Approach Used

We have given around 23,500 sets of pairs of images for training, and we have to predict the labels of around 5000 pairs of images whether they are the same person or not.

- First we have used the already trained **Facenet network** to extract the embeddings of each of the images into a **128** sized vector
  Paper for Facenet network can be found here: https://arxiv.org/pdf/1503.03832.pdf

  This **Facenet architecture** is used to convert the RGB images into their characteristic 128 sized vector embeddings, and these vector embeddings are further used to check whether the images are of same person or not by calculating Euclidean distance between them.
  As per the paper, if the Euclidean Distance is less than 0.7, the images belong to the same person otherwise not. But we have used the Facenet architecture just to collect the embeddings of the pair of images.
- Before the implementation of the Facenet to encode into embeddings, **cv2** is also used to **get the face** from the image by cropping the image and extracting out the facial features only for effective learning and remove the background disturbances

- Next, for each set, we have 2 vector embeddings for a pair of images, each 128 sized vectors, we concatenated it to make a single 256 sized vector, which will be fed to the training model.
- **Artificial Neural Networks** are used to train over the embeddings of size 256 according to the provided labels. You can find the architecture in the **model.py** file

Details of ANN (model.py file)

- The pickle file which consists of the training data as embeddings is imported
- The two 128 sized embeddings are concatenated into a single 256 sized vector
- Then the model is built, where the input dimensions are of 256
- Batch normalization layer is used in each of the layers, there are total 7 layers in the model
- The last 2 layers contain the dropout of **0.25** so as to prevent from the overfitting
- **Binary Cross-Entropy** loss function is used and adam **optimizer** is used
- The activation function of **relu** is used in all the layers except the last one, where **sigmoid** is used

Results from the ANN

The ANN is trained over **150** epochs with **64** batch sizes, with around **2300** images in the validation set and around **21500** images in training set, the following results obtained:

- Training set accuracy: 0.988
- Training loss: 0.023
- Validation accuracy: 0.72
- Validation loss: 1.64

The model is then saved in **h5** format for the further predictions

Predicting the Results (run.py)

This file will run in 2 steps:

1. First, loading the test data to collect the final embeddings of pairs
2. Applying the models and predicting the results from the trained model on the embeddings, and at last saving the data in the form of **“results.csv”** file

- For embeddings, the file uses the same Facenet model along with the cv2 face cropping feature to make embedding of size 256 by concatenating two 128 embeddings
- The **trained_model.h5** file is loaded which is used to predict the test embeddings
- Finally, the results are saved in form of dataframe which further exported in form of **results.csv** file

Submitted by:
Saksham Gautam, sakshamgautam2001@gmail.com, IIT Bombay
Tithankar Adhikari, IIT Bombay