Assignment #3-2 (Bonus)

Flower classification with **self-training**

Due on Nov 26, 11:59 pm

Overview

- In assignment #3, you implemented a multi-class image classifier to recognize flowers.

- You will design and train a deep convolutional network **based on Assignment #3** with **extra data** and **the self-training method** to predict the class label of a flower image.

- Please note that you're still not allowed to use a pre-trained model.

Self-training

- Given: labelled training data & unlabeled training data

- Train the model with labelled data.
- Repeat:
  - Predict the unlabeled data with the model to get **pseudo labels.**
  - Remove the data with high confidence level from the unlabeled dataset and add them to the labelled dataset.
  - Finetune with labeled dataset.
- End: Repeat until all the unlabeled data with the pseudo labels reach certain confidence level, or until there're no unlabeled data left.

Flower Dataset

- The dataset is the same as Assignment #3
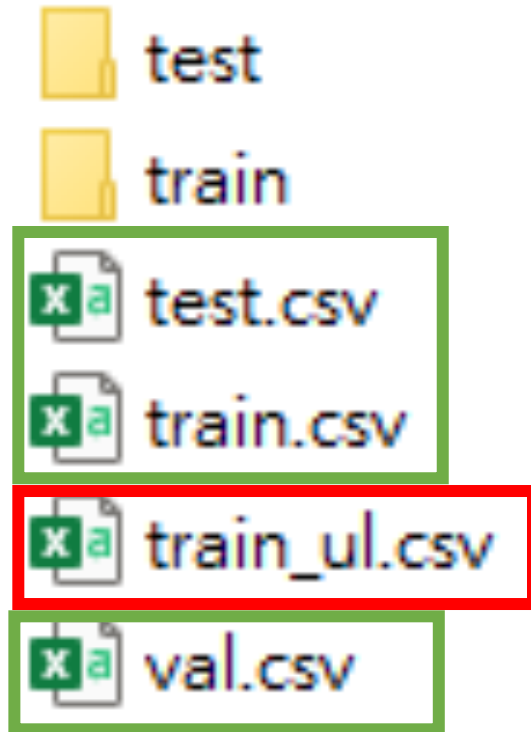

| daisy | dandelion | rose | sunflower | tulip |

- The train(**labelled and unlabeled**)/val/test splits are provided.
- Your model will be evaluated on the test set using the accuracy metric.

Assignment #3-2 Dataset

test

train

test.csv

train.csv   **The same as assignment #3**

train_ul.csv   **The new unlabeled training data for self-training**

val.csv

Your task

- We have code skeleton for you guys.
- https://colab.research.google.com/drive/13QW99mhNFIroKCPoDZwOZ6L56pwuaQIz

- Design a convolutional neural network to recognize the flowers. You must train your model based on your assignment #3.
- The images provided are of different resolutions. You'll need to resize the images into a fixed size of your own choice.
- To get a high accuracy, you'll need to experiment with different filter sizes, different number of layers, and other design principles discussed in class to figure out a network architecture that works best.
- You'll also need to try data augmentation, dropout, batch normalization as well as different optimizers and other tricks to boost performance.
- Again, you cannot use any pre-trained model in this part.

Things you cannot do

- You cannot submit results from Assignment #3.
- 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 weights of pre-trained model.

Any violation will result in no bonus points!

Submission

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

 Kaggle Competition:

https://www.kaggle.com/t/d288f1f130294567b37614e649f9536d

Submit your code + report to the CU.

- The report must include:
  - Your network architecture What you have experimented for performance improvements. This could include experiments that work and those that do not work.
  - Other findings if any
- Grading on report will mostly depend on the report quality. Please show your effort.

Grading

- 10 points competition
- 20 points report