Task 1

Codes for task 1 are stored in task1 folder, including additional two classes written in inference.py and evaluation.py.

The accuracy for testing set prediction is 98.73%

Some observations: when I decrease the epoch number to 2, the model is surprisingly having 97.55% accuracy. For further training, the accuracy doesn’t have much improve.

Task2

1. There are ResNets with 18, 34, 50, 101, 152 layers. In latter part of the paper, the author also proposed ResNet with 20, 32, 44, 56, 110, and 1202 different layers. Therefore, there are 11 different ResNet been proposed in the paper.
2. ResNet has lower complexity than VGG in depth. Even the deepest (152 layers) ResNet proposed in the paper is having lower complexity than VGG nets.
3. The input Size is 224x224.

Task3

When I tried to run my code, it shows ResolutionImpossible error. I ran all the code from Google Colab and record the following observations. The code file is test.py. I also attached the code run on Google Colab, test.ipynb. All code and images are in task3 folder.

1. The verification task is not as idea as I imagined. When I verify one person in two images, if one of the image has the person hindered, by a microphone in my example (image1.jpeg), the verify function would return false. However, when I tried images from different viewpoint, it can still successfully verify the two images as the same person. The verification of single-person image and a crowd image would have poor performance as well. Among all combinations, only image2 (single person image) and 5 (crowd image) are verified to be the same person. Interestingly, the detected person from image5 isn’t even the same person as image 2.

Print results[0-7] to see one of the results specified in the code

1. The facial attribute analysis would be less reliable especially for some ambiguous attributes such as age and emotion. The four images used for facial verification are used for facial attribute analysis. The age would highly depend on the light, viewpoint, hinderance, etc. Also, emotion would also be ambiguous to describe, and the expressions of different people showing the same emotion would also vary a lot. In contrast, gender and race are two attributes that are relatively easy to identify. In addition, running facial attribute on images with crowds would be a completely disaster.
2. The face detection would have perfect performance when there’s only one person in the image. When there’s more than one person in the image, the face detection would have trouble identifying the main character. I ran the function on Google Colab, the printed results are as follow:

Image5\_crowd.jpeg

*A picture containing person, sport, player, crowd

Description automatically generatedA screenshot of a cell phone

Description automatically generated with medium confidence*

Image6\_crowd.webp

A picture containing graphical user interface

Description automatically generated

Image7\_crowd.jpg

Graphical user interface

Description automatically generated

I guess there’s a rule, such as the clarity of face, for deciding face priority and get the face with highest priority. However, the model performs poorly in terms of detecting the main character in the image. The model doesn’t even necessary detect the most clear human face within the image. As we can see in image5 and 7, the model failed giving Trump’s face, but rather random people from the crowd.

For question 2 and 3 in task3, you may alter the parameters of given function to see the results of the pictures you want. The image paths are stored in list image\_names. Use image\_names[0-6] in parameter to run the results.