Learning to Navigate for Fine-grained Classification

Jing Liu

2018.9.29

Contributions:

- **Propose a novel multi-agent cooperative learning scheme.**
- **Design a novel loss function.**
- **Model can be trained end-to-end.**

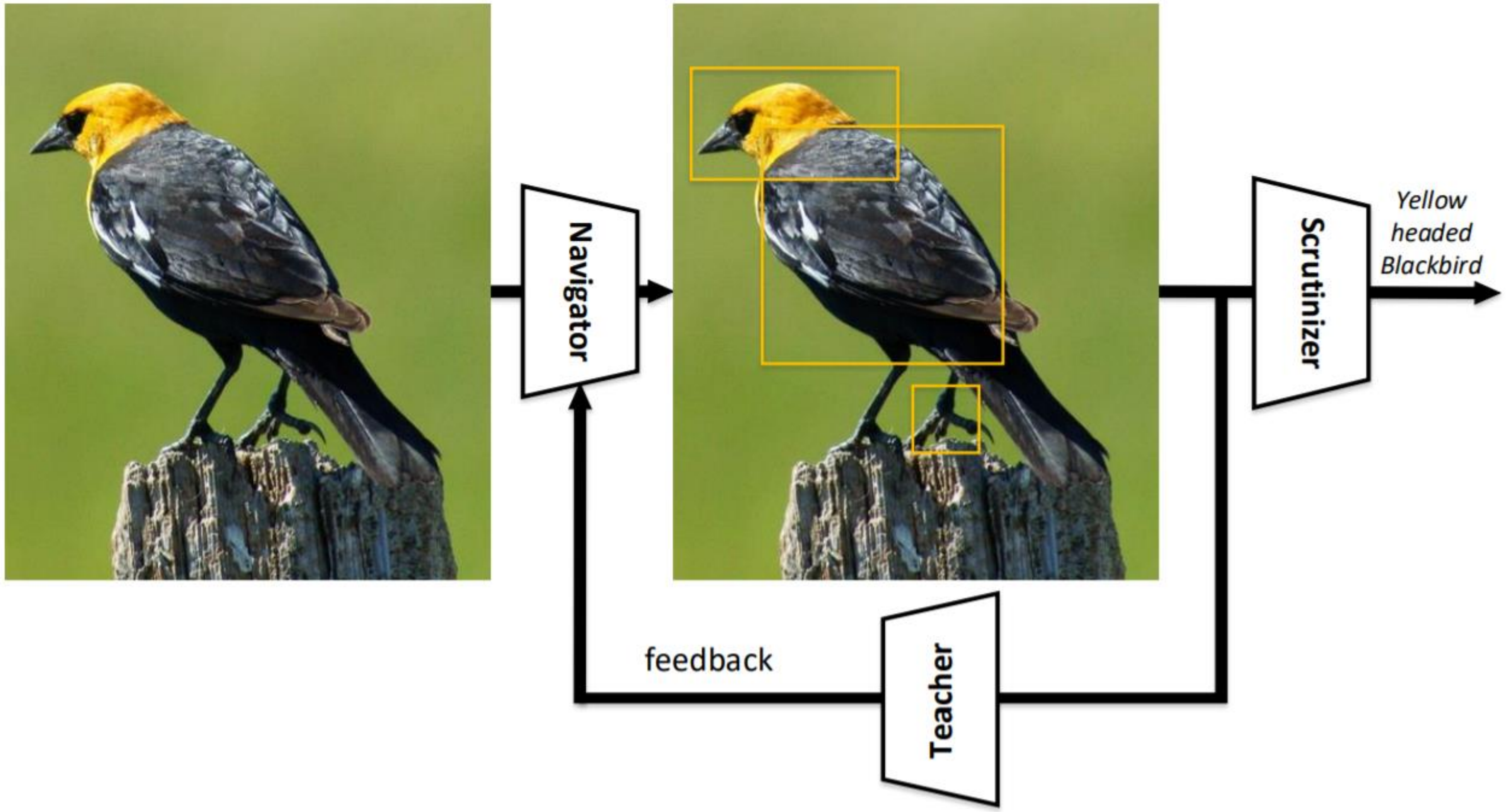


Fig. 1. The overview of the model. The Navigator navigates the model to focus on the most informative regions (denoted by yellow rectangles), while Teacher evaluates the regions proposed by Navigator and provides feedback. After that, the Scrutinizer scrutinizes those regions to make predictions.

Approach Overview:

- Condition. 1: for any $\mathrm{R}_1, \mathrm{R}_2 \in \mathrm{A}$, if $\mathrm{C}(\mathrm{R}_1) > \mathrm{C}(\mathrm{R}_2)$, $\mathrm{I}(\mathrm{R}_1) > \mathrm{I}(\mathrm{R}_2)$

- $\mathrm{I}(\mathrm{R}_1) \geqslant \mathrm{I}(\mathrm{R}_2) \geqslant \cdots \geqslant \mathrm{I}(\mathrm{R}_\mathrm{A})$

Network architecture:Navigator network

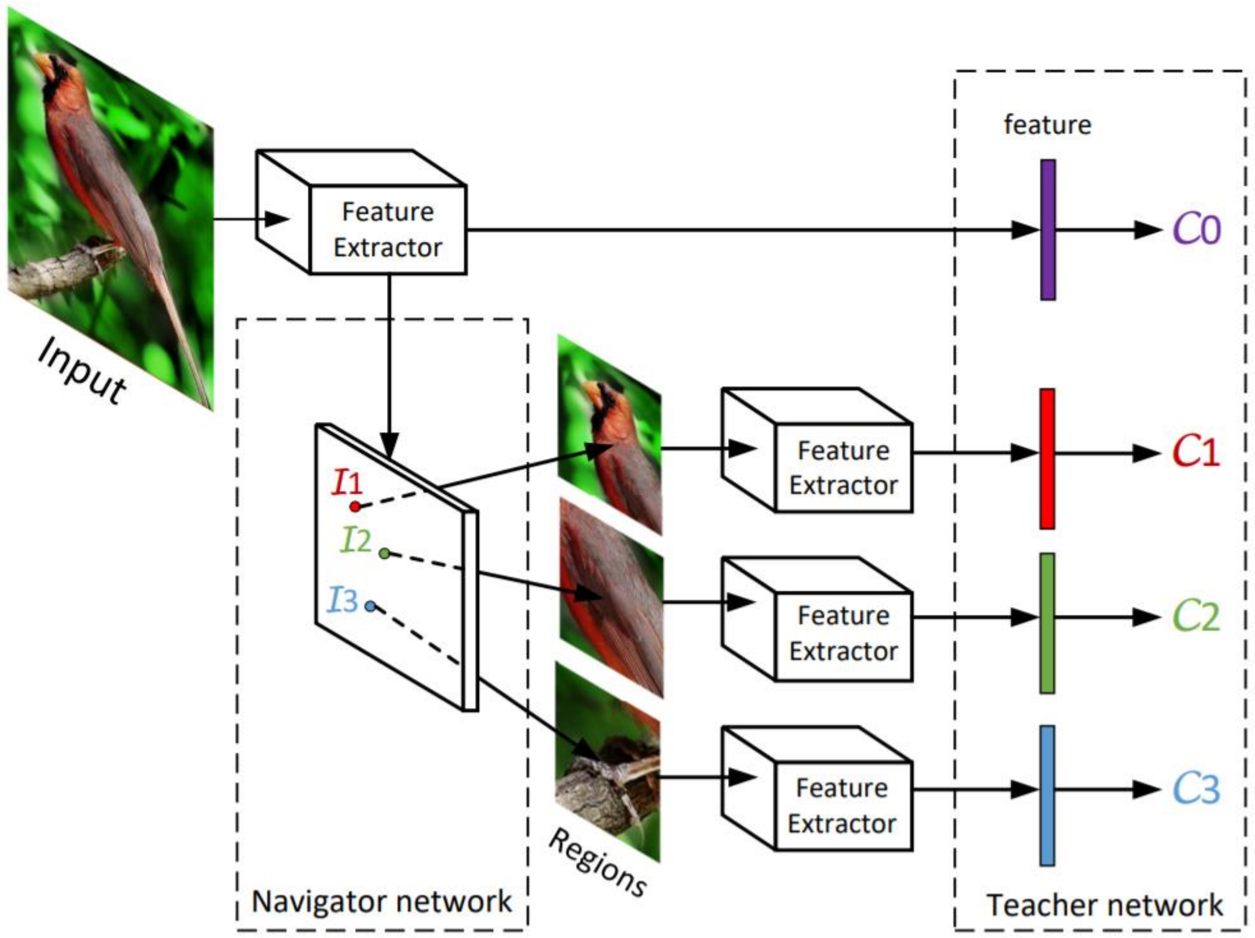


Fig. 2. Training method of Navigator network. For an input image, the feature extractor extracts its deep feature map, then the feature map is fed into Navigator network to compute the informativeness of all regions. We choose top-M (here M = 3 for explanation) informative regions after NMS and denote 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.

Yang, Ze, et al. "Learning to Navigate for Fine-grained Classification." in ECCV, 2018.

Network architecture:Scrutinizer network

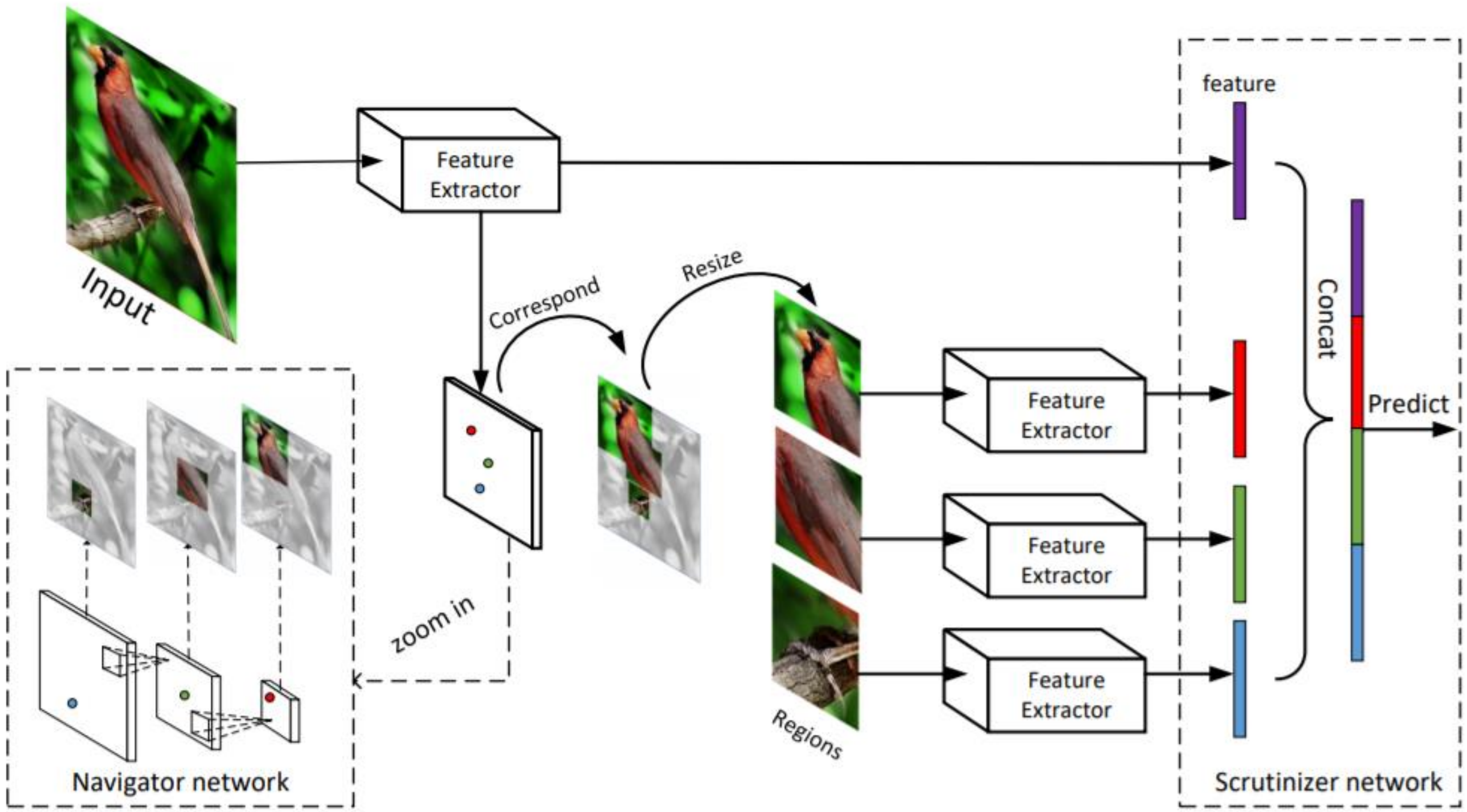


Fig. 3. Inference process of our model (here K = 3 for explanation). The input image is first fed into feature extractor, then the Navigator network proposes the most informative regions of the input. We crop these regions from the input image and resize them to the pre-defined size, then we use feature 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.

Yang, Ze, et al. "Learning to Navigate for Fine-grained Classification." in ECCV, 2018.

Loss function and Optimization:

- Navigation loss:

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

- Teaching loss:

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

Loss function and Optimization:

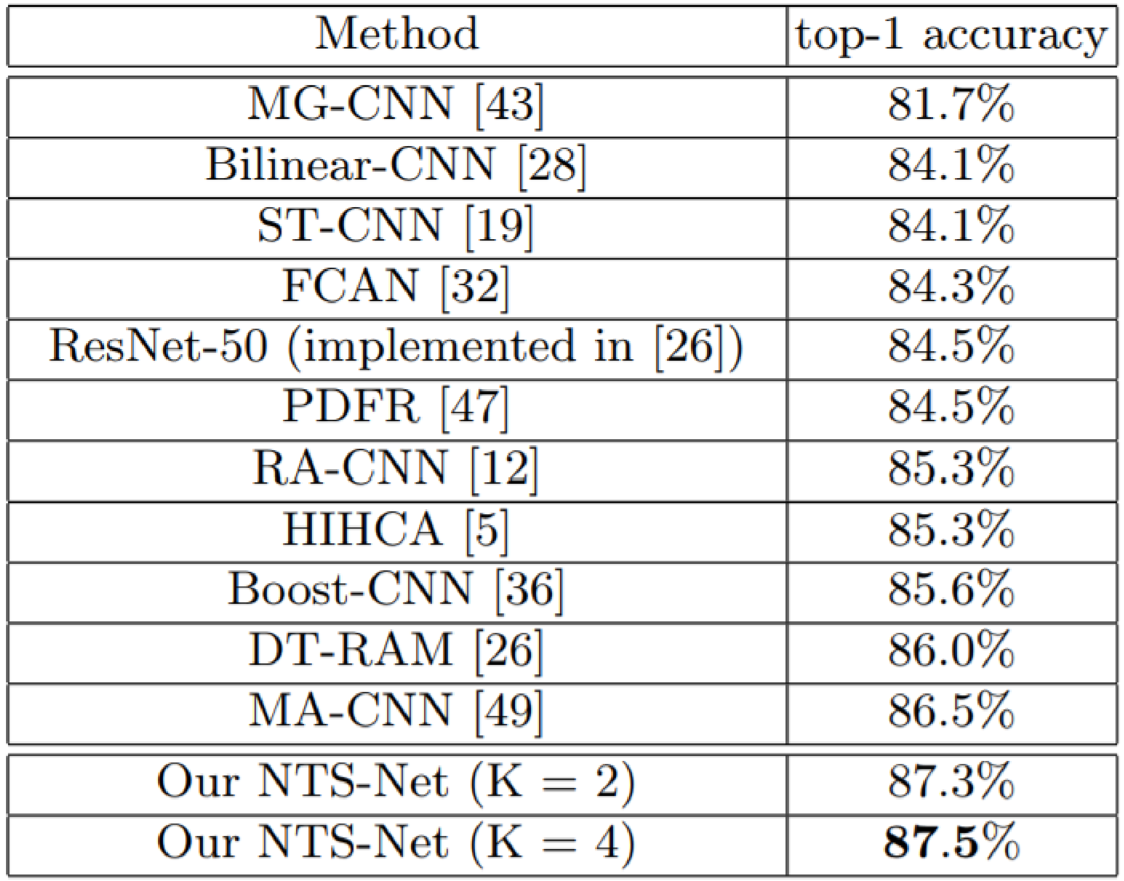
- Scrutinizing loss:

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

- Joint training algorithm:

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

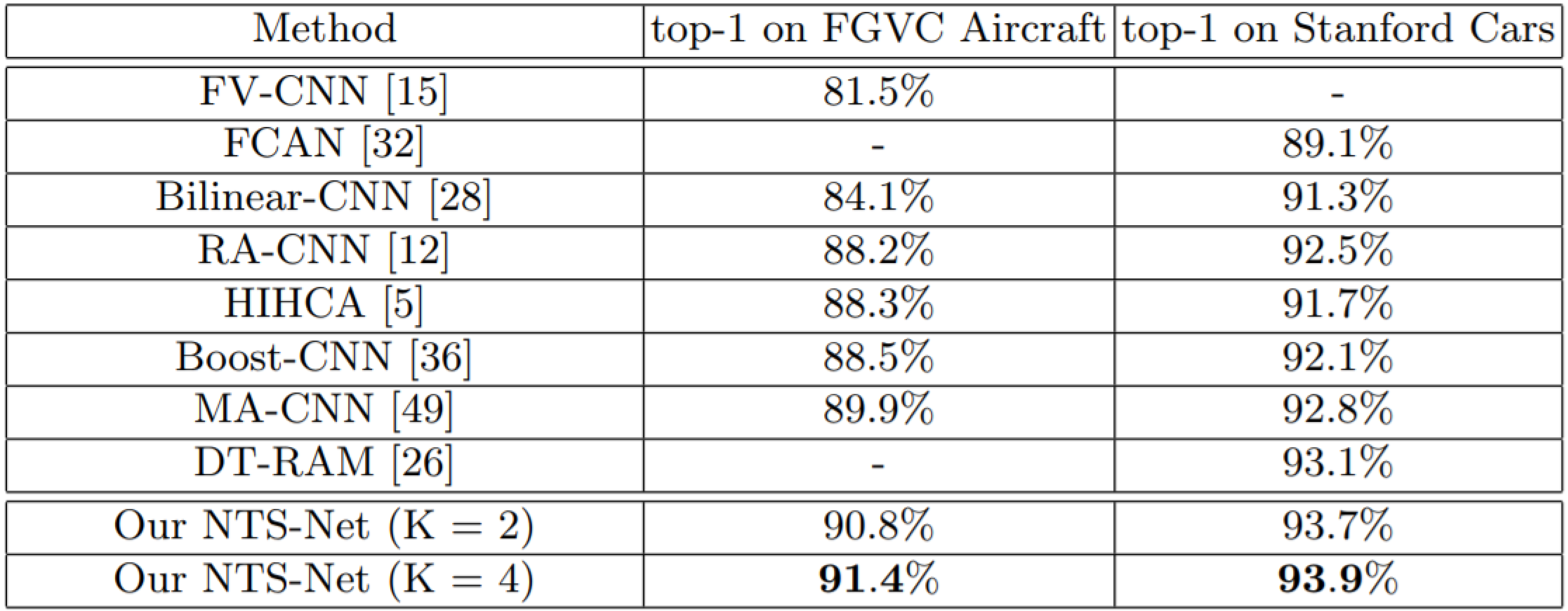
Quantitative Results:



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 1. Experimental results in CUB-200-2011.

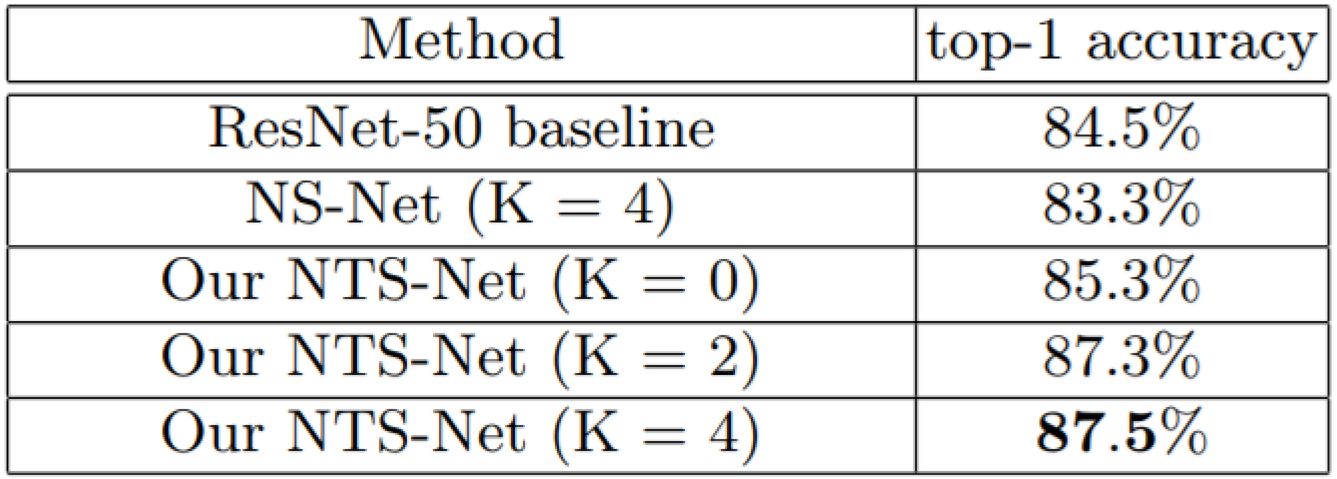
Quantitative Results:



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 2. Experimental results in FGVC Aircraft and Stanford Cars.

Quantitative Results:



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 3. Study of influence factor in CUB-200-2011.

Qualitative Results:

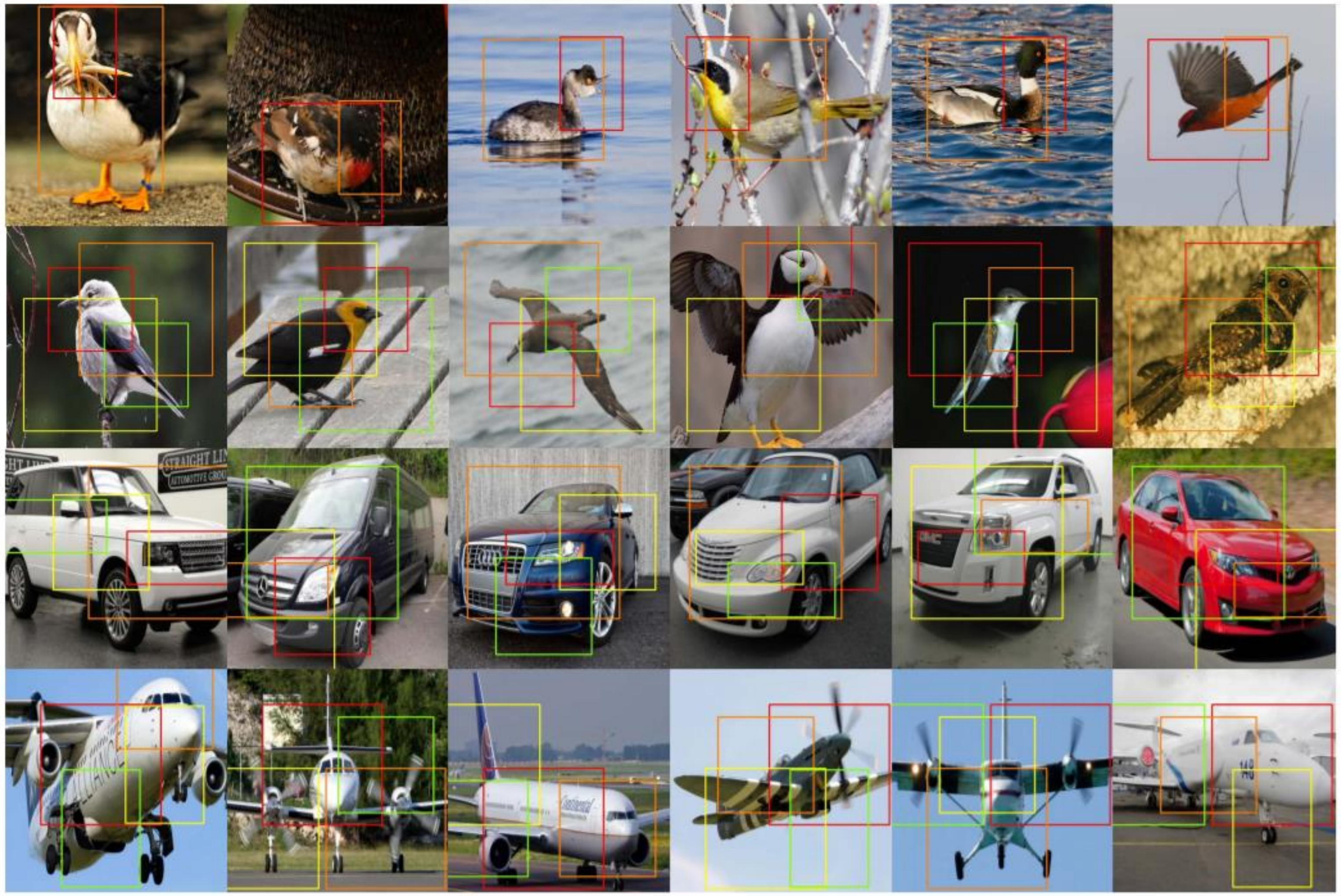


Fig. 4. The most informative regions proposed by Navigator network. The first row shows K = 2 in CUB-200-2011 dataset. The second to fourth rows show K = 4 in CUB-200-2011, Stanford Cars and FGVC Aircraft, respectively.