Fine grained bird species classification

1. Pre-processing

The provided data set consists of 1100 images split between 1000 for training, 100 for testing and around 500 images for submission. We observed that most the submission images where from further distances compared to train and test data sets.

1.1. Image Cropping

We used a Fast-R-CNN [3]architecture to crop the training and validation images. The cropped images where then added to the original data set. The submission images all being taken from afar, training our model only on cropped (and thus zoomed) images would hurt it's accuracy.

1.2. Data Augmentation

The data set being quite small, data augmentation techniques are needed to add variety to our data and thus make our model robust against these changed. The operations we used included (Flips, Color Jitter, Random shear, Random Grayscale).

1.3. Unlabeled data

We parsed the needed classes from the INat dataset, these images where fed during pseudo labeling. Please refer to section 3.1.

2. Architecture

For our model we chose a for Navigator,Teacher,Scrutinizer Network (NTS-NET) from [1]. Which achieves state of the art performance in fine grained classification tasks. The Navigator agent guides the model to focus on most informative regions in the image. Navigator predicts how informative is each region of the image. Teacher agent evaluates the most informative regions proposed by the Navigator and provides feedback. Finally,the Scrutinizer extracts features from the proposed regions, the features of all regions are then concatenated and passed down to a dense layer to make the final classification.

In our work, Teacher and Navigator are trained from scratch, the Scrutinizer consists of a Resnet50 pretrained on ImageNet [2]. We used the pytorch implementation from [7]

3. Training

3.1. Pseudo Labeling

Larger models require more data in order to generalize well. The curb the issue of lack of data we use **Pseudo Labeling** [4] which is a semi-supervised learning technique. It consists of taking a set of unlabeled data points, using the pre-trained model to make predictions on these points and use the predicted targets as ground truth targets for the new data. For this technique to work, our model must have relative generalization properties before training with pseudo labels.

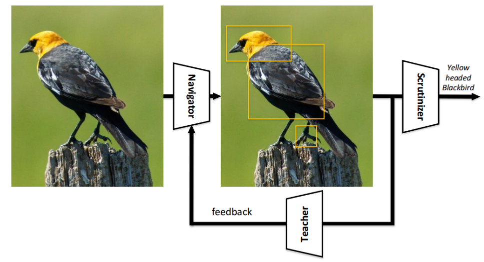

Figure 1. NTS Net Architecture [**?**]

3.2. Hyper parameters

We experimented with various configurations for our optimizer, ranging from the recent **NovoGrad** [5] algorithm for layer wise adaptive moments to more empirical **Ranger-Lars** [6]. To our surprise we obtained the best results using SGD with momentum. Adding Look Ahead [8] wrapper to the optimizer gave more consistent convergence and better results overall.

The architecture being relatively large we couldn't not experiment much with the **batch size** as we settled for the maximum possible of 16.

Other parameters where hand tuned including the number of region proposals (K) and the number of accepted regions (M).

3.3. Procedure

Model was trained on the provided data set with early stopping on the validation accuracy. After this first phase, we alternate between epochs of semi-supervised training, and fully supervised training.

We report on our results W/O pseudo labelling in the table below :

	Validation error	Submission error
Supervised	84	74.8
+ Semi-supervised	90	78.7

References

[1] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, Liwei Wang. [*Learning to Navigate for Fine-grained Classification*] arXiv:1809.00287. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. [*Deep Residual Learning for Image Recognition*] arXiv:1512.03385. 1

[3] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. [*Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*] arXiv:1506.01497. 1

[4] Dong-Hyun Lee. [*Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks*] ICML 2013 Workshop : Challenges in Representation Learning (WREPL), Atlanta, Georgia, USA, 2013 1

[5] Boris Ginsburg, Patrice Castonguay, Oleksii Hrinchuk, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, Huyen Nguyen, Yang Zhang, Jonathan M. Cohen. [*Stochastic Gradient Methods with Layer-wise Adaptive Moments for Training of Deep Networks*] arXiv:1905.11286. 1

[6] https://github.com/mgrankin/over9000

[7] https://github.com/yangze0930/NTS-Net 1

[8] Michael R. Zhang, James Lucas, Geoffrey Hinton, Jimmy Ba [*Lookahead Optimizer: k steps forward, 1 step back*] arXiv:1907.08610. 1

1