Assignment #3

Semi-supervised Learning

Due on Nov 1, 23:59 (11:59 pm)

Overview

- In the assignment, you will implement a multi-class image classifier to recognize flowers.
- In the semi-supervised learning, you will be given both labeled and unlabeled images.
- In this assignment, you will employ the self-training technique to enhance the classifier using both labeled and unlabeled data.

Flower Dataset

- The dataset is collected by Alexander Mamaev.
- It contains 4,317 images in 5 classes, with about 800 images per class
- The train/val/test splits are provided.
- In training set, we will give you 854 images with labels, and 1735 images without labels.
- The val set contains 864 images and test set contains 864 images.


daisy


dandelion


rose


sunflower


tulip

- Your model will be evaluated on the test set using the accuracy metric Your model will be evaluated on the test set using the accuracy metric

Self-training

Step1: Train the base model with labeled training set. (supervised training)

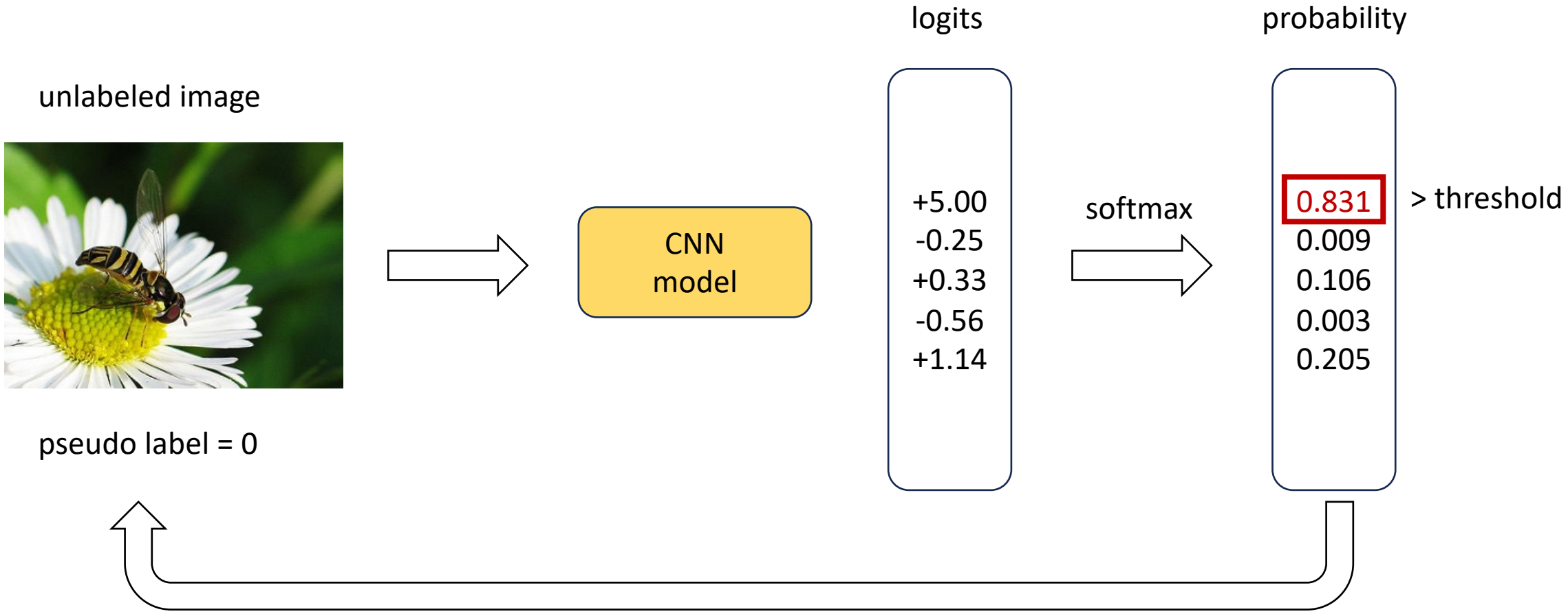
Step2: Iterate the following steps:

1. Use the trained model to predict the unlabeled data.
2. If the confidence scores of unlabeled data's prediction exceeds the threshold, select them as pseudo-labeled data.
3. Combine the labeled training data with the pseudo-labeled data to create a new training set.
4. Retrain the model on the new training set.

Supervised training

- Design a convolutional neural network to recognize the flowers. You must train your model from scratch (e.g., randomly initialized weights).
- The images provided are of different resolutions. You'll need to resize the images into a fixed size of your own choice and try different data augmentation.
- To get a high accuracy, you'll need to experiment with different filter sizes, different number of layers, or other tricks like dropout, batch normalization to boost performance.
- The goal of this assignment is for you to test different convolutional structures. You cannot directly use the blocks/architectures of pre-trained models.

How to generate pseudo label .



You'll need to write the "get_pseudo_label" function. First assign a threshold value, if the probability exceeds the threshold, assign pseudo label to the unlabeled image. You can try different threshold values to generate your pseudo label.

Things you cannot do

- You cannot submit results predicted by others.
- You cannot copy trained models from others.
- You cannot copy code from others, internet, GitHub ...
- You cannot collect more images to train your model in order to boost performance.
- You cannot use the pre-constructed model.
- You cannot use the weights of pre-trained model.

Any violation will result in 0 scores!

Submission

- Submit your predictions on the test images to Kaggle for evaluation.

- Kaggle Competition: https://www.kaggle.com/t/e304bb12c8a84e5c9c1b27a6c3bd4026
  - Use your SID as your team's name

- Submit your code along with the report to the CU.

Grading

- 80% competition (final accuracy = 0.5 * public accuracy + 0.5 * private accuracy)
  - Bonus points to top 3 teams
  - Top 3 teams will share their approaches in class
- 20% report