Learning to Navigate for Fine-grained Classification

Ze Yang, Peking University

Problem

- Fine-grained classification aims at differentiating categories that are very similar. For instance, the subordinate classes of a common superior class. The subordinate classes are similar in appearance.

Lazuli Bunting

Indigo Bunting

Examples

- Determine plant species, breed of dogs, identification of dishes.

Examples

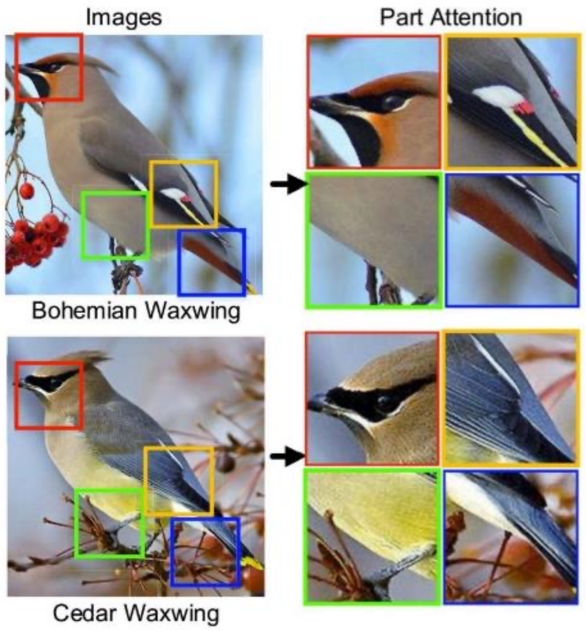
- Clothing recognition and retrieval

Examples

- Product recognition, smart retail

Key points to fine-grained classification

- Categories are different, but share a common part structure.
- The key point to fine-grained classification lies in accurately identifying informative regions in the image.



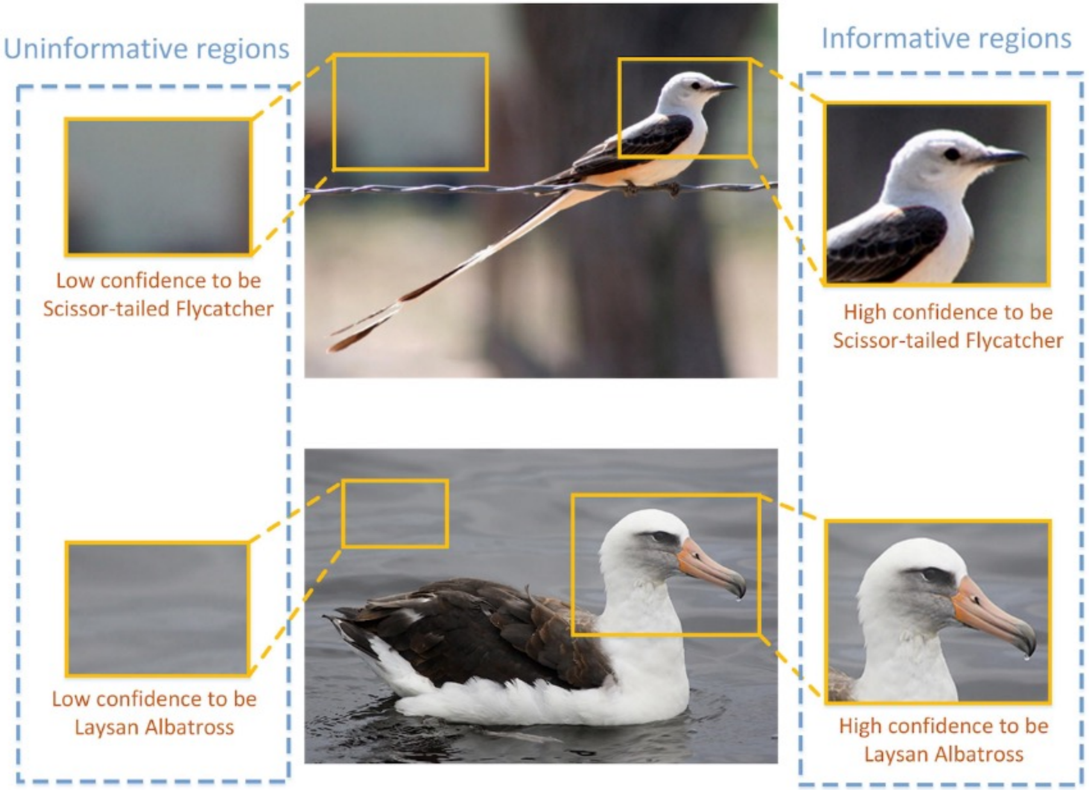
Our works

Learning to Navigate for fine-grained classification

ECCV 2018

Motivations

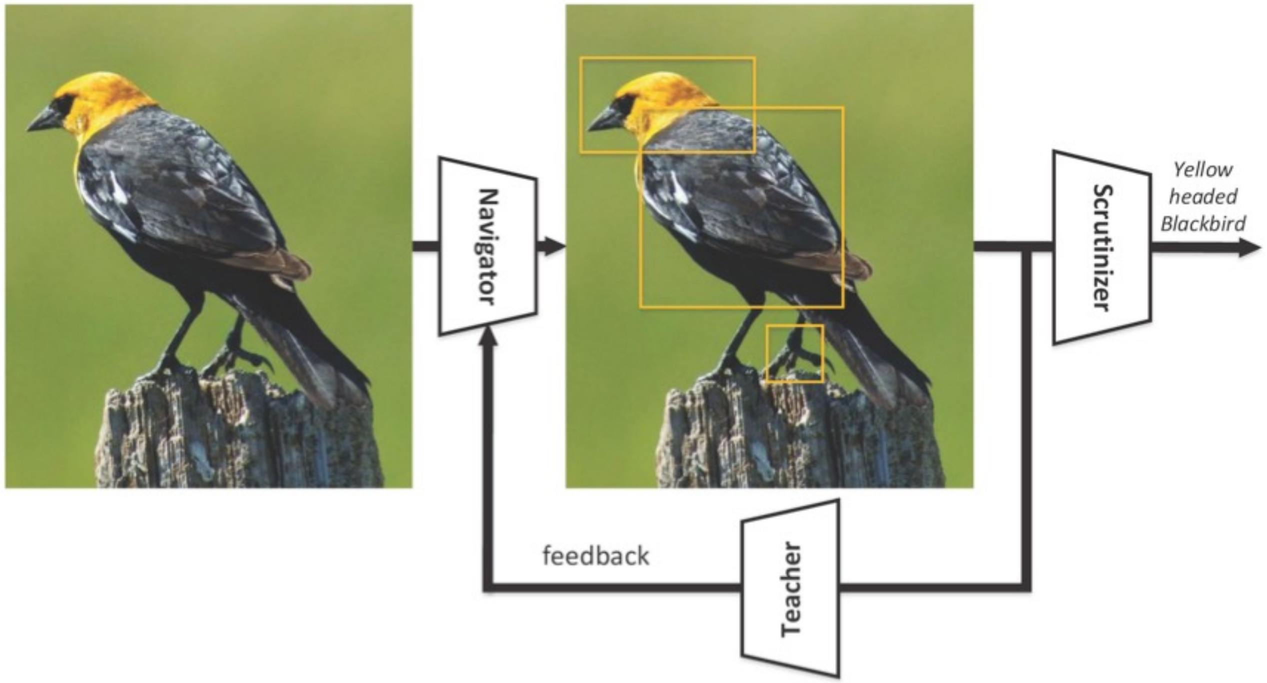
- Intrinsic consistency between informativeness of the regions and their probability being ground-truth class



For informative regions, they will be assigned high probability being ground-truth class. But for uninformative regions that cannot help to differentiate classes, the classifier will not know their class and assigns them low probability being ground-truth class.

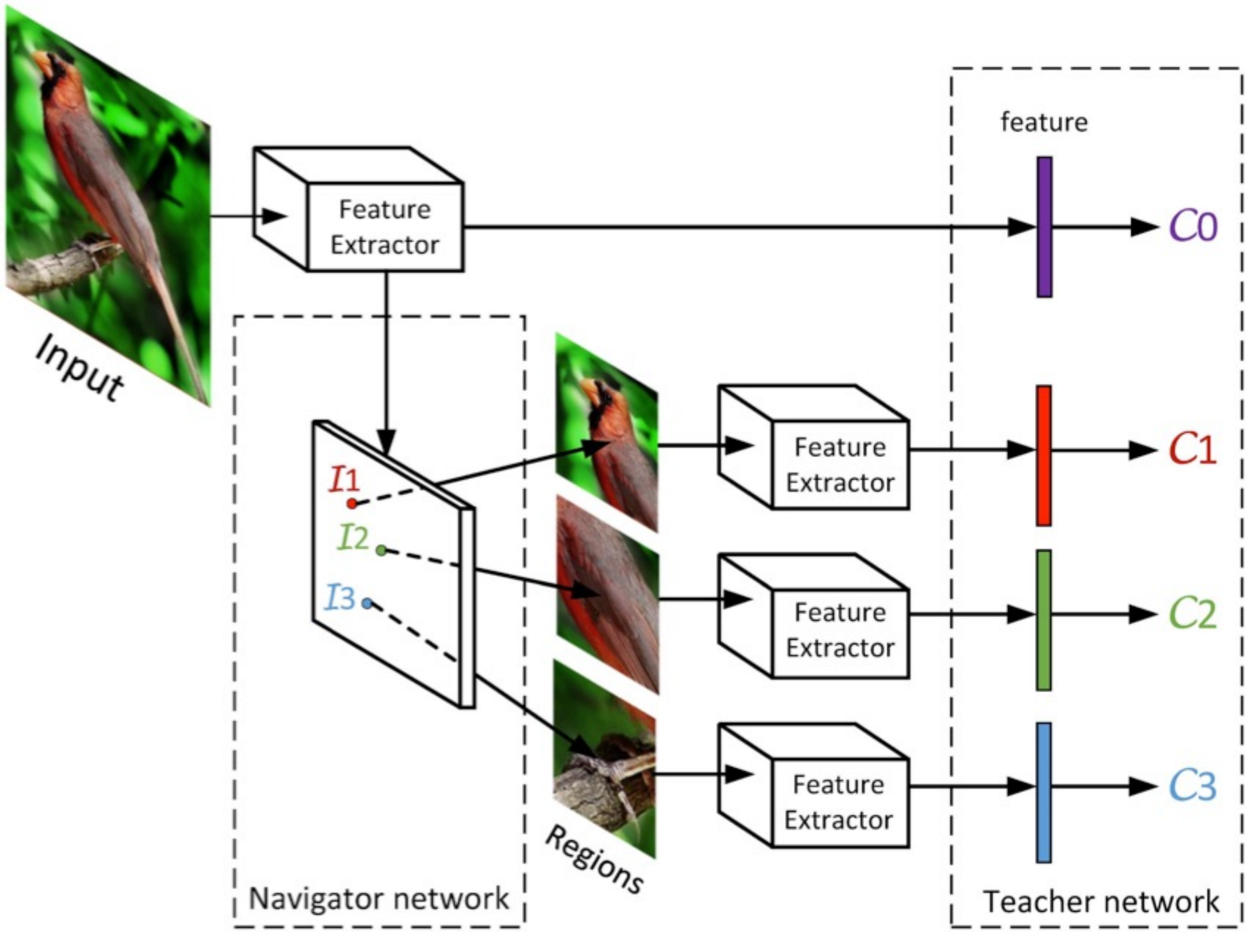
Overview

- Navigator: navigates the model to focus on informative regions.
- Teacher: evaluates the regions and provides feedback.
- Scrutinizer: scrutinizes those regions to make predictions.



Methodology

- Train the Navigator to propose informative regions.



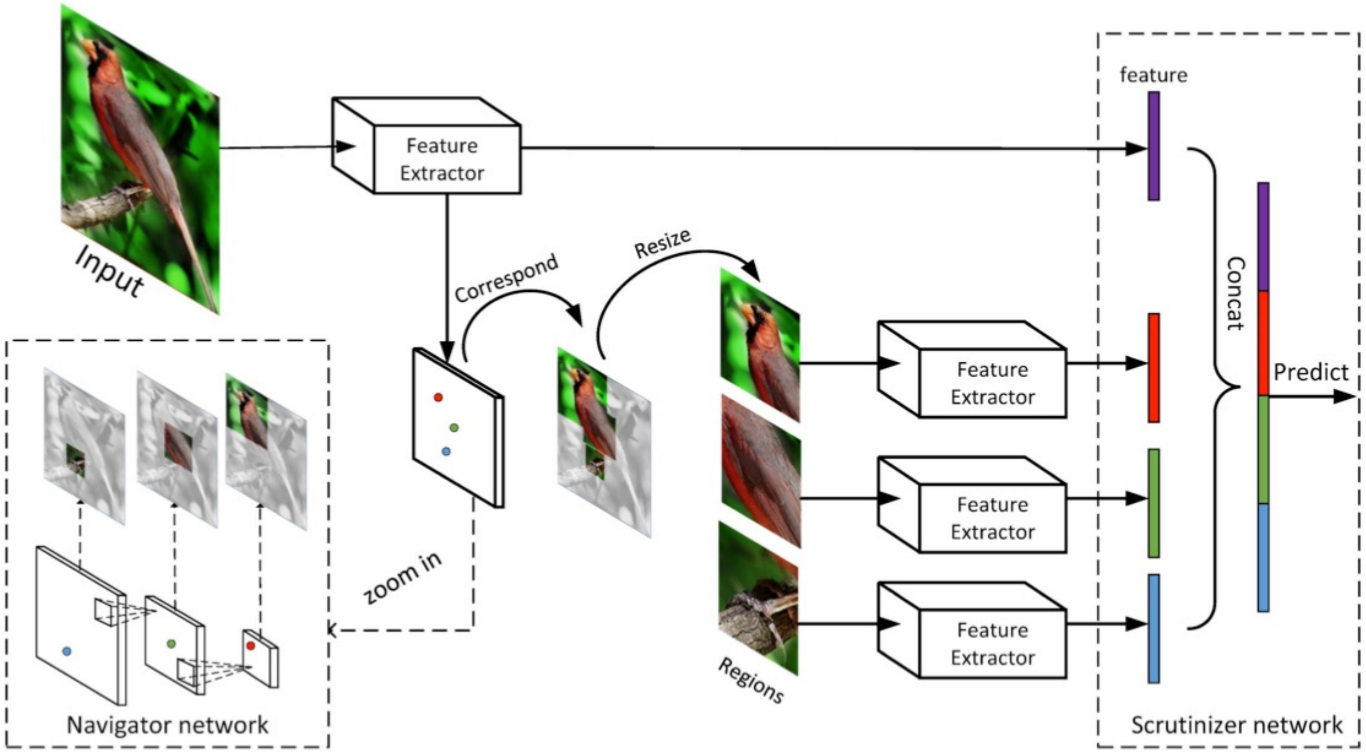
Navigator network is a RPN to compute the informativeness of all regions. We choose top-M (here M=3) informative regions with informativeness {I1, I2, I3}. Then the Teacher network compute their confidences being GT class {C1, C2, C3}. We use ranking loss to optimize Navigator network to make {I1, I2, I3} and {C1, C2, C3} having the same order (function f is non-decreasing).

Ranking loss: $\sum_{(i,s):C_i<C_s} f(I_s - I_i)$

where the function f is a non-increasing function that encourages $I_s > I_i$ if $C_s > C_i$

Methodology

- The Scrutinizer makes predictions.



Navigator network proposes the top-K (here K=3) informative regions. Then the Scrutinizer network uses these regions and full image to make predictions.

We use cross entropy loss to optimize the Teacher and the Scrutinizer.

Methodology

- Algorithm overview.

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, ..., R'_A}
    {I'_1, ..., I'_A} := I({R'_1, ..., 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, ..., C_K} := C({R_1, ..., R_K})
    P = S(X, R_1, R_2, ..., R_K)
    Calculate L_total = L_I + λ · L_S + μ · L_C
    BP(L_total) get gradient w.r.t. W_I, W_C, W_S
    Update W_I, W_C, W_S using SGD
end
```

$\textbf{for } t = 1, T \textbf{ do}$

- Take full image $= X$
- Generate anchors $\{R'_1, R'_2, \ldots, R'_A\}$
- $\{I'_1, \ldots, I'_A\} := \mathcal{I}(\{R'_1, \ldots, R'_A\})$
- $\{I_i\}_{i=1}^{A}, \{R_i\}_{i=1}^{A} := \mathrm{NMS}(\{I'_i\}_{i=1}^{A}, \{R'_i\}_{i=1}^{A})$
- 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 $L_{total} = L_{\mathcal{I}} + \lambda \cdot L_{\mathcal{S}} + \mu \cdot L_{\mathcal{C}}$
- $\mathrm{BP}(L_{total})$ 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

$\textbf{end}$

Experiments

- 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%**

Experimental results in CUB-200-2011. The table shows the comparison between our results and previous best results in CUB-200-2011. We use M=6 casually, which means top-6 informative regions are used to train the Navigator. We also study the role of hyper-parameter K, *i.e.* how many part regions have been used for fine-grained classification.

Experiments

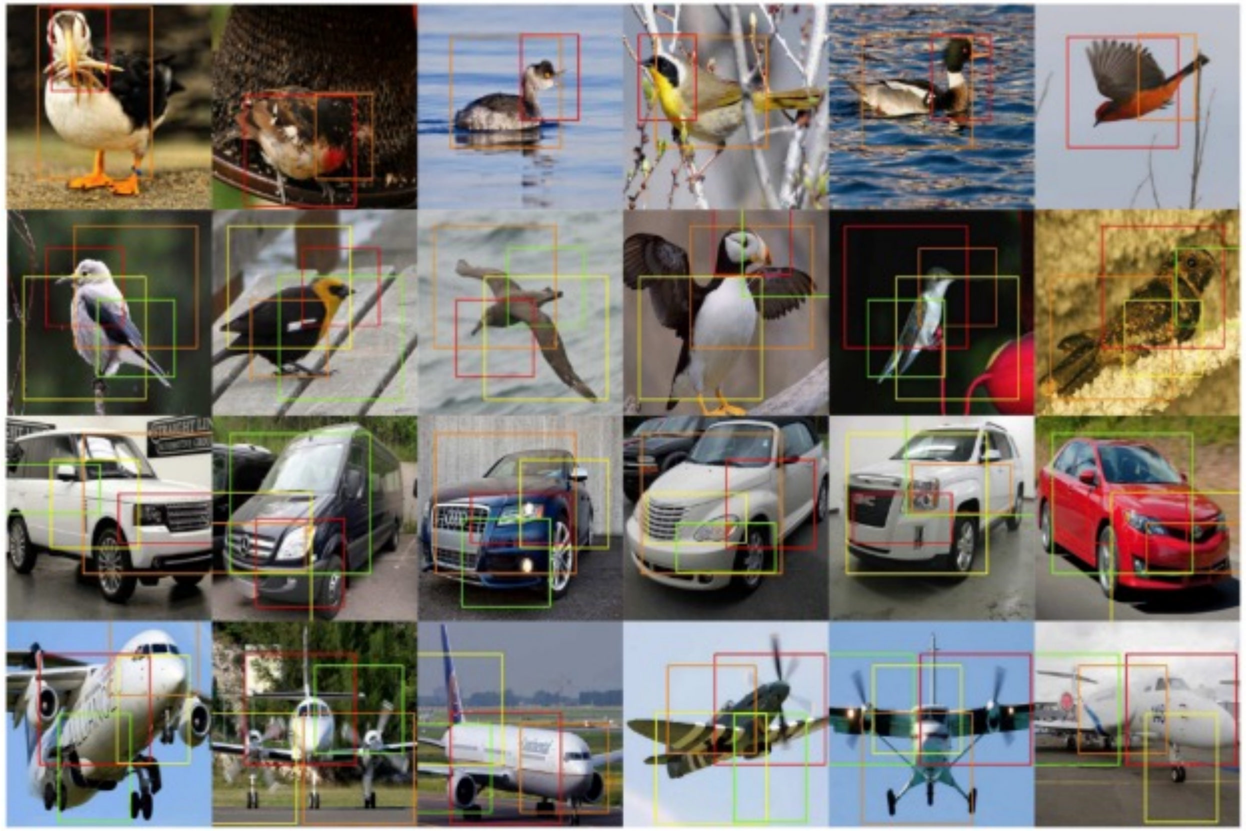
- Quantitative results.

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**%

Experimental results in FGVA Aircraft and Stanford Cars datasets.

Experiments

- Qualitative results.



The most informative regions proposed by Navigator network. We can see that the most informative regions are consistent with the human perception

- Birds: head, wings and main body
- Cars: headlamps and grilles
- Airplanes: wings and heads

Especially in the blue box picture where the color of the bird and the background is quite similar.

Thank you