backbone architecture, LSAC yields better classification accuracy on the Stanford Cars-196 data set and surpasses the recent methods [81], [86], [89].

D. Sensitivity to the Number of Annotations

We use ground-truth annotations (i.e., bounding boxes and pixel-wise categories) to train deep LSAC model. To examine the scalability of our model, we control the number of annotations for training. Initially, we use full annotations for training. Subsequently, we reduce 10% of annotations at each time, and evaluate the classification accuracies on the CUB-200-2011, CUB-200-2010, and Stanford Cars-196 data sets. All results can be found in Fig. 13.

By reducing the annotations, we decrease the classification performances on all data sets. This is because the localization, segmentation, and alignment become less reliable with less

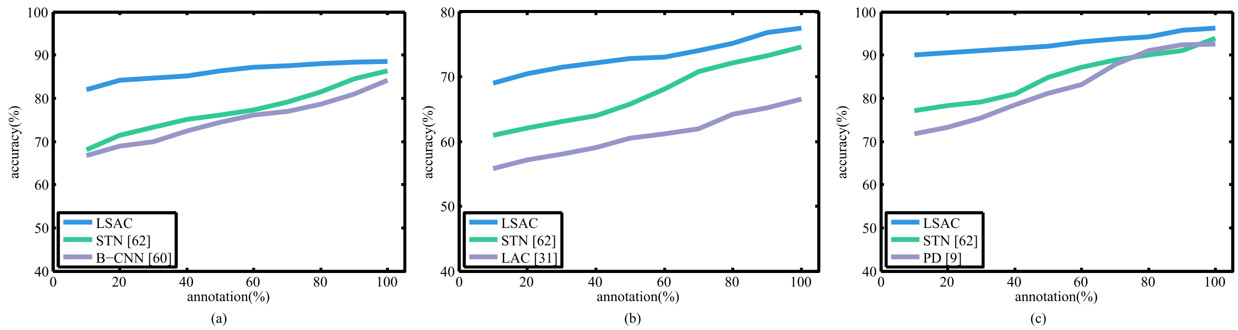

Fig. 13. Sensitivity to the number of annotations. Classification accuracies are evaluated on the (a) CUB-200-2011, (b) CUB-200-2010, and (c) Stanford Cars-196 data sets.

training data. Specifically, we compare the cases, where full annotations and only 10% of annotations are used, respectively. We generally find about seven points of performance drop on all data sets. Using 10% of annotations, we have no more than 800 images for training LSAC model. Rather than using full annotations, our LSAC model still achieves reasonable performance with much less data, remarkably saving the labeling effort. We compare our approach to other methods [9], [31], [63], [65]. Given inadequate annotations, these methods yield unsatisfactory classification accuracies on different data sets.

VII. Concluding Remarks

We present a deep neural network to achieve fine-grained recognition. We share the same observation with previous work that proper localization, segmentation, and alignment of salient object parts are important. Based on this, we contribute a unified LSAC system to incorporate localization, alignment and classification. The modules are connected with an optimally defined VLF to enable smooth FP and BP. Results show that this process improves part finding and matching, as well as object segmentation, which eventually helps classification.

Acknowledgment

The authors thank the anonymous reviewers and editors for their constructive suggestions.

References

[1] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep., 2011.

[2] P. Welinder *et al.*, "Caltech-UCSD birds 200," California Inst. Technol., Pasadena, CA, USA, Tech. Rep., 2010.

[3] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog breed classification using part localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 172–185.

[4] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013, *arXiv:1306.5151*. [Online]. Available: http://arxiv.org/abs/1306.5151

[5] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 554–561.

[6] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[7] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird species categorization using pose normalized deep convolutional nets," 2014, *arXiv:1406.2952*. [Online]. Available: http://arxiv.org/abs/1406.2952

[8] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 834–849.

[9] J. Krause, H. Jin, J. Yang, and L. Fei-Fei, "Fine-grained recognition without part annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5546–5555.

[10] H. Zhang *et al.*, "SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1143–1152.

[11] Z. Ma, X. Chang, Z. Xu, N. Sebe, and A. G. Hauptmann, "Joint attributes and event analysis for multimedia event detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2921–2930, Jul. 2018.

[12] W. Shi, Y. Gong, X. Tao, D. Cheng, and N. Zheng, "Fine-grained image classification using modified DCNNs trained by cascaded softmax and generalized large-margin losses," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 683–694, Mar. 2019.

[13] Y. Yu, S. Tang, K. Aizawa, and A. Aizawa, "Category-based deep CCA for fine-grained venue discovery from multimodal data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1250–1258, Apr. 2019.

[14] T. Chen, L. Lin, L. Liu, X. Luo, and X. Li, "DISC: Deep image saliency computing via progressive representation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1135–1149, Jun. 2016.

[15] C. Zhang, J. Cheng, C. Li, and Q. Tian, "Image-specific classification with local and global discriminations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4479–4486, Sep. 2018.

[16] P. Tang, X. Wang, B. Shi, X. Bai, W. Liu, and Z. Tu, "Deep FisherNet for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2244–2250, Jul. 2019.

[17] C. Zhang, J. Cheng, L. Li, C. Li, and Q. Tian, "Object categorization using class-specific representations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4528–4534, Sep. 2018.

[18] C. Zhang, C. Liang, L. Li, J. Liu, Q. Huang, and Q. Tian, "Fine-grained image classification via low-rank sparse coding with general and class-specific codebooks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1550–1559, Jul. 2017.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[23] J. Ngiam, Z. Chen, D. Chia, P. W. Koh, Q. V. Le, and A. Y. Ng, "Tiled convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1279–1287.

[24] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 392–407.

[25] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[26] J. Liu, Y. Wang, Y. Li, J. Fu, J. Li, and H. Lu, "Collaborative deconvolutional neural networks for joint depth estimation and semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5655–5666, Nov. 2018.

[27] H. Zheng, J. Fu, Z.-J. Zha, J. Luo, and T. Mei, "Learning rich part hierarchies with progressive attention networks for fine-grained image recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 476–488, 2020.

[28] X. He, Y. Peng, and J. Zhao, "StackDRL: Stacked deep reinforcement learning for fine-grained visual categorization," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 741–747.

[29] T. Yan, S. Wang, Z. Wang, H. Li, and Z. Luo, "Progressive learning for weakly supervised fine-grained classification," *Signal Process.*, vol. 171, Jun. 2020, Art. no. 107519.

[30] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked CNN for fine-grained visual categorization," 2015, *arXiv:1512.08086*. [Online]. Available: http://arxiv.org/abs/1512.08086

[31] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep LAC: Deep localization, alignment and classification for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1666–1674.

[32] N. Zhang, E. Shelhamer, Y. Gao, and T. Darrell, "Fine-grained pose prediction, normalization, and recognition," 2015, *arXiv:1511.07063*. [Online]. Available: http://arxiv.org/abs/1511.07063

[33] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 321–328.

[34] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[35] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[36] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 2169–2178.

[37] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.

[38] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1794–1801.

[39] 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 *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 161–168.

[40] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 729–736.

[41] H. Azizpour and I. Laptev, "Object detection using strongly-supervised deformable part models," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 836–849.

[42] D. Forsyth, "Object detection with discriminatively trained part-based models," *Computer*, vol. 47, no. 2, pp. 6–7, Feb. 2014.

[43] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *Proc. CVPR*, Jun. 2011, pp. 1577–1584.

[44] S. Yang, L. Bo, J. Wang, and L. G. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 3122–3130.

[45] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1713–1720.

[46] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[47] T. Berg and P. N. Belhumeur, "POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 955–962.

[48] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, "Hierarchical part matching for fine-grained visual categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1641–1648.

[49] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 811–818.

[50] M. Simon, E. Rodner, T. Darrell, and J. Denzler, "The whole is more than its parts? From explicit to implicit pose normalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 749–763, Mar. 2020.

[51] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5012–5021.

[52] C. Figueroa Flores, A. Gonzalez-García, J. van de Weijer, and B. Raducanu, "Saliency for fine-grained object recognition in domains with scarce training data," 2018, *arXiv:1808.00262*. [Online]. Available: http://arxiv.org/abs/1808.00262

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[54] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[55] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[56] D. Lin *et al.*, "ZigZagNet: Fusing top-down and bottom-up context for object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7490–7499.

[57] D. Lin and H. Huang, "Zig-zag network for semantic segmentation of RGB-D images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2642–2655, Oct. 2020.

[58] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 603–619.

[59] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[60] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," 2016, *arXiv:1606.00915*. [Online]. Available: http://arxiv.org/abs/1606.00915

[61] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.

[62] G.-S. Xie, X.-Y. Zhang, W. Yang, M. Xu, S. Yan, and C.-L. Liu, "LG-CNN: From local parts to global discrimination for fine-grained recognition," *Pattern Recognit.*, vol. 71, pp. 118–131, Nov. 2017.

[63] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1449–1457.

[64] M. Simon, E. Rodner, and J. Denzler, "Part detector discovery in deep convolutional neural networks," in *Proc. ACCV*, 2014, pp. 162–177.

[65] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[66] S. Kong and C. Fowlkes, "Low-rank bilinear pooling for fine-grained classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 365–374.

[67] J. Krause *et al.*, "The unreasonable effectiveness of noisy data for fine-grained recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 301–320.

[68] M. Guillaumin, D. Küttel, and V. Ferrari, "ImageNet auto-annotation with segmentation propagation," *Int. J. Comput. Vis.*, vol. 110, no. 3, pp. 328–348, Dec. 2014.

[69] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut' interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[70] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: A survey," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 986–1004, Aug. 2003.

[71] A. Y. Ng *et al.*, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.

[72] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*. [Online]. Available: http://arxiv.org/abs/1312.6229

[73] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3150–3158.

[74] Y. He, W.-C. Chiu, M. Keuper, and M. Fritz, "STD2P: RGBD semantic segmentation using spatio-temporal data-driven pooling," 2016, *arXiv:1604.02388*. [Online]. Available: http://arxiv.org/abs/1604.02388

[75] Y. Jia. (2013). *Caffe: An Open Source Convolutional Architecture for Fast Feature Embedding*. [Online]. Available: http://caffe.berkeleyvision.org/

[76] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 806–813.

[77] Y. J. Lee, A. A. Efros, and M. Hebert, "Style-aware mid-level representation for discovering visual connections in space and time," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1857–1864.

[78] N. Zhang, R. Farrell, and T. Darrell, "Pose pooling kernels for subcategory recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3665–3672.

[79] K. J. Shih, A. Mallya, S. Singh, and D. Hoiem, "Part localization using multi-proposal consensus for fine-grained categorization," 2015, *arXiv:1507.06332*. [Online]. Available: http://arxiv.org/abs/1507.06332

[80] Y. Zhang, K. Jia, and Z. Wang, "Part-aware fine-grained object categorization using weakly supervised part detection network," *IEEE Trans. Multimedia*, vol. 22, no. 5, pp. 1345–1357, May 2020.

[81] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang, "Learning to navigate for fine-grained classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 420–435.

[82] A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 542–549.

[83] C. Goering, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric part transfer for fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2489–2496.

[84] M. Sun, Y. Yuan, F. Zhou, and E. Ding, "Multi-attention multi-class constraint for fine-grained image recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 805–821.

[85] M. Mitsuhara *et al.*, "Embedding human knowledge into deep neural network via attention map," 2019, *arXiv:1905.03540*. [Online]. Available: http://arxiv.org/abs/1905.03540

[86] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5157–5166.

[87] J. Deng, J. Krause, and L. Fei-Fei, "Fine-grained crowdsourcing for Fine-grained recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 580–587.

[88] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[89] Y. Wang, V. I. Morariu, and L. S. Davis, "Learning a discriminative filter bank within a CNN for fine-grained recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4148–4157.

Di Lin (Member, IEEE) received the bachelor's degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2016.

He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests are computer vision and machine learning.

Yi Wang (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2017.

He is currently serving as an Assistant Professor with the School of Biomedical Engineering, Shenzhen University, Shenzhen, China. His current research interests include medical image computing, computer vision, image processing, and machine learning.

Lingyu Liang (Member, IEEE) received the B.E. and Ph.D. degrees from the South China University of Technology (SCUT), Guangzhou, China, in 2009 and 2014, respectively.

From 2014 to 2016, he was a Postdoctoral Fellow with the School of Computer Science and Engineering, SCUT. From 2016 to 2017, he was an Honorary Postdoctoral Fellow with The Chinese University of Hong Kong, Hong Kong. He is currently an Associate Professor at SCUT. His research interests include image analysis and recognition, machine learning, and computational photography.

Ping Li (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2013.

He is currently a Research Assistant Professor with The Hong Kong Polytechnic University, Hong Kong. His current research interests include image/video stylization, GPU acceleration, and creative media. He has one image/video processing national invention patent, and has excellent research project reported worldwide by *Association for Computing Machinery (ACM) TechNews*.

C. L. Philip Chen (Fellow, IEEE) graduated from the University of Michigan, Ann Arbor, MI, USA, in 1985. He received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is currently a Chair Professor and the Dean of the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. Being a Program Evaluator of the Accreditation Board of Engineering and Technology Education (ABET), Baltimore, MD, USA, for computer engineering, electrical engineering, and software engineering programs, he successfully architects the University of Macau's Engineering and Computer Science Programs receiving accreditations from Washington/Seoul Accord through Hong Kong Institute of Engineers (HKIE), of which is considered as his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty of Science and Technology. His current research interests include systems, cybernetics, and computational intelligence.

He is a fellow of American Association for the Advancement of Science (AAAS), International Association for Pattern Recognition (IAPR), Chinese Association of Automation (CAA), and Hong Kong Institute of Engineers (HKIE), a member of the Academia Europaea (AE), the European Academy of Sciences and Arts (EASA), and the International Academy of Systems and Cybernetics Science (IASCYS). He received the IEEE Norbert Wiener Award in 2018 for his contribution in systems and cybernetics, and machine learnings. He was a recipient of the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University, in 1988. He was the IEEE Systems, Man, and Cybernetics Society President from 2012 to 2013, the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS from 2014 to 2019, and he is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON CYBERNETICS, and an Associate Editor for the IEEE TRANSACTIONS ON FUZZY SYSTEMS. He was the Chair of TC 9.1 Economic and Business Systems of International Federation of Automatic Control from 2015 to 2017. He is currently a Vice President of Chinese Association of Automation (CAA). He is also a highly cited Researcher by Clarivate Analytics in 2018 and 2019.