Learning to Navigate for Fine-grained Classification

Ze Yang[1][0000−0002−6299−7649], Tiange Luo[1], Dong Wang[1], Zhiqiang Hu[1], Jun Gao[1], and Liwei Wang[1,2]

[1] Key Laboratory of Machine Perception, MOE, School of EECS, Peking University.
[2] Center for Data Science, Peking University, Beijing Institute of Big Data Research.
{yangze,luotg,wangdongcis,huzq,jun.gao}@pku.edu.cn
wanglw@cis.pku.edu.cn

Abstract. Fine-grained classification is challenging due to the difficulty of finding discriminative features. Finding those subtle traits that fully characterize the object is not straightforward. To handle this circumstance, we propose a novel self-supervision mechanism to effectively localize informative regions without the need of bounding-box/part annotations. Our model, termed NTS-Net for Navigator-Teacher-Scrutinizer Network, consists of a Navigator agent, a Teacher agent and a Scrutinizer agent. In consideration of intrinsic consistency between informativeness of the regions and their probability being ground-truth class, we design a novel training paradigm, which enables Navigator to detect most informative regions under the guidance from Teacher. After that, the Scrutinizer scrutinizes the proposed regions from Navigator and makes predictions. Our model can be viewed as a multi-agent cooperation, wherein agents benefit from each other, and make progress together. NTS-Net can be trained end-to-end, while provides accurate fine-grained classification predictions as well as highly informative regions during inference. We achieve state-of-the-art performance in extensive benchmark datasets.

1 Introduction

Fine-grained classification aims at differentiating subordinate classes of a common superior class, *e.g.* distinguishing wild bird species, automobile models, *etc.* Those subordinate classes are usually defined by domain experts with complicated rules, which typically focus on subtle differences in particular regions. While deep learning has promoted the research in many computer vision [24,38,33] tasks, its application in fine-grained classification is more or less unsatisfactory, due in large part to the difficulty of finding informative regions and extracting discriminative features therein. The situation is even worse for subordinate classes with varied poses like birds.

As a result, the key to fine-grained classification lies in developing automatic methods to accurately identify informative regions in an image. Some previous works [45,8,3,46,13,2,29] take advantage of fine-grained human annotations, like annotations for bird parts in bird classification. While achieving decent results,

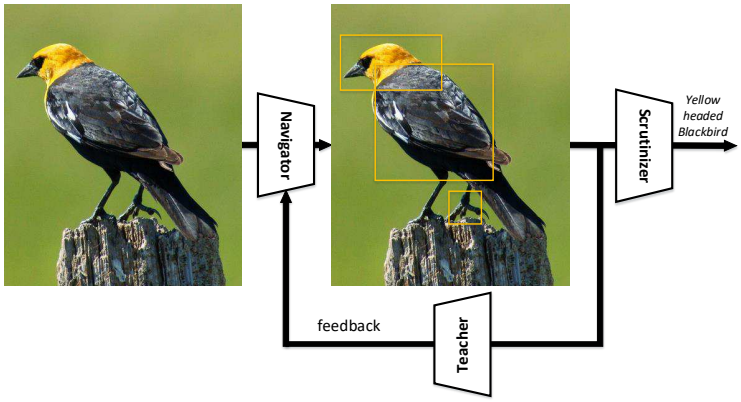

Fig. 1. The overview of our model. The Navigator navigates the model to focus on the most informative regions (denoted by yellow rectangles), while Teacher evaluates the regions proposed by Navigator and provides feedback. After that, the Scrutinizer scrutinizes those regions to make predictions.

the fine-grained human annotations they require are expensive, making those methods less applicable in practice. Other methods [49,47,48,43] employ an unsupervised learning scheme to localize informative regions. They eliminate the need for the expensive annotations, but lack a mechanism to guarantee that the model focuses on the right regions, which usually results in degraded accuracy.

In this paper, we propose a novel self-supervised mechanism to effectively localize informative regions without the need of fine-grained bounding-box/part annotations. The model we develop, which we term NTS-Net for Navigator-Teacher-Scrutinizer Network, employs a multi-agent cooperative learning scheme to address the problem of accurately identifying informative regions in an image. Intuitively, the regions assigned higher probability to be ground-truth class should contain more object-characteristic semantics enhancing the classification performance of the whole image. Thus we design a novel loss function to optimize the informativeness of each selected region to have the same order as its probability being ground-truth class, and we take the ground-truth class of full image as the ground-truth class of regions.

Specifically, our NTS-Net consists of a Navigator agent, a Teacher agent and a Scrutinizer agent. The Navigator navigates the model to focus on the most informative regions: for each region in the image, Navigator predicts how informative the region is, and the predictions are used to propose the most informative regions. The Teacher evaluates the regions proposed by Navigator and provides feedbacks: for each proposed region, the Teacher evaluates its probability belonging to ground-truth class; the confidence evaluations guide the Navigator to propose more informative regions with our novel ordering-consistent loss function. The Scrutinizer scrutinizes proposed regions from Navigator and makes fine-grained classifications: each proposed region is enlarged to the same

size and the Scrutinizer extracts features therein; the features of regions and of the whole image are jointly processed to make fine-grained classifications. As a whole, our method can be viewed as an actor-critic [21] scheme in reinforcement learning, where the Navigator is the actor and the Teacher is the critic. With a more precise supervision provided by the Teacher, the Navigator will localize more informative regions, which in turn will benefit the Teacher. As a result, agents make progress together and end up with a model which provides accurate fine-grained classification predictions as well as highly informative regions. Fig. 1 shows an overview of our methods.

Our main contributions can be summarized as follows:

- We propose a novel multi-agent cooperative learning scheme to address the problem of accurately identifying informative regions in the fine-grained classification task without bounding-box/part annotations.
- We design a novel loss function, which enables Teacher to guide Navigator to localize the most informative regions in an image by enforcing the consistency between regions' informativeness and their probability being ground-truth class.
- Our model can be trained end-to-end, while provides accurate fine-grained classification predictions as well as highly informative regions during inference. We achieve state-of-the-art performance in extensive benchmark datasets.

The remainder of this paper is organized as follows: We will review the related work in Section. 2. In Section. 3 we will elaborate our methods. Experimental results are presented and analyzed in Section. 4 and finally, Section. 5 concludes.

2 Related Work

2.1 Fine-grained classification

There have been a variety of methods designed to distinguish fine-grained categories. Since some fine-grained classification datasets provide bounding-box/part annotations, early works [45,8,2] take advantage of those annotations at both training and inference phase. However in practice when the model is deployed, no human annotations will be available. Later on, some works [3,46] use bounding-box/part annotations only at training phase. Under this setting, the framework is quite similar to detection: selecting regions and then classifying the pose-normalized objects. Besides, Jonathan *et al.* [22] use co-segmentation and alignment to generate parts without part annotations but the bounding-box annotations are used during training. Recently, a more general setting has emerged that does not require bounding box/part annotations either at training or inference time. This setting makes fine-grained classification more useful in practice. This paper will mainly consider the last setting, where bounding-box/part annotations are not needed either at training or inference phase.

In order to learn without fine-grained annotations, Jaderberg *et al.* [19] propose Spatial Transformer Network to explicitly manipulate data representation

within the network and predict the location of informative regions. Lin *et al.* [28] use a bilinear model to build discriminative features of the whole image; the model is able to capture subtle differences between different subordinate classes. Zhang *et al.* [47] propose a two-step approach to learn a bunch of part detectors and part saliency maps. Fu *et al.* [12] use an alternate optimization scheme to train attention proposal network and region-based classifier; they show that two tasks are correlated and can benefit each other. Zhao *et al.* [48] propose Diversified Visual Attention Network (DVAN) to explicitly pursues the diversity of attention and better gather discriminative information. Lam *et al.* [25] propose a Heuristic-Successor Network (HSNet) to formulate the fine-grained classification problem as a sequential search for informative regions in an image.

2.2 Object detection

Early object detection methods employ SIFT [34] or HOG [10] features. Recent works are mainly focusing on convolutional neural networks. Approaches like R-CNN [14], OverFeat [40] and SPPnet [16] adopt traditional image-processing methods to generate object proposals and perform category classification and bounding box regression. Later works like Faster R-CNN [38] propose Region Proposal Network (RPN) for proposal generation. YOLO [37] and SSD [31] improve detection speed over Faster R-CNN [38] by employing a single-shot architecture. On the other hand, Feature Pyramid Networks (FPN) [27] focuses on better addressing multi-scale problem and generates anchors from multiple feature maps. Our method requires selecting informative regions, which can also be viewed as object detection. To the best of our knowledge, we are the first one to introduce FPN into fine-grained classification while eliminates the need of human annotations.

2.3 Learning to rank

Learning to rank is drawing attention in the field of machine learning and information retrieval [30]. The training data consist of lists of items with assigned orders, while the objective is to learn the order for item lists. The ranking loss function is designed to penalize pairs with wrong order. Let $X = \{X_1, X_2, \cdots, X_n\}$ denote the objects to rank, and $Y = \{Y_1, Y_2, \cdots, Y_n\}$ the indexing of the objects, where $Y_i \geq Y_j$ means X_i should be ranked before X_j. Let $\mathbb{F}$ be the hypothesis set of ranking function. The goal is to find a ranking function $\mathcal{F} \in \mathbb{F}$ that minimize a certain loss function defined on $\{X_1, X_2 \cdots X_n\}$, $\{Y_1, Y_2, \cdots, Y_n\}$ and $\mathcal{F}$. There are many ranking methods. Generally speaking, these methods can be divided into three categories: the point-wise approach [9], pair-wise approach [18,4] and list-wise approach[6,44].

Point-wise approach assign each data with a numerical score, and the learning-to-rank problem can be formulated as a regression problem, for example with $L2$ loss function:

$$L_{point}(\mathcal{F}, X, Y) = \sum_{i=1}^{n} (\mathcal{F}(X_i) - Y_i)^2 \tag{1}$$

In the pair-wise ranking approach, the learning-to-rank problem is formulated as a classification problem. *i.e.* to learn a binary classifier that chooses the superiority in a pair. Suppose $\mathcal{F}(X_i, X_j)$ only takes a value from $\{1, 0\}$, where $\mathcal{F}(X_i, X_j) = 0$ means X_i is ranked before X_j. Then the loss is defined on all pairs as in Eqn. 2, and the goal is to find an optimal $\mathcal{F}$ to minimize the average number of pairs with wrong order.

$$L_{pair}(\mathcal{F}, X, Y) = \sum_{(i,j):Y_i<Y_j} \mathcal{F}(X_i, X_j) \tag{2}$$

List-wise approach directly optimizes the whole list, and it can be formalized as a classification problem on permutations. Let $\mathcal{F}(X, Y)$ be the ranking function, the loss is defined as:

$$L_{list}(\mathcal{F}, X, Y) = \begin{cases} 1, & \text{if } \mathcal{F}(X) \neq Y \\ 0, & \text{if } \mathcal{F}(X) = Y \end{cases} \tag{3}$$

In our approach, our navigator loss function adopts from the multi-rating pair-wise ranking loss, which enforces the consistency between region's informativeness and probability being ground-truth class.

3 Methods

3.1 Approach Overview

Our approach rests on the assumption that informative regions are helpful to better characterize the object, so fusing features from informative regions and the full image will achieve better performance. Therefore the goal is to localize the most informative regions of the objects. We assume all regions[3] are rectangle, and we denote $\mathbb{A}$ as the set of all regions in the given image[4]. We define information function $\mathcal{I} : \mathbb{A} \to (-\infty, \infty)$ evaluating how informative the region $R \in \mathbb{A}$ is, and we define the confidence function $\mathcal{C} : \mathbb{A} \to [0, 1]$ as a classifier to evaluate the confidence that the region belongs to ground-truth class. As mentioned in Sec. 1, more informative regions should have higher confidence, so the following condition should hold:

- Condition. 1: for any $R_1, R_2 \in \mathbb{A}$, if $\mathcal{C}(R_1) > \mathcal{C}(R_2)$, $\mathcal{I}(R_1) > \mathcal{I}(R_2)$

We use Navigator network to approximate information function $\mathcal{I}$ and Teacher network to approximate confidence function $\mathcal{C}$. For the sake of simplicity, we choose M regions $\mathbb{A}_M$ in the region space $\mathbb{A}$. For each region $R_i \in \mathbb{A}_M$, the Navigator network evaluates its informativeness $\mathcal{I}(R_i)$, and the Teacher network evaluates its confidence $\mathcal{C}(R_i)$. In order to satisfy Condition. 1, we optimize Navigator

[3] Without loss of generality, we also treat full image as a region

[4] Notation: we use $\mathcal{C}$alligraphy font to denote mapping, $\mathbb{B}$lackboard bold font to denote special sets, And we use $\mathbf{B}$old font to denote parameters in network.

network to make $\{\mathcal{I}(R_1), \mathcal{I}(R_2), \cdots, \mathcal{I}(R_M)\}$ and $\{\mathcal{C}(R_1), \mathcal{C}(R_2), \cdots, \mathcal{C}(R_M)\}$ having the same order.

As the Navigator network improves in accordance with the Teacher network, it will produce more informative regions to help Scrutinizer network make better fine-grained classification result.

In Section. 3.2, we will describe how informative regions are proposed by Navigator under Teacher's supervision. In Section. 3.3, we will present how to get fine-grained classification result from Scrutinizer. In Section. 3.4 and 3.5, we will introduce the network architecture and optimization in detail, respectively.

3.2 Navigator and Teacher

Navigating to possible informative regions can be viewed as a region proposal problem, which has been widely studied in [41,11,1,7,20]. Most of them are based on a sliding-windows search mechanism. Ren *et al.* [38] introduce a novel region proposal network (RPN) that shares convolutional layers with the classifier and mitigates the marginal cost for computing proposals. They use anchors to simultaneously predict multiple region proposals. Each anchor is associated with a sliding window position, aspect ratio, and box scale. Inspired by the idea of anchors, our Navigator network takes an image as input, and produce a bunch of rectangle regions $\{R'_1, R'_2, \ldots R'_A\}$, each with a score denoting the informativeness of the region (Fig. 2 shows the design of our anchors). For an input image X of size 448, we choose anchors to have scales of {48, 96, 192} and ratios {1:1, 3:2, 2:3}, then Navigator network will produce a list denoting the informativeness of all anchors. We sort the information list as in Eqn. 4 , where A is the number of anchors, $\mathcal{I}(R_i)$ is the i-th element in sorted information list.

$$\mathcal{I}(R_1) \geq \mathcal{I}(R_2) \geq \cdots \geq \mathcal{I}(R_A) \tag{4}$$

To reduce region redundancy, we adopt non-maximum suppression (NMS) on the regions based on their informativeness. Then we take the top-M informative regions $\{R_1, R_2, \ldots, R_M\}$ and feed them into the Teacher network to get the confidence as $\{\mathcal{C}(R_1), \mathcal{C}(R_2), \ldots \mathcal{C}(R_M)\}$. Fig. 3 shows the overview with $M = 3$, where M is a hyper-parameters denoting how many regions are used to train Navigator network. We optimize Navigator network to make $\{\mathcal{I}(R_1), \mathcal{I}(R_2), \ldots \mathcal{I}(R_M)\}$ and $\{\mathcal{C}(R_1), \mathcal{C}(R_2), \ldots \mathcal{C}(R_M)\}$ having the same order. Every proposed region is used to optimize Teacher by minimizing the cross-entropy loss between ground-truth class and the predicted confidence.

3.3 Scrutinizer

As Navigator network gradually converges, it will produce informative object-characteristic regions to help Scrutinizer network make decisions. We use the top-K informative regions combined with the full image as input to train the Scrutinizer network. In other words, those K regions are used to facilitate fine-grained recognition. Fig. 4 demonstrates this process with $K = 3$. Lam *et al.* [25]

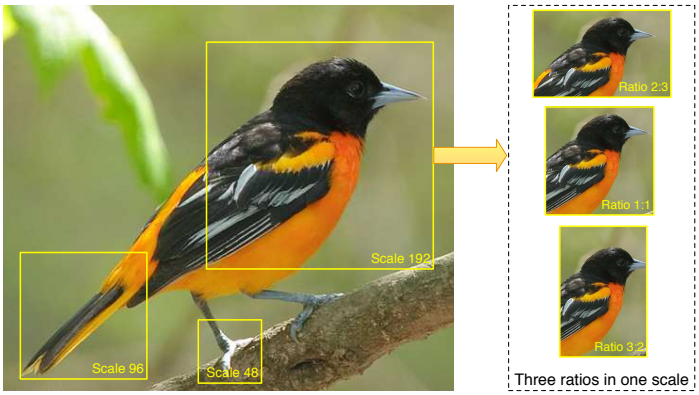

Fig. 2. The design of anchors. We use three scales and three ratios. For an image of size 448, we construct anchors to have scales of {48, 96, 192} and ratios {1:1, 2:3, 3:2}.

show that using informative regions can reduce intra-class variance and are likely to generate higher confidence scores on the correct label. Our comparative experiments show that adding informative regions substantially improve fine-grained classification results in a wide range of datasets including CUB-200-2001, FGVC Aircraft, and Stanford Cars, which are shown in Table. 2, 3.

3.4 Network architecture

In order to obtain correspondence between region proposals and feature vectors in feature map, we use fully-convolutional network as the feature extractor, without fully-connected layers. Specifically, we choose ResNet-50 [17] pre-trained on ILSVRC2012 [39] as the CNN feature extractor, and Navigator, Scrutinizer, Teacher network all share parameters in feature extractor. We denote parameters in feature extractor as $\mathbf{W}$. For input image X, the extracted deep representations are denoted as $X \otimes \mathbf{W}$, where $\otimes$ denotes the combinations of convolution, pooling, and activation operations.

Navigator network. Inspired by the design of Feature Pyramid Networks (FPN) [27], we use a top-down architecture with lateral connections to detect multi-scale regions. We use convolutional layers to compute feature hierarchy layer by layer, followed by ReLU activation and max-pooling. Then we get a series of feature maps of different spatial resolutions. The anchors in larger feature maps correspond to smaller regions. Navigator network in Figure. 4 shows the sketch of our design. Using multi-scale feature maps from different layers we can generate informativeness of regions among different scales and ratios. In our setting, we use feature maps of size $\{14 \times 14, 7 \times 7, 4 \times 4\}$ corresponding to regions of scale $\{48 \times 48, 96 \times 96, 192 \times 192\}$. We denote the parameters in Navigator network as $\mathbf{W}_{\mathcal{I}}$ (including shared parameters in feature extractor).

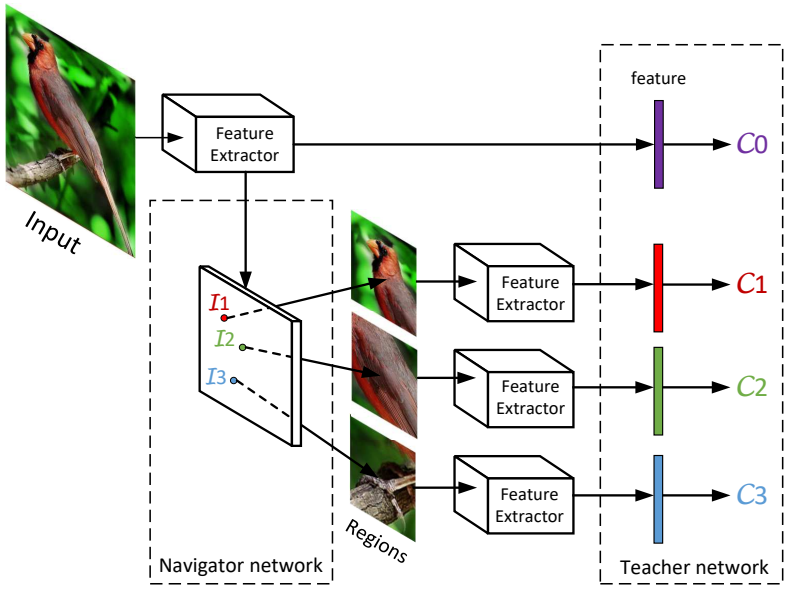

Fig. 3. Training method of Navigator network. For an input image, the feature extractor extracts its deep feature map, then the feature map is fed into Navigator network to compute the informativeness of all regions. We choose top-M (here $M = 3$ for explanation) informative regions after NMS and denote their informativeness as $\{I_1, I_2, I_3\}$. Then we crop the regions from the full image, resize them to the pre-defined size and feed them into Teacher network, then we get the confidences $\{C_1, C_2, C_3\}$. We optimize Navigator network to make $\{I_1, I_2, I_3\}$ and $\{C_1, C_2, C_3\}$ having the same order.

Teacher network. The Teacher network (Fig. 3) approximates the mapping $\mathcal{C} : \mathbb{A} \rightarrow [0, 1]$ which denotes the confidence of each region. After receiving M scale-normalized (224×224) informative regions $\{R_1, R_2, \ldots, R_M\}$ from Navigator network, Teacher network outputs confidence as teaching signals to help Navigator network learn. In addition to the shared layers in feature extractor, the Teaching network has a fully connected layer which has 2048 neurons. We denote the parameters in Teacher network as $\mathbf{W}_{\mathcal{C}}$ for convenience.

Scrutinizer network. After receiving top-K informative regions from Navigator network, the K regions are resized to the pre-defined size (in our experiments we use 224×224) and are fed into feature extractor to generate those K regions' feature vector, each with length 2048. Then we concatenate those K features with input image's feature, and feed it into a fully-connected layer which has $2048 \times (K + 1)$ neurons (Fig. 4). We use function $\mathcal{S}$ to represent the composition of these transformations. We denote the parameters in Scrutinizer network as $\mathbf{W}_{\mathcal{S}}$.

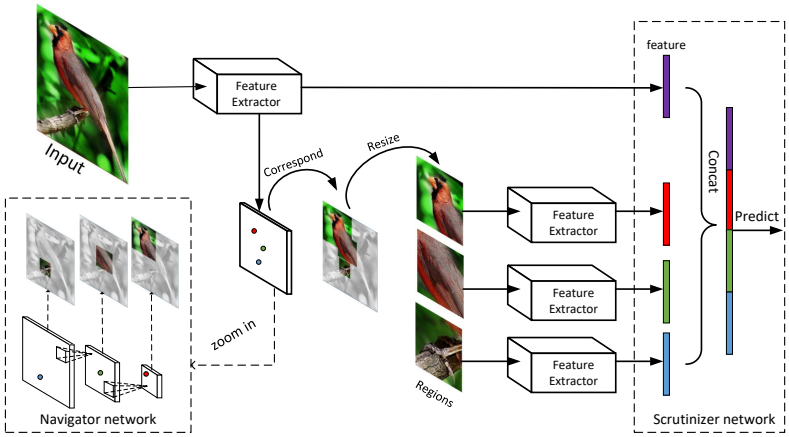

Fig. 4. Inference process of our model (here $K = 3$ for explanation). The input image is first fed into feature extractor, then the Navigator network proposes the most informative regions of the input. We crop these regions from the input image and resize them to the pre-defined size, then we use feature extractor to compute the features of these regions and fuse them with the feature of the input image. Finally, the Scrutinizer network processes the fused feature to predict labels.

3.5 Loss function and Optimization

Navigation loss. We denote the M most informative regions predicted by Navigator network as $R = \{R_1, R_2, \ldots, R_M\}$, their informativeness as $I = \{I_1, I_2, \ldots, I_M\}$, and their confidence predicted by Teacher network as $C = \{C_1, C_2, \ldots, C_M\}$. Then the navigation loss is defined as follow:

$$L_{\mathcal{I}}(I, C) = \sum_{(i,s):C_i<C_s} f(I_s - I_i) \tag{5}$$

where the function f is a non-increasing function that encourages $I_s > I_i$ if $C_s > C_i$, and we use hinge loss function $f(x) = \max\{1-x, 0\}$ in our experiment. The loss function penalize reversed pairs[5] between I and C, and encourage that I and C is in the same order. Navigation loss function is differentiable, and calculating the derivative *w.r.t.* $\mathbf{W}_{\mathcal{I}}$ by the chain rule in back-propagation we get:

$$\begin{aligned} &\frac{\partial L_{\mathcal{I}}(I, C)}{\partial \mathbf{W}_{\mathcal{I}}} \\ &= \sum_{(i,s):C_i<C_s} f'(I_s - I_i) \cdot \left(\frac{\partial \mathcal{I}(x)}{\partial \mathbf{W}_{\mathcal{I}}}\Big|_{x=R_s} - \frac{\partial \mathcal{I}(x)}{\partial \mathbf{W}_{\mathcal{I}}}\Big|_{x=R_i}\right) \end{aligned} \tag{6}$$

[5] Given a list $x = \{x_1, x_2, \cdots, x_n\}$ be the data and a permutation $\pi = \{\pi_1, \pi_2, \cdots, \pi_n\}$ be the order of the data. Reverse pairs are pairs of elements in x with reverse order. *i.e.* if $x_i < x_j$ and $\pi_i > \pi_j$ holds at same time, then x_i and x_j is an reverse pair.

The equation follows directly by the definition of $I_i = \mathcal{I}(R_i)$.

Teaching loss. We define the Teacher loss $L_{\mathcal{C}}$ as follows:

$$L_{\mathcal{C}} = -\sum_{i=1}^{M} \log \mathcal{C}(R_i) - \log \mathcal{C}(X) \tag{7}$$

where $\mathcal{C}$ is the confidence function which maps the region to its probability being ground-truth class. The first term in Eqn. 7 is the sum of cross entropy loss of all regions, the second term is the cross entropy loss of full image.[6]

Scrutinizing loss. When the Navigator network navigates to the most informative regions $\{R_1, R_2, \cdots, R_K\}$, the Scrutinizer network makes the fine-grained recognition result $P = \mathcal{S}(X, R_1, R_2, \cdots, R_K)$. We employ cross entropy loss as classification loss:

$$L_{\mathcal{S}} = -\log \mathcal{S}(X, R_1, R_2, \cdots, R_K) \tag{8}$$

Joint training algorithm. The total loss is defined as:

$$L_{total} = L_{\mathcal{I}} + \lambda \cdot L_{\mathcal{S}} + \mu \cdot L_{\mathcal{C}} \tag{9}$$

where λ and μ are hyper-parameters. In our setting, $\lambda = \mu = 1$. The overall algorithm is summarized in Algorithm. 1. We use stochastic gradient method to optimize L_{total}.

Algorithm 1: NTS-Net algorithm

Input: full image X, hyper-parameters K, M, λ, μ, assume $K \leq M$
Output: predict probability P

for t = *1,T* **do**
  Take full image = X
  Generate anchors $\{R'_1, R'_2, \ldots, R'_A\}$
  $\{I'_1, \ldots, I'_A\} := \mathcal{I}(\{R'_1, \ldots, R'_A\})$
  $\{I_i\}_{i=1}^{A}, \{R_i\}_{i=1}^{A} :=$ NMS$(\{I'_i\}_{i=1}^{A}, \{R'_i\}_{i=1}^{A})$
  Select top M: $\{I_i\}_{i=1}^{M}, \{R_i\}_{i=1}^{M}$
  $\{C_1, \ldots, C_K\} := \mathcal{C}(\{R_1, \ldots, R_K\})$
  $P = \mathcal{S}(X, R_1, R_2, \cdots, R_K)$
  Calculate L_{total} from Eqn. 9
  BP(L_{total}) get gradient *w.r.t.* $\mathbf{W}_{\mathcal{I}}$, $\mathbf{W}_{\mathcal{C}}$, $\mathbf{W}_{\mathcal{S}}$
  Update $\mathbf{W}_{\mathcal{I}}$, $\mathbf{W}_{\mathcal{C}}$, $\mathbf{W}_{\mathcal{S}}$ using SGD
end

[6] The second term helps training. For simplicity, we also denote the confidence function of full image as $\mathcal{C}$.

4 Experiments

4.1 Dataset

We comprehensively evaluate our algorithm on Caltech-UCSD Birds (CUB-200-2011) [42], Stanford Cars [23] and FGVC Aircraft [35] datasets, which are widely used benchmark for fine-grained image classification. We do not use any bounding box/part annotations in all our experiments. Statistics of all 3 datasets are shown in Table. 1, and we follow the same train/test splits as in the table.
Caltech-UCSD Birds. CUB-200-2011 is a bird classification task with 11,788 images from 200 wild bird species. The ratio of train data and test data is roughly $1:1$. It is generally considered one of the most competitive datasets since each species has only 30 images for training.
Stanford Cars. Stanford Cars dataset contains 16,185 images over 196 classes, and each class has a roughly 50-50 split. The cars in the images are taken from many angles, and the classes are typically at the level of production year and model (*e.g.* 2012 Tesla Model S).
FGVC Aircraft. FGVC Aircraft dataset contains 10,000 images over 100 classes, and the train/test set split ratio is around $2:1$. Most images in this dataset are airplanes. And the dataset is organized in a four-level hierarchy, from finer to coarser: Model, Variant, Family, Manufacturer.

Dataset	#Class	#Train	#Test
CUB-200-2011	200	5,994	5,794
Stanford Cars	196	8,144	8,041
FGVC Aircraft	100	6,667	3,333

Table 1. Statistics of benchmark datasets.

4.2 Implementation Details

In all our experiments, we preprocess images to size 448×448, and we fix $M = 6$ which means 6 regions are used to train Navigator network for each image (there is no restriction on hyper-parameters K and M). We use fully-convolutional network ResNet-50 [17] as feature extractor and use Batch Normalization as regularizer. We use Momentum SGD with initial learning rate 0.001 and multiplied by 0.1 after 60 epochs, and we use weight decay 1e−4. The NMS threshold is set to 0.25, no pre-trained detection model is used. Our model is robust to the selection of hyper-parameters. We use Pytorch to implement our algorithm and the code will be available at https://github.com/yangze0930/NTS-Net.

4.3 Quantitative Results

Overall, our proposed system outperforms all previous methods. Since we do not use any bounding box/part annotations, we do not compare with methods which

depend on those annotations. Table. 2 shows the comparison between our results and previous best results in CUB-200-2011. ResNet-50 is a strong baseline, which by itself achieves 84.5% accuracy, while our proposed NTS-Net outperforms it by a clear margin 3.0%. Compared to [26] which also use ResNet-50 as feature extractor, we achieve a 1.5% improvement. It is worth noting that when we use only full image ($K = 0$) as input to the Scrutinizer, we achieve 85.3% accuracy, which is also higher than ResNet-50. This phenomenon demonstrates that, in navigating to informative regions, Navigator network also facilitates Scrutinizer by sharing feature extractor, which learns better feature representation.

Method	top-1 accuracy
MG-CNN [43]	81.7%
Bilinear-CNN [28]	84.1%
ST-CNN [19]	84.1%
FCAN [32]	84.3%
ResNet-50 (implemented in [26])	84.5%
PDFR [47]	84.5%
RA-CNN [12]	85.3%
HIHCA [5]	85.3%
Boost-CNN [36]	85.6%
DT-RAM [26]	86.0%
MA-CNN [49]	86.5%
Our NTS-Net (K = 2)	87.3%
Our NTS-Net (K = 4)	**87.5**%

Table 2. Experimental results in CUB-200-2011.

Table. 3 shows our result in FGVC Aircraft and Stanford Cars, respectively. Our model achieves new state-of-the-art results with 91.4% top-1 accuracy in FGVC Aircraft and 93.9% top-1 accuracy in Stanford Cars.

4.4 Ablation Study

In order to analyze the influence of different components in our framework, we design different runs in CUB-200-2011 and report the results in Table. 4. We use NS-Net to denote the model without Teacher's guidance, NS-Net let the Navigator network alone to propose regions and the accuracy drops from 87.5% to 83.3%, we hypothesize it is because the navigator receives no supervision from teacher and will propose random regions, which we believe cannot benefit classification. We also study the role of hyper-parameter K, *i.e.* how many part regions have been used for classification. Referring to Table. 4, accuracy only increases 0.2% when K increases from 2 to 4, the accuracy improvement is minor while feature dimensionality nearly doubles. On the other hand, accuracy

Method	top-1 on FGVC Aircraft	top-1 on Stanford Cars
FV-CNN [15]	81.5%	-
FCAN [32]	-	89.1%
Bilinear-CNN [28]	84.1%	91.3%
RA-CNN [12]	88.2%	92.5%
HIHCA [5]	88.3%	91.7%
Boost-CNN [36]	88.5%	92.1%
MA-CNN [49]	89.9%	92.8%
DT-RAM [26]	-	93.1%
Our NTS-Net (K = 2)	90.8%	93.7%
Our NTS-Net (K = 4)	**91.4**%	**93.9**%

Table 3. Experimental results in FGVC Aircraft and Stanford Cars.

increases 2.0% when K increases from 0 to 2, which demonstrate simply increasing feature dimensionality will only get minor improvement, but our multi-agent framework will achieve considerable improvements (0.2% vs 2%).

Method	top-1 accuracy
ResNet-50 baseline	84.5%
NS-Net (K = 4)	83.3%
Our NTS-Net (K = 0)	85.3%
Our NTS-Net (K = 2)	87.3%
Our NTS-Net (K = 4)	**87.5**%

Table 4. Study of influence factor in CUB-200-2011.

4.5 Qualitative Results

To analyze where Navigator network navigates the model, we draw the navigation regions predicted by Navigator network in Fig. 5. We use red, orange, yellow, green rectangles to denote the top four informative regions proposed by Navigator network, with red rectangle denoting most informative one. It can be seen that the localized regions are indeed informative for fine-grained classification. The first row shows $K = 2$ in CUB-200-2011 dataset: we can find that using two regions are able to cover informative parts of birds, especially in the second picture where the color of the bird and the background is quite similar. The second row shows $K = 4$ in CUB-200-2011: we can see that the most informative regions of birds are head, wings and main body, which is consistent with the human perception. The third row shows $K = 4$ in Stanford Cars: we can find that the headlamps and grilles are considered the most informative regions

of cars. The fourth row shows $K = 4$ in FGVC Airplane: the Navigator network locates the airplane wings and head, which are very helpful for classification.

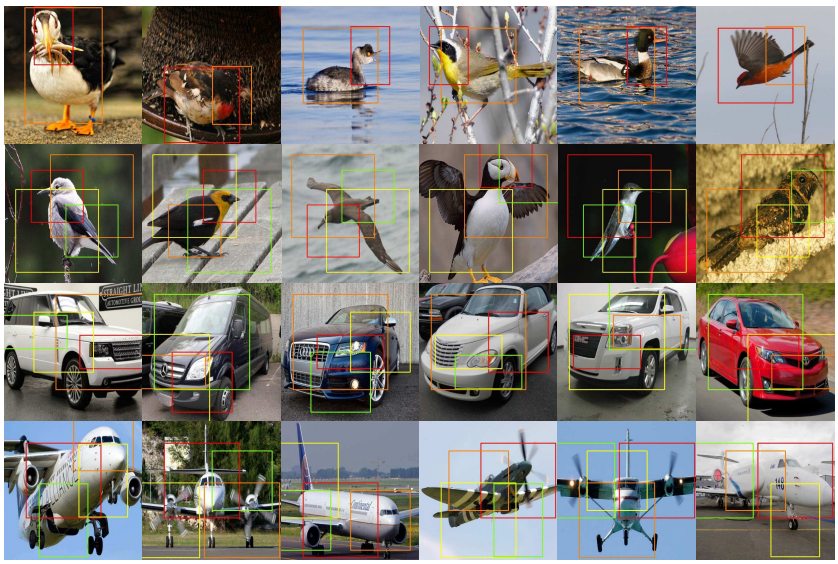

Fig. 5. The most informative regions proposed by Navigator network. The first row shows $K = 2$ in CUB-200-2011 dataset. The second to fourth rows show $K = 4$ in CUB-200-2011, Stanford Cars and FGVC Aircraft, respectively.

5 Conclusions

In this paper, we propose a novel method for fine-grained classification without the need of bounding box/part annotations. The three networks, Navigator, Teacher and Scrutinizer cooperate and reinforce each other. We design a novel loss function considering the ordering consistency between regions' informativeness and probability being ground-truth class. Our algorithm is end-to-end trainable and achieves state-of-the-art results in CUB-200-2001, FGVC Aircraft and Stanford Cars datasets.

6 Acknowledgments

This work is supported by National Basic Research Program of China (973 Program) (grant no. 2015CB352502), NSFC (61573026) and BJNSF (L172037).

References

1. Arbelaez, P., Ponttuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR. pp. 328–335 (2014)
2. Berg, T., Belhumeur, P.N.: Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In: CVPR (2013)
3. Branson, S., Horn, G.V., Belongie, S., Perona, P.: Bird species categorization using pose normalized deep convolutional nets. In: BMVC (2014)
4. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: ICML. pp. 89–96 (2005)
5. Cai, S., Zuo, W., Zhang, L.: Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In: ICCV (Oct 2017)
6. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank:from pairwise approach to listwise approach. In: ICML. pp. 129–136 (2007)
7. Carreira, J., Sminchisescu, C.: CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts. IEEE Computer Society (2012)
8. Chai, Y., Lempitsky, V., Zisserman, A.: Symbiotic segmentation and part localization for fine-grained categorization. In: ICCV. pp. 321–328 (2013)
9. Cossock, D., Zhang, T.: Statistical analysis of bayes optimal subset ranking. IEEE Transactions on Information Theory **54**(11), 5140–5154 (2008)
10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. pp. 886–893 (2005)
11. Endres, I., Hoiem, D.: Category independent object proposals. In: ECCV. pp. 575–588 (2010)
12. Fu, J., Zheng, H., Mei, T.: Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In: CVPR
13. Gavves, E., Fernando, B., Snoek, C.G.M., Smeulders, A.W.M., Tuytelaars, T.: Fine-grained categorization by alignments. In: ICCV. pp. 1713–1720 (2014)
14. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. pp. 580–587 (2014)
15. Gosselin, P.H., Murray, N., Jgou, H., Perronnin, F.: Revisiting the fisher vector for fine-grained classification. Pattern Recognition Letters **49**, 92–98 (2014)
16. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. TPAMI **37**(9), 1904–16 (2015)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
18. Herbrich, R.: Large margin rank boundaries for ordinal regression. Advances in Large Margin Classifiers **88** (2000)
19. Jaderberg, M., Simonyan, K., Zisserman, A., kavukcuoglu, k.: Spatial transformer networks. In: NIPS, pp. 2017–2025 (2015)
20. Jie, Z., Liang, X., Feng, J., Jin, X., Lu, W., Yan, S.: Tree-structured reinforcement learning for sequential object localization. In: NIPS, pp. 127–135 (2016)
21. Konda, V.R.: Actor-critic algorithms. Siam Journal on Control and Optimization **42**(4), 1143–1166 (2002)
22. Krause, J., Jin, H., Yang, J., Fei-Fei, L.: Fine-grained recognition without part annotations. In: CVPR (June 2015)
23. Krause, J., Stark, M., Jia, D., Li, F.F.: 3d object representations for fine-grained categorization. In: ICCV Workshops. pp. 554–561 (2013)
24. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)

25. Lam, M., Mahasseni, B., Todorovic, S.: Fine-grained recognition as hsnet search for informative image parts. In: CVPR (July 2017)
26. Li, Z., Yang, Y., Liu, X., Zhou, F., Wen, S., Xu, W.: Dynamic computational time for visual attention. In: ICCV (Oct 2017)
27. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (July 2017)
28. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: ICCV (2015)
29. Liu, J., Kanazawa, A., Jacobs, D., Belhumeur, P.: Dog breed classification using part localization. In: ECCV. pp. 172–185 (2012)
30. Liu, T.Y.: Learning to rank for information retrieval. Found. Trends Inf. Retr. **3**(3), 225–331 (Mar 2009)
31. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. pp. 21–37 (2016)
32. Liu, X., Xia, T., Wang, J., Lin, Y.: Fully convolutional attention localization networks: Efficient attention localization for fine-grained recognition. CoRR (2016)
33. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. CVPR (Nov 2015)
34. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
35. Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Tech. rep. (2013)
36. Moghimi, M., Belongie, S., Saberian, M., Yang, J., Vasconcelos, N., Li, L.J.: Boosted convolutional neural networks. In: BMVC. pp. 24.1–24.13 (2016)
37. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
38. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
39. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV **115**(3), 211–252 (2015)
40. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., Lecun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. Arxiv (2013)
41. Uijlings, J.R., Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV **104**(2), 154–171 (2013)
42. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. rep. (2011)
43. Wang, D., Shen, Z., Shao, J., Zhang, W., Xue, X., Zhang, Z.: Multiple granularity descriptors for fine-grained categorization. In: ICCV. pp. 2399–2406 (2015)
44. Xia, F., Liu, T.Y., Wang, J., Li, H., Li, H.: Listwise approach to learning to rank: theory and algorithm. In: ICML. pp. 1192–1199 (2008)
45. Xie, L., Tian, Q., Hong, R., Yan, S.: Hierarchical part matching for fine-grained visual categorization. In: ICCV. pp. 1641–1648 (2013)
46. Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based rcnn for fine-grained detection. In: ECCV (2014)
47. Zhang, X., Xiong, H., Zhou, W., Lin, W., Tian, Q.: Picking deep filter responses for fine-grained image recognition. In: CVPR (June 2016)
48. Zhao, B., Wu, X., Feng, J., Peng, Q., Yan, S.: Diversified visual attention networks for fine-grained object classification. Trans. Multi. **19**(6), 1245–1256 (Jun 2017)
49. Zheng, H., Fu, J., Mei, T., Luo, J.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: ICCV (Oct 2017)