NYCU Introduction to Machine Learning, Final Project

110705013, 沈昱宏

Part. 1, Environment Details (5%):

Training:

Python 3.10.12

Hardware

cuda version 12.3

GPU - NVIDIA GeForce GTX 1080 Ti

(inference on google colab)

Framework

Torch 2.1.1

Torchvision 0.16.1

Part. 2, Implementation Details (15%)

Model Architecture

The original paper: Fine-grained Visual Classification with High-temperature Refinement and Background Suppression.

Implementation supported by chou141253/FGVC-HERBS.

The paper introduces a neural network architecture called "high-temperature refinement and background suppression" (HERBS). It aims to solve fine-grained visual classification. There are two challenges in this task: (1) subtle differences between categories and (2) the significance of background information. HERBS comprises two modules: the high-temperature refinement module and the background suppression module. The refinement module refines feature maps at different scales, enabling the model to learn diverse and appropriate features (solving first challenge). Meanwhile, the suppression module segregates foreground and background features based on classification confidence scores and suppresses low-confidence areas, enhancing discriminative features while reducing background noise (solving second challenge).

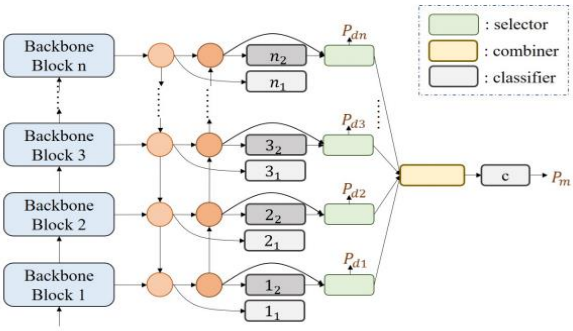
Model structure:

The selectors filter out unimportant areas (deal with background noise)

The classifier is for refining (make important pixels more distinguishable)

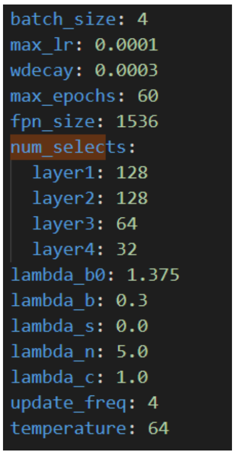
Backbone: Swin-transformer

Orange circle: FPN



Hyperparameters

Details as shown below. I did not tune most parameters, the hyperparameters of the original implementation is good enough. I only tune num_select, which is the number of features selected by each layer's selector.



```
batch_size: 4
max_lr: 0.0001
wdecay: 0.0003
max_epochs: 60
fpn_size: 1536
num_selects:
  layer1: 128
  layer2: 128
  layer3: 64
  layer4: 32
lambda_b0: 1.375
lambda_b: 0.3
lambda_s: 0.0
lambda_n: 5.0
lambda_c: 1.0
update_freq: 4
temperature: 64
```

Training strategy

1. Modified batch size and num_selects to put the model into GPU (I only got 11.8G GPU memory)
2. Tuned the num_selects because the original model tends to overfit. It has high accuracy on training set but has only about 90% accuracy on validation set. I made the model smaller by tuning num_selects and the result on validation set turns out to be about 92%
3. Split validation set for model evaluation with shell script.
4. I used the swin-transformer pretrained model and did not use the model pretrained on CUB-200-2011 because I think it is a little like cheating if I use it or train on the entire CUB-200-2011 dataset.

Part. 3, Experimental results

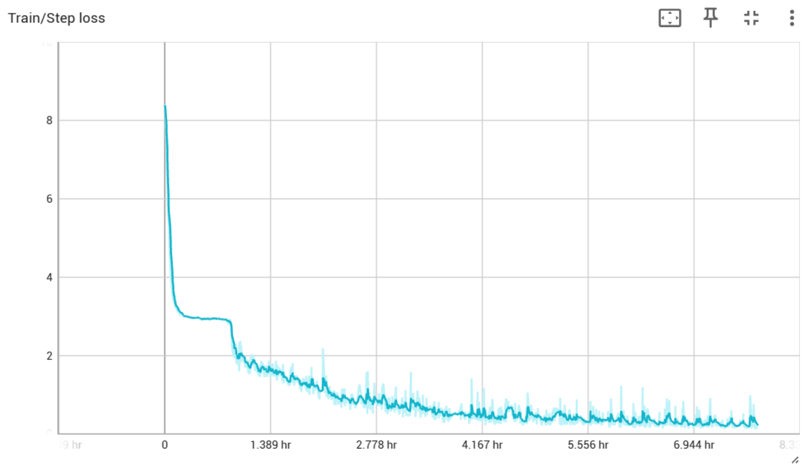
Evaluation metrics

I trained 3 models in total (the original one / the one mentioned in training strategy part / the one used for ablation study). The first and the last one used the same training set, and the second model used a different validation/training set to make the model learn differently. I tried several ways to combine these three models.

1. Voting -> accuracy 0.914 (in kaggle competition)
2. Confidence -> given an image, each model will output a score for each class, I recorded the name of the class with the highest score with it's score minus the second highest score. This approach has accuracy at 0.917. Then, I find out that the second model has the best accuracy, so I gave it a little bonus (0.15 points), this point is tunable, and the result is 0.918.

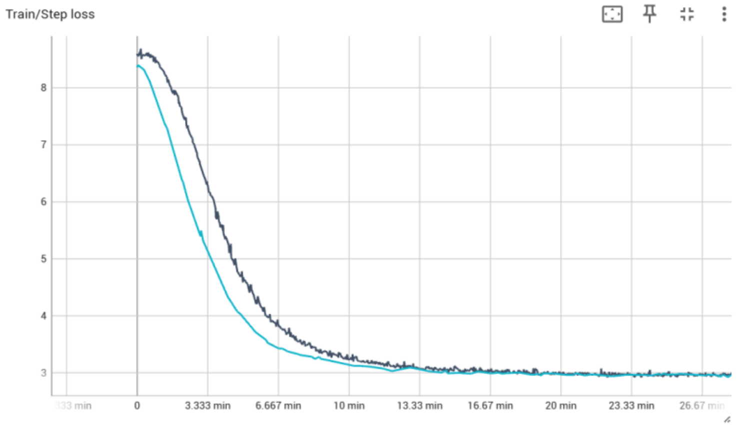
Training curve

The model can be trained end-to-end, and I recorded the total loss with tensorboard. (I don't know why there is a weird flat part on the curve, but it is just there…)



Ablation study

The original paper has done several ablation studies. These ablation studies are also done on CUB-200-2011 dataset, and I don't think I will achieve a different a different result, so I decided to do the ablation study on the amplifier of the model. I kept everything the same except for the amplifier. The curve with amplifier (beginning part of the above figure) converges faster than the one without amplifier. The original curve dropped after the 1hr point but I did not train the one without amplifier for that much time.



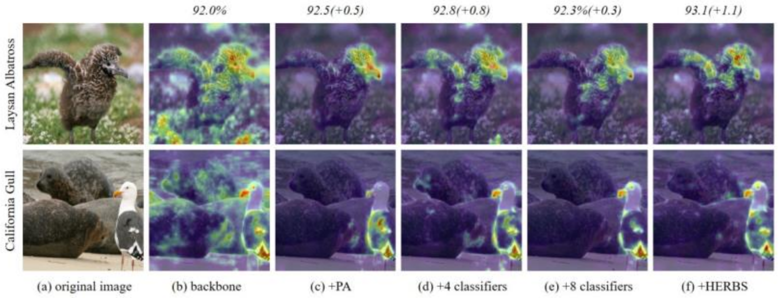


Fig. 5. Visualization of heat maps generated from different model. (a) original color image, (b) Swin Transformer backbone, (c) backbone + PA, (d) backbone + PA with four classifier, (e) backbone + PA with eight classifiers. (f) backbone + HERBS. The number on the top of the images represents the accuracy of the corresponding model.

Ablation studies shown in paper.

Part. 4, Paper Review (5%)

HERBS employs two innovative approaches, high-temperature refinement and background suppression, to address key challenges in fine-grained classification. The effect of these model is clearly shown in the graph. The way they show the effect is nice and clear (in the graph). In the heat map, you can observe that HERB have most light points on important areas (the effect of background suppresssion) and the darkest point (red points) are placed on the most important feature (high-temperature refinement), showing that their approach did achieve the effect they want.