1 Introduction

In this exercise, you will implement K-Nearest Neighbour (KNN) algorithm and get to see it work on your image. Before starting on this programming exercise, we recommend clearing concepts of KNN.

Files included in this exercise

1. **KNN.ipynb:** Jupyter Notebook script that steps you through the exercise
2. **data.mat:** Mat file contains image data for assignment

You need to upload the following to your google drive before proceeding: ”KNN.ipynb”, ”data.mat” and your image. you need to open ”KNN.ipynb with google Colab”

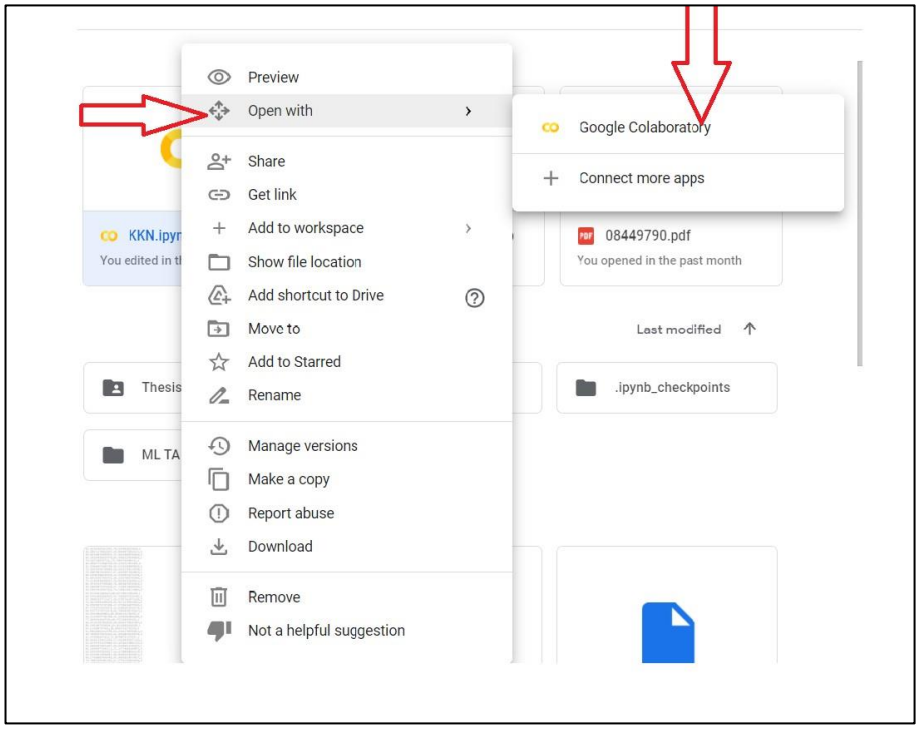

Figure 1: opening ipython notebook with colab

Throughout the exercise, you will be using the scripts KNN.ipyb. This scripts set up the data-set for the problem and contains portion of code that you will write. You only need to fill in the codes for the required tasks.

The data-set ”data.mat” consists of 50 32x32 color images of actors in 10 classes (10 actors), with 5 images per class (5 images of each actor).

Basically a grey scale images is stores as two-dimensional matrices, in which each element of the matrix corresponds to a single discrete pixel in the image. (Pixel is derived from picture element and usually denotes a single dot on a computer display.) For example, an image of128x128 would be stored as a 128x128 matrix with 164,864 discrete pixel values between 0 to 255 each, which represents the intensity of individual pixel.

A Colour image is represented using a three-dimensional array. In colour images, the first plane in the third dimension represents the red pixel intensities, the second plane represents the green pixel intensities, and the third plane represents the blue pixel intensities.Each pixel value is between 0 and 255 which represents the intensity of each colour pixel. For example in our data set a 32x32 colour image has 1024 pixels for each of three colour and total 3072 pixel values between 0 and 255.

Data: A 50x3072 array of uint8s. Each row of the array stores a 32x32 colour image. The first 1024 entries contain the red channel values, the next 1024 are green, and the final 1024 are blue. The image is stored in row-major order. **labels:** a list of 50 names. The names at index i indicates the label of the ith image in the array data.

Here are the 20 random images shown:

Figure 2: Random images

2 Tasks

You want to check that which actor or actress you resemble. It is easier than you thought. You task is to fill the code for the portions mentioned in the script. Following are the steps that might help you to build classifier.

The first step is to read your image. You need to write the code to read your image and display it. There are multiple ways to do this. You can use OpenCV tool kit, or matplotlib to import your image. You can upload your image into Google Colab 'files' and import it in your code.

For example: the code below is for importing image in Colab which is uploaded in colab's current runtime 'files'.

```
import cv2 from google.colab.patches import
cv2_imshow img=cv2.imread("Robot.jpg")
cv2_imshow(img)
```

However this is not recommended because Colab runtime expires and you need to upload your image again when reconnect to new runtime. The better approach is to upload your image in Google drive and provide its path in the code.

The next task is to write the code to resize your image to 32x32x3. This is important because you need to compare your image with data-set which is also 32x32x3 sizes to avoid dimensional error.

Then you need to reshape your image as we reshaped the images the dataset and display its final shape.

Now you need to fill in the code for calculating Eucleadian distance between your image and data set.

In This section you need to write code for 1NN. For this find the minimum distance and at which point minimum value exists. You need to display the image that most resembles you.

Now you need to fill in the codes for 3NN and 5NN You can use majority voting scheme for k=3 and for k=5. for this,find minimum distances and their instances You can loop through distances to find minimum. you need to display images that resembles you. Don't be surprise as you may observe gender change!.