NTU AMMAI 2021 Spring
HW #1: Face Verification
Due date: 12pm, April 8, 2021

Note:
1. If you have any questions, please send an email to TA (Jia-Fong Yeh) <jiafongyeh@ieee.org> or post the question on ceiba forum.
2. You can download the dataset here.
2. Submit a zip file named with your id to Ceiba before the deadline.
3. You are encouraged to discuss the assignment with classmates, but DO NOT copy your implementation from others!
4. You are encouraged to write the homework in English, and please DO comment on your source code appropriately.

Problem:
In this assignment, you need to solve the face verification problem using deep learning skills.
You need to train the model on the Asian Politician Dataset (APD) and test its performance under two different settings.

Closed-set setting:
In this setting, we will sample face image pairs from the APD dataset (same as the training dataset, your model has seen these persons (but not these images)), your model needs to verify that they are from the same person (output 1) or not (output 0).
Open-set setting:
In this setting, we will sample face image pairs from LFW dataset, which means your model hasn't seen these persons during training, your model still needs to verify whether they are the same person.

(optional) We will provide a set of unlabeled images sampled from LFW dataset, you can use this set of images to fine-tune your model before the inference. If you choose this option, please also provide your accuracy with and without fine-tuning in the field of open-set setting.

Baseline Model
Any face verification model can be your baseline model, for example, CosFace, ArcFace, SphereFace. Please re-implement the model or find third-party implementation (make sure you provide the source) in Pytorch.

There are no limitations of the training procedure, you can train your model using a standard way, or meta-learning etc. Please note that **you must train your model on APD only**.
And your model will only receive an image pair each time in the testing phase. It needs to output the verification result using these two images.

Submission Format

You need to package all your codes and report in to a zip file and name it with your student ID (eg. d08922010.zip).
When decompressing your zip file, the architecture of the folder should be like:

```
d08922010/
--------- report.pdf
--------- codes/
------------------- main.py
------------------- OTHER_CODES_FILE1.py
                    :
------------------- OTHER_CODES_FILEN.py
------------------- data/
------------------- trained_model/
------------------- OTHER_SUB_FOLDERS/
```

data/ is our provided dataset, please make sure that you name and place it correctly.
trained_model/ includes your best trained model parameter file, if the size is too large, please upload to the online drive and provide the link in your report.
And you also need to have a main.py that I can execute with the following command:

```
python main.py --mode phase
```

The *phase* could be train/closed/open.
You can have other control references for changing model versions, etc.

Requirements

1. Re-implement or find third party implementation of baseline model.
2. Modify the model with your novelty.
3. Write a report (pdf file) that includes:
    a. The following table.

Model	Closed-set Accuracy	Open-set Accuracy
baseline		
Variant 1		
:	:	:
Variant N		

    b. Other experimental results.

c. Any discussion / finding you want to discuss.

Grading

(60%) Re-implement or modify third party implementation of baseline models. Train it on the APD dataset only, and it can be executed correctly during testing.

(20%) Modify the model with your novelty and provide the evaluation results.

(15%) Your effort (discussion, error analysis, effectiveness of each component).

(You will get at least 10% if you use the optional unlabeled image set in open set setting)

(5%) Your performance is in the top 5 results.

Deal with the Dataset

First step, please download the dataset here.

When you extracting it, the architecture of the dataset will be like:

```
data/
-------- train/
---------- A/
---------- B/
---------- C/
-------- test/
---------- closed_set/
------------------------- test_pairs/
------------------------- labels.txt
---------- open_set/
------------------------- test_paris/
------------------------- unlabeled_data/
------------------------- labels.txt
```

During training, you can access all images in the **train** folder, please make sure you don't access any images in the test folder. The sub-folder A, B, C contains face images, face images with detection, face images with detection and alignment. For example, if there is a image named 丁守中_8.jpg in A, you can find its detection version in B, and the alignment version in C. That's helpful to build components to extract the position of the face out from images.

We don't set any limitation for the training procedure, just make sure you only access images in the **train** folder.

During testing, you need to read two images each time in the test_pairs folder, each pair will be named as **test_pair_id_1.jpg**, and **test_pair_id_2.jpg**, there are a total 1000 pairs in each track. Your model needs to output these two images are the same person (**1**) or not (**0**).
Please use the labels.txt to calculate the accuracy and list it in the report.
For the open-set track, there are 2,000 unlabeled images that can be used for fine-tuning before verifying the test pairs. You can apply any unsupervised/semi-supervised methods to assist your models. Please don't try to find its real labels. In addition, we sampled the unlabeled images randomly, therefore, it's possible that some people don't appear in unlabeled images.

Please make sure the label.txt is only used for accuracy calculation, **DO NOT** use the label to train/fine-tune/any other operation to your models.