Weakly Supervised Complementary Parts Models for Fine-Grained Image Classification from the Bottom Up

Weifeng Ge[1,2*]   Xiangru Lin[2*]   Yizhou Yu[1†]

[1]Deepwise AI Lab   [2]The University of Hong Kong

Abstract

Given a training dataset composed of images and corresponding category labels, deep convolutional neural networks show a strong ability in mining discriminative parts for image classification. However, deep convolutional neural networks trained with image level labels only tend to focus on the most discriminative parts while missing other object parts, which could provide complementary information. In this paper, we approach this problem from a different perspective. We build complementary parts models in a weakly supervised manner to retrieve information suppressed by dominant object parts detected by convolutional neural networks. Given image level labels only, we first extract rough object instances by performing weakly supervised object detection and instance segmentation using Mask R-CNN and CRF-based segmentation. Then we estimate and search for the best parts model for each object instance under the principle of preserving as much diversity as possible. In the last stage, we build a bi-directional long short-term memory (LSTM) network to fuze and encode the partial information of these complementary parts into a comprehensive feature for image classification. Experimental results indicate that the proposed method not only achieves significant improvement over our baseline models, but also outperforms state-of-the-art algorithms by a large margin (6.7%, 2.8%, 5.2% respectively) on Stanford Dogs 120, Caltech-UCSD Birds 2011-200 and Caltech 256.

1. Introduction

Deep neural networks have demonstrated its ability to learn representative features for image classification [34, 25, 37, 17]. Given training data, image classification [9, 25] often builds a feature extractor that accepts an input image and a subsequent classifier that generates prediction probability for the image. This is a common pipeline in many high-level vision tasks, such as object detection [14, 16],

tracking [42, 33, 38], and scene understanding [8, 31].

Although a model trained with the aforementioned pipeline can achieve competitive results on many image classification benchmarks, its performance gain primarily comes from the model's capacity to discover the most discriminative parts in the input image. To better understand a trained deep neural network and obtain insights about this phenomenon, many techniques [1, 54, 2] have been proposed to visualize the intermediate results of deep networks. In Fig 1, it can be found that deep convolutional neural networks trained with image labels only tend to focus on the most discriminative parts while missing other object parts. However, focusing on the most discriminative parts alone can have limitations. Some image classification tasks need to grasp object descriptions that are as complete as possible. A complete object description does not have to come in one piece, but could be assembled together using multiple partial descriptions. To remove redundancies, such partial descriptions should be complementary to each other. Image classification tasks, that could benefit from such complete descriptions, include fine-grained classification tasks on Stanford Dogs 120 [21] and CUB 2011-200 [47], where appearances of different object parts collectively contribute to the final classification performance.

According to the above analysis, we approach image classification from a different perspective and propose a new pipeline that aims to mine complementary parts instead of the aforementioned most discriminative parts, and fuse the mined complementary parts before making final classification decisions.

Object Detection Phase. Object detection [10, 14, 16] is able to localize objects by performing a huge number of classifications at a large number of locations. In Fig 1, the red bounding boxes are the ground truth, the green ones are positive object proposals, and the blue ones are negative proposals. The differences between the positive and negative proposals are whether they contain sufficient information (overlap ratio with the ground truth bounding box) to describe objects. If we look at the activation map in Fig 1, it is obvious that the positive bounding boxes spread much wider than the core regions. As a result, we hypoth-

---

esize that *the positive object proposals that lay around the core regions can be helpful for image classification since they contain partial information of the objects in the image.* However, the challenges in improving image classification
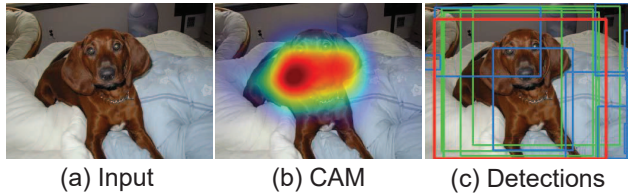

| (a) Input | (b) CAM | (c) Detections |

Figure 1. Visualization of class activation map (CAM [54]) and weakly supervised object detections.

by detection are two-fold. First, how can we perform object detection without groundtruth bounding box annotations? Second, how can we exploit object detection results to boost the performance of image classification? In this paper, we attempt to tackle these two challenges in a weakly supervised manner.

To avoid missing any important object parts, we propose a weakly supervised object detection pipeline regularized by iterative object instance segmentation. We start by training a deep classification neural network that produces a class activation map (CAM) as in [54]. Then the activations in CAM are taken as the pixelwise probabilities of the corresponding class. A conditional random field (CRF) [40] then incorporates low level pairwise appearance information to perform unsupervised object instance segmentation. To refine object locations and pixel labels, a Mask R-CNN [16] is trained using the object instance masks from the CRF. Results from the Mask R-CNN are used as a pixel probability map to replace the CAM in the CRF. We alternate Mask R-CNN and CRF regularization a few times to generate the final object instance masks.

Image Classification Phase. Directly reporting classification results in the object detection phase gives rise to inferior performance because object detection algorithms make much effort to determine location in addition to class labels. In order to mine representative object parts with the help of object detection, we utilize the proposals generated in the previous object detection phase and build a complementary parts model, which consists of a subset of the proposals that cover as much complementary object information as possible. At the end, we exploit a bi-directional long short-term memory network to encode the deep features of the object parts for final image classification.

In summary, this paper has the following contributions:

• We introduce a new representation for image classification, called weakly supervised complementary parts model, that attempts to grasp complete object descriptions using a selected subset of object proposals. It is an important step

forward in exploiting weakly supervised detection to boost image classification performance.

• We develop a novel pipeline for weakly supervised object detection and instance segmentation. Specifically, we iterate the following two steps, object detection and segmentation using Mask R-CNN, and instance segmentation enhancement using CRF. In this way, we get strong object detection results and build accurate object part model.

• To encode complementary information in different object parts, we exploit a bi-directional long short-term memory network to make the final classification decision. Experimental results demonstrate that we achieve state-of-the-art performance on multiple image classification tasks, including fine-grained classification on Stanford Dogs 120 [21] and Caltech-UCSD Birds 200-2011 [47], and generic classification on Caltech 256 [15].

2. Related Work

Weakly Supervised Object Detection and Segmentation. Weakly supervised object detection and segmentation respectively locates and segments objects with image label only [5]. In [7, 6], the object detection is solved as a classification problem by specific pooling layers in CNNs. The method in [44] proposed an iterative bottom-up and top-down framework to expand object regions and optimize segmentation network iteratively. Ge *et al.* in [12] progressively mine the object locations and pixel labels with the filtering and fusion of multiple evidences.

While here we perform the weakly supervised object instance detection and segmentation by feeding a coarse segmentation mask and proposal for Mask R-CNN [16] using CAM [54] and rectifying the object locations and masks with CRF [40] iteratively. In this way, we avoid losing important object parts for subsequent object parts modeling.

Part Based Fine-grained Image Classification. Learning a diverse collection of discriminative parts in a supervised[51, 50] or unsupervised manner [35, 52, 26] is very popular in fine-grained image classification. Many works [51, 50] have been done to build object part models with part bounding box annotations. The method in [51] builds two deformable part models [10] to localize objects and discriminative parts. Zhang *et al.* in [50] treats objects and semantic parts equally by assigning them in different object classes with R-CNN [14]. Another line of works [35, 52, 26, 44] estimate the part location in a unsupervised setting. In [35], parts are discovered based the neural activation, and then are optimized using a EM similar algorithm. The work in [35] extracts the highlight responses in CNN as the part prior to initialize convolutional filters, and then learn discriminative patch detectors end-to-end.

In this paper, we do not aim to build strong part detectors to provide local appearance information for the final classification decision. The goal of our complementary parts model is to efficiently utilize the rich information hidden in the object proposals produced during object detection phase.

Context Encoding with LSTM. LSTM network shows its powerfulness in encoding the context information for image classification. In [26], Lam *et al.* address fine-grained image classification by mining informative image parts using a heuristic network, a successor network and a single layer LSTM. The heuristic network is responsible for extracting features from proposals and the successor network is responsible for predicting the new proposal offset. A single layer LSTM is used to fuse the information both for final object class prediction and also for the offset prediction. Attentional regions is discovered recurrently by incorporating a LSTM sub-network for multi-label image classification in [46]. The LSTM sub-network sequentially predict semantic labeling scores on the located regions and captures the spatial dependencies at the same time.

LSTM is used in our complementary part model to integrate the rich information hidden in different object proposals detected. Different from the single direction LSTM in [26, 46], we exploit a bi-directional LSTM to learn deep hierachical representation of all image patches. Experimental results show this strategy improve the performance substantially compared to the single layer LSTM.

3. Weakly Supervised Complementary Parts Model

3.1. Overview

Given an image I and its corresponding image label c, the method proposed in this paper aims to mine discriminative parts $\mathcal{M}$ of an object that capture complementary information via object detection and then fuse the mined complementary parts for image classification. This is a reversal of a current trend [16, 32, 29], which fine-tunes image classification models for object detection. Since we do not have labeled part locations but image level labels only, we formulate our problem in a weakly supervised manner. We adopt an iterative refinement pipeline to improve the estimation of object parts. Then we build a classifier utilizing the rich context representation focusing on object parts to boost classification performance. We decompose our pipeline into three stages, as shown in Fig 2, namely, weakly supervised object detection and instance segmentation, complementary part model mining and image classification with context encoding.

3.2. Weakly Supervised Object Detection and Instance Segmentation

Coarse Object Mask Initialization. Given an image I and its image label c, the feature map of the last convolutional layer of a classification network is denoted as $\phi(I, \theta) \in \mathbb{R}^{K \times h \times w}$, where θ represents the parameters of network ϕ, K is the number of channels, h and w are the height and width of the feature map respectively. Next, global average pooling is performed on ϕ to obtain the pooled feature $F_k = \sum_{x,y} \phi_k(x, y)$. The classification layer is added at the end and thus, the class activation map (CAM) for class c is given as follows,

$$M_c(x, y) = \sum_k w_k^c \phi_k(x, y), \qquad (1)$$

where w_k^c is the weight corresponding to class c for the k-th channel in the global average pooling layer. The obtained class activation map M_c is upsampled to the original image size $\mathbb{R}^{H \times W}$ through bilinear interpolation. Since an image could have multiple object instances, multiple locally maximum responses could be observed on the class activation map M_c. We apply multi-region level set segmentation [3] to this map to segment candidate object instances. Next, for each instance, we normalize the class activation to the range, $[0, 1]$. Suppose we have n object instances in CAM, we set up an object probability map $F \in \mathbb{R}^{(n+1) \times H \times W}$ according to the normalized CAM. The first n object probability maps denote the probability of a certain object existing in the image and the $(n + 1)$-th probability map represents the probability of the background. The background probability map is calculated as

$$F_{i \in \mathbb{R}^{H \times W}}^{n+1} = \max(1 - \sum_{\iota=1}^{n} F_{i \in \mathbb{R}^{H \times W}}^{\iota}, 0). \qquad (2)$$

Then a conditional random field (CRF) [40] is used to extract higher-quality object instances. In order to apply CRFs, a label map L is generated according to the following formula,

$$L_{i \in \mathbb{R}^{H \times W}} = \begin{cases} \lambda, \arg \max_\lambda F_{i \in \mathbb{R}^{H \times W}}^\lambda > \sigma_c \\ 0, otherwise \end{cases} \qquad (3)$$

where σ_c is always set to 0.8, a fixed threshold used to determine how certain a pixel belongs to an object or background. The label map L is then fed into a CRF to generate object instance segments, that are treated as pseudo groundtruth annotations for Mask-RCNN training. The parameters in the CRF are the same as in [23]. Fig 2 stage 1 shows the whole process of object instance segmentation.

Jointly Detect and Segment Object Instances. Given a set of segmented object instances, $\mathcal{S} = [\mathcal{S}_1, \mathcal{S}_2, ... \mathcal{S}_n]$ of I, and their corresponding class labels, generated in the previous

Stage 1: Weakly Supervised Instance Detection and Segmentation

Class Activation Map *Initial Segmentation*   *Mask R-CNN*   *Probability Map* *Instance Segmentation*

CRFs   CRFs

Stage 2: Complementary Parts Model

Redundant Object Proposals   Constrained Object Part Model

NMS   Searching

Input   *Weakly Supervised Object Detection*   *Detections*   *Complementary Part Model*   *Complementary Parts*

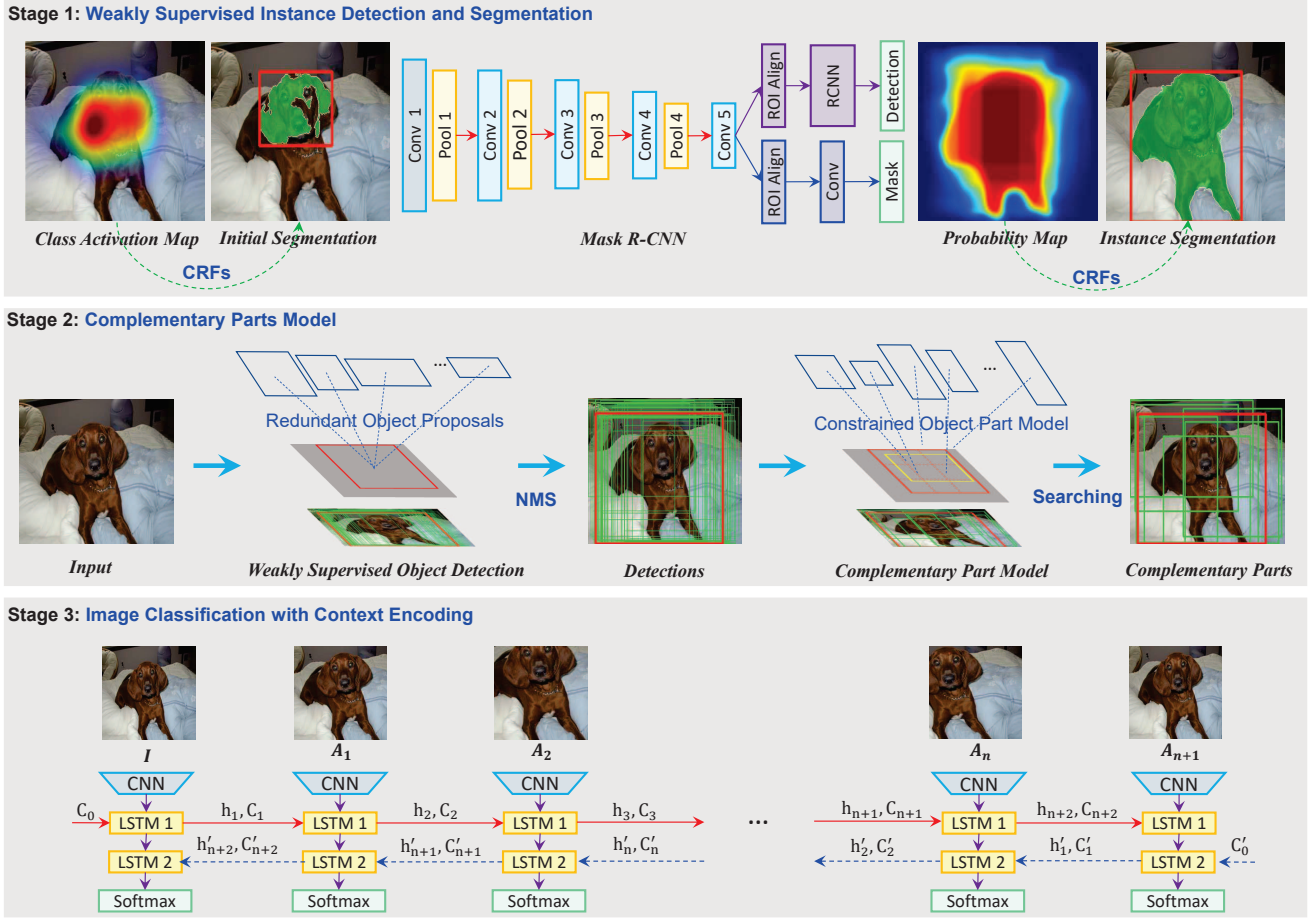**Stage 3: Image Classification with Context Encoding**

Figure 2. The proposed image classification pipeline based on weakly supervised complementary parts model. From top to bottom: (a) Weakly Supervised Object Detection and Instance Segmentation: The first step initializes the segmentation probability map by CAM [54], and obtaining coarse instance segmentation maps by CRF [40]. Then the segments and bounding boxes are used as groundtruth annotations for training Mask R-CNN [16] in an iterative manner. (b) Complementary Parts Model: Search for complementary object proposals to form the object parts model. (c) Image Classification with Context Encoding: Two LSTMs [18] are stacked together to fuse and encode the partial information provided by different object parts.

stage, we obtain the minimum bounding box of each segment to form a set of proposals, $\mathcal{P} = [\mathcal{P}_1, \mathcal{P}_2, ...\mathcal{P}_n]$. The proposals $\mathcal{P}$, segments $\mathcal{S}$ and their corresponding class labels are used for training Mask R-CNN for further proposal and mask refinement. In this way, we turn object detection and instance segmentation into fully supervised learning. We train Mask R-CNN with the same setting as in [16].

CRF-Based Segmentation. Suppose there are m object proposals, $\mathcal{P}^\star = [\mathcal{P}_1^\star, \mathcal{P}_2^\star, ..., \mathcal{P}_m^\star]$, and their corresponding segments, $\mathcal{S}^\star = [\mathcal{S}_1^\star, \mathcal{S}_2^\star, ..., \mathcal{S}_m^\star]$ for image class c, whose classification score is above σ_0, a threshold used to remove outlier proposals. Then, a non-maximum suppression (N-MS) procedure is applied to m proposals with overlapping threshold τ. Suppose n object proposals remain afterwards, $\mathcal{O} = [\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_n]$, where $n \ll m$.

Most existing research utilizes NMS to suppress a large number of proposals sharing the same class label in order to

obtain a small number of distinct object proposals. However, in our weakly supervised setting, proposals suppressed in the NMS process actually contain rich object parts information as shown in Fig 2. Specifically, each proposal $\mathcal{P}_i^\star \in \mathcal{P}^\star$ suppressed by object proposal $\mathcal{O}_j$ can be considered as a complementary part of $\mathcal{O}_j$. Therefore, the suppressed proposals, $\mathcal{P}_i^\star$, can be used to further refine $\mathcal{O}_j$. We implement this idea by initializing a class probability map $F^\star \in \mathbb{R}^{(n+1) \times H \times W}$. For each proposal $\mathcal{P}_i^\star$ suppressed by $\mathcal{O}_j$, we add the probability map of its proposal segmentation mask $\mathcal{S}_i^\star$ to the corresponding locations on $F_j^\star$ by bilinear interpolation. The class probability map is then normalized to [0, 1]. For the $(n+1)$-th probability map for the background, it is defined as

$$F_{i \in \mathbb{R}^{H \times W}}^{\star, n+1} = \max(1 - \sum_{\iota=1}^{n} F_{i \in \mathbb{R}^{H \times W}}^{\star, \iota}, 0). \qquad (4)$$

Given the class probability maps $\boldsymbol{F}^{\star}$, CRF is applied again to refine and rectify instance segmentation results as described in the previous stage.

Iterative Instance Refinement. We alternate CRF-based segmentation and Mask R-CNN based detection and instance segmentation several times to gradually refine the localization and segmentation of object instances. Fig 2 shows the iterative instance refinement process.

3.3. Complementary Parts Model

Model Definition. According to the analysis in the previous stage, given a detected object $\mathcal{O}_i$, its corresponding suppressed proposals, $\mathcal{P}^{\star,i} = \left[\mathcal{P}_1^{\star,i}, \mathcal{P}_2^{\star,i}, ..., \mathcal{P}_k^{\star,i}\right]$, may contain useful object information and can localize correct object position. Then, it is necessary to identify the most informative proposals for the following classification task. In this section, we propose a complementary parts model $\mathcal{A}$ for image classification. This model is defined by a root part covering the entire object as well as its context, a center part covering the core region of the object and a fixed number of surrounding proposals that cover different object parts but still keep enough discriminative information.

A complementary parts model for an object with n parts is defined as a $(n+1)$-tuple $\mathcal{A} = [\boldsymbol{A}_1, ..., \boldsymbol{A}_n, \boldsymbol{A}_{n+1}]$, where $\boldsymbol{A}_1$ is the object center part, $\boldsymbol{A}_{n+1}$ is the root part, and $\boldsymbol{A}_i$ is the i-th part. Each part model is defined by a tuple $\boldsymbol{A}_i = [\phi_i, \boldsymbol{u}_i]$, where ϕ_i is the feature of the i-th part, $\boldsymbol{u}_i$ is a $\mathbb{R}^4$ dimensional tuple that describes the geometric information of a part, namely part center and part size (x_i, y_i, w_i, h_i). A potential parts model without any missing parts is called an object hypothesis. To make object parts complementary to each other, the differences in their appearance features or locations should be as large as possible while the combination of parts scores should also be as large as possible. Such criteria serve as constraints during the search for discriminative parts that are complementary to each other. The score $\mathcal{S}(\mathcal{A})$ of an object hypothesis is given by the summed score of all object parts minus appearance similarities and spatial overlap between different parts.

$$
\begin{aligned}
\mathcal{S}(\mathcal{A}) = &\sum_{\iota=1}^{n+1} f(\phi_\iota) \\
&- \lambda_0 \sum_{p=1}^{n} \sum_{q=p+1}^{n+1} \left[d_s(\phi_p, \phi_q) + \beta_0 IoU(\boldsymbol{u}_p, \boldsymbol{u}_q) \right],
\end{aligned}
\tag{5}
$$

where $f(\phi_k)$ is the score of the k-th part in the classification branch of Mask R-CNN, $d_s(\phi_p, \phi_q) = \|\phi_p - \phi_q\|^2$ is the semantic similarity and $IoU(\boldsymbol{u}_p, \boldsymbol{u}_q)$ is the spatial overlap between parts p and q, and there are two constant parameters $\lambda_0 = 0.01$ and $\beta_0 = 0.1$. Given a set of object hypotheses, we can choose a hypothesis that achieves the maximum

score as the final object part model. Searching for the optimal subset of proposals maximizing the above score is a combinatorial optimization problem, which is computationally expensive. In the following, we seek an approximate solution using a fast heuristic algorithm.

Part Location Initialization. To initialize a parts model, we simplify part estimation by designing a grid-based object parts template that follows two basic rules. First, every part should contain enough discriminative information; Second, the differences between part pairs should be as large as possible. As shown in Fig 2, deep convolutional neural networks have demonstrated its ability in localizing the most discriminative parts of an object. Thus, we set the root part $\boldsymbol{A}_{n+1}$ to be the object proposal $\mathcal{O}_i$ that represents the entire object. Then, a $s \times s(= n)$ grid centered at $\boldsymbol{A}_{n+1}$ is created. The size of each grid cell is $\frac{w_{n+1}}{s} \times \frac{h_{n+1}}{s}$, where w_{n+1} and h_{n+1} are the width and height of the root part $\boldsymbol{A}_{n+1}$. The center grid cell is assigned to the object center part. The rest of the grid cells are assigned to part $\boldsymbol{A}_i$, where $i \in [2, 3, ..., n]$. Then, we initialize each part $\boldsymbol{A}_i \in \boldsymbol{A}$ to be the proposal $\mathcal{P}_j^\star \in \mathcal{P}^\star$ closest to the assigned grid cell.

Parts Model Search. For a model with n object parts (we exclude the $(n+1)$-th part as it is a root part) and k candidate suppressed proposals, the objective function is defined as

$$
\widehat{\mathcal{A}} = \underset{\mathcal{A} \in \mathcal{S}_{\mathcal{A}}}{\arg\max} \, \mathcal{S}(\mathcal{A}),
\tag{6}
$$

where $K = C_k^n, k \gg n$ is the total number of object hypotheses, $\mathcal{S}_{\mathcal{A}} = \left[\mathcal{A}^1, \mathcal{A}^1, ..., \mathcal{A}^K\right]$ is the set of object hypotheses. As mentioned earlier, directly searching for an optimal parts model can be intractable. Thus, we adopt a greedy search strategy to search for $\widehat{\mathcal{A}}$. Specifically, we sequentially go through every $\boldsymbol{A}_i$ in $\boldsymbol{A}$ and find the optimal object part for $\boldsymbol{A}_i$ in $\mathcal{P}^\star$ that minimizes $\widehat{\mathcal{A}}$. The overall time complexity is reduced from exponential to linear ($O(nk)$). In Fig 2, we can see that the object hypotheses generated during the search process cover different parts of the object and do not focus on the core region only.

3.4. Image Classification with Context Encoding

CNN Feature Extractor Fine-tuning. Given an input image $\boldsymbol{I}$ and the parts model $\mathcal{A} = [\boldsymbol{A}_1, ..., \boldsymbol{A}_n, \boldsymbol{A}_{n+1}]$ constructed in the previous stage, the image patches corresponding to the parts are denoted as $\boldsymbol{I}(\mathcal{A}) = [\boldsymbol{I}(\boldsymbol{A}_1), \boldsymbol{I}(\boldsymbol{A}_2), ..., \boldsymbol{I}(\boldsymbol{A}_n), \boldsymbol{I}(\boldsymbol{A}_{n+1})]$. During image classification, random crops of images are often used to train the model. Thus, apart from the $(n+1)$ patches, we append a random crop of the original image as the $(n+2)$-nd image patch. The motivation for adding a randomly cropped patch is to include more context information during training since those patches corresponding to object parts primarily focus on the object itself. Every patch shares the same label with the original image it is cropped from. All patches
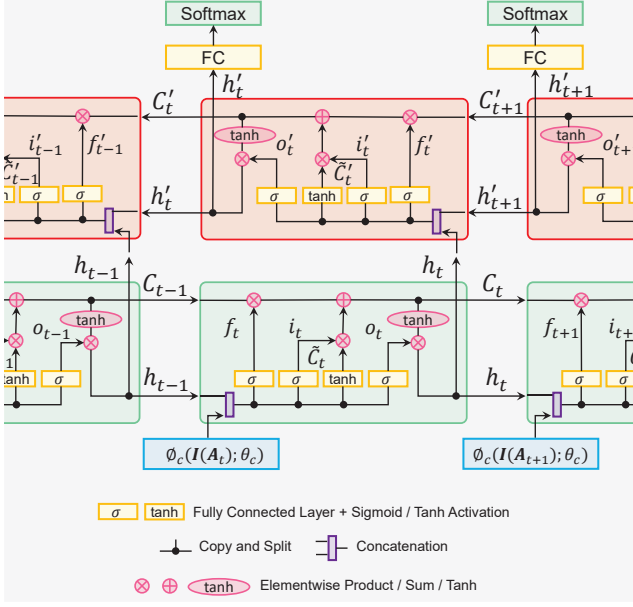
Figure 3. Context encoded image classification based on LSTMs. Two standard LSTMs [18] are stacked together. They have opposite scanning orders.

from all the original training images form a new training set, which is used to fine-tune a CNN model pretrained on ImageNet. This fine-tuned model serves as the feature extractor for all image patches.

Stacked LSTM for Feature Fusion. Here we propose a stacked LSTM module $\phi_l(\cdot; \theta_l)$ for feature fusion and performance boosting, which is shown in Fig 3. First, the $(n + 2)$ patches from a complementary parts model are fed through the CNN feature extractor $\phi_c(\cdot; \theta_c)$ trained in the previous step. The output from this step is denoted as $\Psi(I) = [\phi_c(I; \theta_c), \phi_c(I(A_1); \theta_c), ..., \phi_c(I(A_{n+2}); \theta_c)]$. Next, we build a two-layer stacked LSTM to fuse the extracted features $\Psi(I)$. The hidden state of the first LSTM is fed into the second LSTM layer, but the second LSTM follows the reversed order of the first one. Let $D(= 256)$ be the dimension of the hidden state. We use softmax to generate the class probability vector for each part A_i, $f(\phi_l(I(A_i); \theta_l)) \in \mathbb{R}^{\mathcal{C} \times 1}$. The loss function for final image classification is defined as follows,

$$
\begin{aligned}
\mathcal{L}(I, y_I) = & - \sum_{k=1}^{\mathcal{C}} y^k \log f^k(\phi_l(I; \theta_l)) \\
& - \sum_{i=1}^{n+2} \sum_{k=1}^{\mathcal{C}} \gamma_i y^k \log f^k(\phi_l(I(A_i); \theta_l)),
\end{aligned}
\tag{7}
$$

where $f^k(\phi_l(I; \theta_l))$ is the probability that image I belongs to the k-th class, $f^k(\phi_l(I(A_i); \theta_l))$ is the probability that image patch $I(A_i)$ belongs to the k-th class, and γ_i is a

constant weight for the i-th patch. Here we have two settings: first, the single loss sets $\gamma_i = 0 \, (i = 2, ..., n + 2)$, and keeps only one loss at the start of the sequence; second, the multiple losses sets $\gamma_i = 1 \, (i = 2, ..., n + 2)$. Experimental results indicate that, in comparison to a single loss for the last output from the second LSTM, multiple losses used here improve classification accuracy by a significant margin.

4. Experimental Results

4.1. Implementation Details

All experiments have been conducted on NVIDIA TITAN X(Maxwell) GPUs with 12GB memory using Caffe [20]. No annotated parts are used. n is set to 9 for all experiments.

In the mask initialization stage, we fine-tune from ImageNet pre-trained GoogleNet with batch normalization [19] on target datasets. The initial learning rate is 0.001 and is divided by 10 after every 40000 iterations with the standard SGD optimizer. Training converges after 70000 iterations. In the Mask R-CNN refinement process, we adopt ResNet-50 with Feature Pyramid Network (FPN) as the backbone and pre-train the network on the COCO dataset following the same setting described in [16]. We then fine-tune the model on our target datasets. During training, image-centric training is used and the input images are resized such that their shorter side is 800 pixels. Each mini-batch contains 1 image per GPU and each image has 512 sampled ROIs. The model is trained on 4 GPUs for 150k iterations with an initial learning rate 0.001, which is divided by 10 at 120k iterations. We use the standard SGD optimizer and a weight decay of 0.0001. The momentum is set to 0.9. Unless specified, the settings we use for different algorithms follow their original settings respectively [54, 41, 3, 23, 16]. Example intermediate results of Mask R-CNN training are shown in Fig 4.
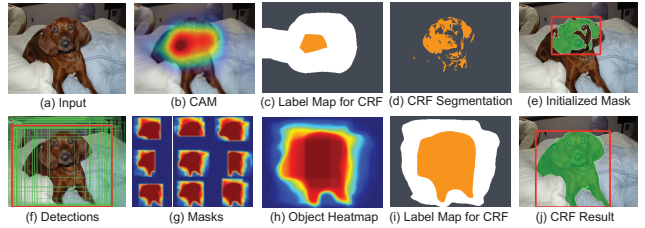

Figure 4. Example intermediate results for training Mask R-CNN. *First row*: pseudo object mask and object bounding box are generated with CAM and CRF refinement. *Second row*: With previous pseudo groundtruth generated, object mask and object bounding box are further refined with Mask R-CNN.

For the last stage, we adopt GoogleNet with batch normalization [19] as the backbone network for Stanford Dogs 120 and Caltech-UCSD Birds 2011-200 datasets and the

Caltech256 dataset. First, we fine-tune the pretrained network on the target dataset with the generated object parts. The parameters are the same as those used in the first stage. Next, we build a Stacked LSTM module and treat the features of the $n + 2$ image patches as training sequences. We train the model with 4 GPUs and set the learning rate to 0.001, which is decreased by a factor of 10 for every 8000 iterations. We adopt the standard SGD optimizer, momentum is set to 0.9, and the weight decay is 0.0002. Training converges at 16000 iterations.

4.2. Fine-grained Image Classification

Stanford Dogs 120. Stanford Dogs 120 contains 120 categories of dogs. There are 12000 images for training, and 8580 images for testing. The training procedure follows the steps described in Section 4.1.

To perform fair comparisons with existing state-of-the-art algorithms, we divide our experiments into two groups. The first group consists of algorithms that use the original training data only and the second group is composed of methods that use extra training data. In each group, we set our baseline accordingly. In the first group, we directly fine-tune the GoogleNet pretrained on ImageNet with the input image size set to 448 x 448, which is adopted by other algorithms [11, 30, 39] in the comparison and the classification accuracy achieved is 85.2%. This serves as our baseline model and we then add the proposed stacked LSTM over a complementary parts model. Our stacked LSTM is trained with both single loss and multiple losses, which achieves a classification accuracy of 92.4% and 93.9% respectively. Both of our proposed variants outperform existing state-of-the-art by a clear margin. In the second group, we perform selective joint fine-tuning (SJFT) with images retrieved from ImageNet, and the input image size is set to 224 x 224 to obtain our baseline network. The classification accuracy of our baseline is 92.1%, 1.8% higher than the SJFT with ResNet-152 counterpart. With our stacked LSTM plugged in and trained with both single loss and multiple losses, the performance is further boosted to 96.3% and 97.1% respectively, surpassing the current state of the art by 6% and 6.8%. These experimental results suggest that our proposed pipeline is superior than all existing algorithms. It is worth noting that the method in [24] is not directly comparable to ours because it uses a large amount of extra training data from the Internet in addition to ImageNet.

Caltech-UCSD Birds 2011-200. Caltech-UCSD Birds 2011-200 (CUB200) consists of 200 bird categories. 5994 images are used for training, and 5794 images for testing.

Our experiments here are split into two groups. In the first group, no extra training data is used. Our baseline model in this group is a directly fine-tuned GoogleNet model that achieves a classification accuracy of 82.6%. We then add the Stacked LSTM module and train the model

Method	Accuracy(%)
MAMC [39]	85.2
Inception-v3 [24]	85.9
RA-CNN [11]	87.3
FCAN [30]	88.9
GoogleNet (our baseline)	85.2
baseline + Feature Concatenation	88.1
baseline + Multiple Average	85.2
baseline + Stacked LSTM + Single Loss	92.4
baseline + Stacked LSTM + Multi-Loss (default)	**93.9**
Web Data + Original Data [24]	85.9
SJFT with ResNet-152 [13]	90.3
SJFT with GoogleNet (our baseline)	92.1
baseline + Feature Concatenation	93.2
baseline + Multiple Average	92.2
baseline + Stacked LSTM + Single Loss	96.3
baseline + Stacked LSTM + Multi-Loss (default)	**97.1**

Table 1. Classification results on Stanford Dogs 120. Two sections are divided by the horizontal separators, namely (from top to bottom) Experiments without SJFT and Experiments with SJFT.

with both single loss and multiple losses, which achieves a classification accuracy of 87.6% and 90.3% respectively, outperforming all other algorithms in this comparison [53, 48, 45, 27]. Compared to HSNet, our model does not use any parts annotations in the training stage while HSNet is trained with groundtruth parts annotations. In the second group, our baseline model still uses GoogleNet as the backbone and performs SJFT with images retrieved from ImageNet. It achieves a classification accuracy of 82.8%. By adding the Stacked LSTM module, the accuracy of the model trained with single loss is 87.7% and the model trained with multiple losses is 90.4%. When the top performing result in the first group is compared to that of the second group, it can be concluded that SJFT contributes little to the performance gain (0.1% gains) and our proposed method is effective and solid, contributing much to the final performance (7.7% higher than the baseline). It is worth noting that, in [4], a subset of ImageNet and iNaturalist [43] most similar to CUB200 are used for training, and in [24], a large amount of web data are also used in the training phase.

4.3. Generic Object Recognition

Caltech 256. There are 256 object categories and 1 background cluster class in Caltech 256. A minimum number of 80 images per category are provided for training, validation and testing. As a convention, results are reported with the number of training samples per category falling between 5 and 60. We follow the same convention and report the result with the number of training sample per category set to 60. In this experiment, GoogleNet is adopted as our backbone network and the input image size is 224 x 224. We train our

Method	Accuracy(%)
MACNN [53]	86.5
HBP [48]	87.2
DFB [45]	87.4
HSNet [27]	87.5
GoogleNet (our baseline)	82.6
baseline + Stacked LSTM + Single Loss	87.6
baseline + Stacked LSTM + Multi-Loss	**90.3**
ImageNet + iNat Finetuning [4]	89.6
SJFT with GoogleNet (our baseline)	82.8
baseline + Stacked LSTM + Single Loss	87.7
baseline + Stacked LSTM + Multi-Loss	**90.4**

Table 2. Classification results on CUB200. Two sections are divided by the horizontal separators, namely (from top to bottom) Experiments without SJFT and Experiments with SJFT.

Method	Accuracy(%)
ZF Net [49]	74.2±0.3
VGG-19 + VGG-16 [36]	86.2±0.3
VGG-19 + GoogleNet +AlexNet [22]	86.1
L^2-SP [28]	87.9±0.2
GoogleNet (our baseline)	84.1±0.2
baseline + Stacked LSTM + Single Loss	90.1±0.2
baseline + Stacked LSTM + Multi-Loss	**93.5±0.2**
SJFT with ResNet-152 [13]	89.1±0.2
SJFT with GoogleNet (our baseline)	86.3±0.2
baseline + Stacked LSTM + Single Loss	90.1±0.2
baseline + Stacked LSTM + Multi-Loss	**94.3±0.2**

Table 3. Classification results on Caltech 256. Two sections are divided by the horizontal separators, namely (from top to bottom) Experiments without SJFT and Experiments with SJFT.

model with mini-batch size set to 8 on each GPU.

In Table 3, as described previously, we conduct our experiments under two settings. For the first setting, no extra training data is used. We fine-tune the pretrained GoogleNet on the target dataset and treat the fine-tuned model as our baseline model, which achieves a classification accuracy of 84.1%. By adding our proposed Stacked LSTM module, the accuracy is increased by a large margin to 90.1% for Single Loss and to 93.5% for multiple losses respectively, outperforming all methods listed in the table. Also, it is 4.1% higher than its ResNet-152 counterpart. For the second setting, we adopt SJFT [13] with GoogleNet as our baseline model, which achieves a classification accuracy of 86.3%. Then we add our proposed Stacked LSTM module and the final performance is increased by 3.8% for single loss and 8.0% for multiple losses. Our method with GoogleNet as backbone network outperfoms current state-of-the-art by 5.2%, demonstrating that our proposed algorithm is solid and effective.

4.4. Ablation Study

Ablation Study on Complementary Parts Mining. The ablation study is performed on the CUB200 dataset with GoogleNet as the backbone network. The classification accuracy of our reference model with $n = 9$ parts on this dataset is 90.3%. First, when the number of parts n is set to 2, 4, 6, 9, 12, 16, and 20 in our model, the corresponding classification accuracy is respectively 85.3%, 87.9%, 89.1%, 90.3%, 87.6%, 86.8% and 85.9%. Obviously the best result is achieved when $n = 9$. Second, if we use object features only in our reference model, the classification accuracy drops to 90.0%. Third, if we use image features only, the performance drops to 82.8%. Fourth, if we simply use the uniform grid cells as the object parts without further optimization, the performance drops to 78.3%, which indicates our search for the best parts model plays an important role in escalating the performance. Fifth, instead of a grid-based object parts initialization, we randomly sample $n = 9$ suppressed object proposals around the bounding box of the surviving proposal, and the performance drops to 86.9%. Lastly, we discover that the part order in LSTM does not matter. We randomly shuffle the part order during training and testing, and the classification accuracy remains the same.

4.5. Inference Time Complexity.

The inference time of our implementation is summarised as follows: in the complementary parts model search phase, the time for processing an image with its shorter edge set to 800 pixels is around $277ms$; in the context encoding phase, the running time on an image of size 448×448 is about $63ms$, and on an image of size 224×224 is about $27ms$.

5. Conclusions

In this paper, we have presented a new pipeline for fine-grained image classification, which is based on a complementary part model. Different from previous work which focuses on learning the most discriminative parts for image classification, our scheme mines complementary parts that contain partial object descriptions in a weakly supervised manner. After getting object parts that contain rich information, we fuse all the mined partial object descriptions with bi-directional stacked LSTM to encode these complementary information for classification. Experimental results indicate that the proposed method is effective and outperforms existing state-of-the-art by a large margin. Nevertheless, how to build the complementary part model in a more efficient and accurate way remains an open problem for further investigation.

References

[1] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech

Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 2015.

[2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[3] Thomas Brox and Joachim Weickert. Level set segmentation with multiple regions. *IEEE Transactions on Image Processing*, 2006.

[4] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge J. Belongie. Large scale fine-grained categorization and domain-specific transfer learning. 2018.

[5] Ali Diba, Vivek Sharma, Ali Mohammad Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017.

[6] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.

[7] Thibaut Durand, Nicolas Thome, and Matthieu Cord. Weldon: Weakly supervised learning of deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[8] Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid. Blitznet: A real-time deep network for scene understanding. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 2010.

[10] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[11] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017.

[12] Weifeng Ge, Sibei Yang, and Yizhou Yu. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[13] W. Ge and Y. Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[15] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. pages 448–456, 2015.

[20] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014.

[21] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, 2011.

[22] Yong-Deok Kim, Taewoong Jang, Bohyung Han, and Seungjin Choi. Learning to select pre-trained deep representations with bayesian evidence framework. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016.

[23] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc., 2011.

[24] Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Fei-Fei Li. The unreasonable effectiveness of noisy data for fine-grained recognition. 2015.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012.

[26] Michael Lam, Behrooz Mahasseni, and Sinisa Todorovic. Fine-grained recognition as hsnet search for informative image parts. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017.

[27] Michael Lam, Behrooz Mahasseni, and Sinisa Todorovic. Fine-grained recognition as hsnet search for informative image parts. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017.

[28] Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *ICML*, 2018.

[29] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[30] Xiao Liu, Tian Xia, Jiang Wang, and Yuanqing Lin. Fully convolutional attention localization networks: Efficient attention localization for fine-grained recognition. 2016.

[31] Cewu Lu, Hao Su, Yonglu Li, Yongyi Lu, Li Yi, Chi-Keung Tang, and Leonidas J. Guibas. Beyond holistic object recognition: Enriching image understanding with part states. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 2015.

[33] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[35] Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[38] Chong Sun, Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Learning spatial-aware regressions for visual tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding. Multi-attention multi-class constraint for fine-grained image recognition. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, 2018.

[40] Charles Sutton, Andrew McCallum, et al. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4), 2012.

[41] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,*, 2016.

[42] Zhu Teng, Junliang Xing, Qiang Wang, Congyan Lang, Songhe Feng, and Yi Jin. Robust object tracking based on temporal and spatial deep networks. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[43] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[44] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[45] Yaming Wang, Vlad I. Morariu, and Larry S. Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[47] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[48] Chaojian Yu, Xinyi Zhao, Qi Zheng, Peng Zhang, and Xinge You. Hierarchical bilinear pooling for fine-grained visual recognition. In *The European Conference on Computer Vision (ECCV)*, 2018.

[49] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, 2014.

[50] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[51] Ning Zhang, Ryan Farrell, Forrest Iandola, and Trevor Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[52] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017.

[53] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017.

[54] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.