Intelligent Data Analysis

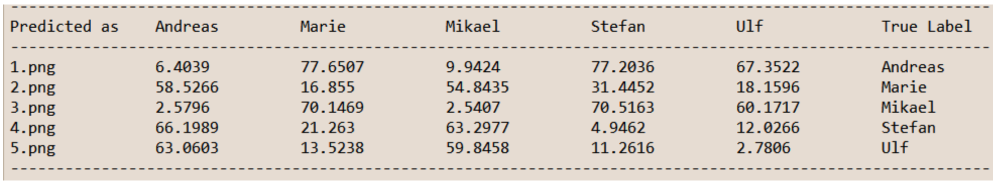
DV1597 (Exercise 4)

A Simple Face Recognition Algorithm

Preface: "Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand." [*Ref: What is computer vision? by IBM*]. Face recognition is a subfield of computer vision which aims at identifying or verifying the identity of an individual using their digital face [*more insights into this field is available online*].

The task: You have a data set of different faces (5 persons) in the folder "**People**". Your task is to writea Python/MATLAB code that automatically recognizes the faces in the folder "**Test**" (you have 5 faces) using the *Euclidean distance* metric. The final result should assign the 5 faces to their correct names based on the smallest *Euclidean distance* as shown below:

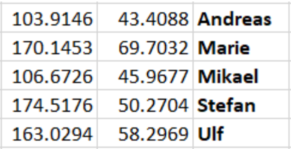
Table.1. Calculated distances of images in "Test" to the means of other people (100% accuracy!). The prediction is the one with the least distance (diagonal direction). Example: The face in the image "4.png" is predicted to be "Stefan" as it has the least Euclidean distance (4.9462).



```
----------------------------------------------------------------------------------------------
Predicted as    Andreas         Marie           Mikael          Stefan          Ulf             True Label
----------------------------------------------------------------------------------------------
1.png           6.4039          77.6507         9.9424          77.2036         67.3522         Andreas
2.png           58.5266         16.855          54.8435         31.4452         18.1596         Marie
3.png           2.5796          70.1469         2.5407          70.5163         60.1717         Mikael
4.png           66.1989         21.263          63.2977         4.9462          12.0266         Stefan
5.png           63.0603         13.5238         59.8458         11.2616         2.7806          Ulf
----------------------------------------------------------------------------------------------
```

How it works?

Step 1: First calculate the statistical *mean* and the standard deviation (*std*) of all images in each of the 5 folders (Andreas, Marie, Mikael, Stefan, Ulf). Then calculate the average of the *means* and the average of the *stds*, you will get something like:



$\bar{\mu}$	$\bar{\sigma}$	
103.9146	43.4088	**Andreas**
170.1453	69.7032	**Marie**
106.6726	45.9677	**Mikael**
174.5176	50.2704	**Stefan**
163.0294	58.2969	**Ulf**

Step 2: Calculate the *mean* and *std* of each of the images in the folder "Test".

Step 3: Calculate the *Euclidean distances* of each image in the folder "Test" to the mean of other people as shown in Table.1 (the program's output should be as in the table).

$$d_{tkpj} = \sqrt{\left(\mu_{tk} - \bar{\mu}_{pj}\right)^2 + \left(\sigma_{tk} - \bar{\sigma}_{pj}\right)^2} \qquad \textbf{Eq. (1)}$$

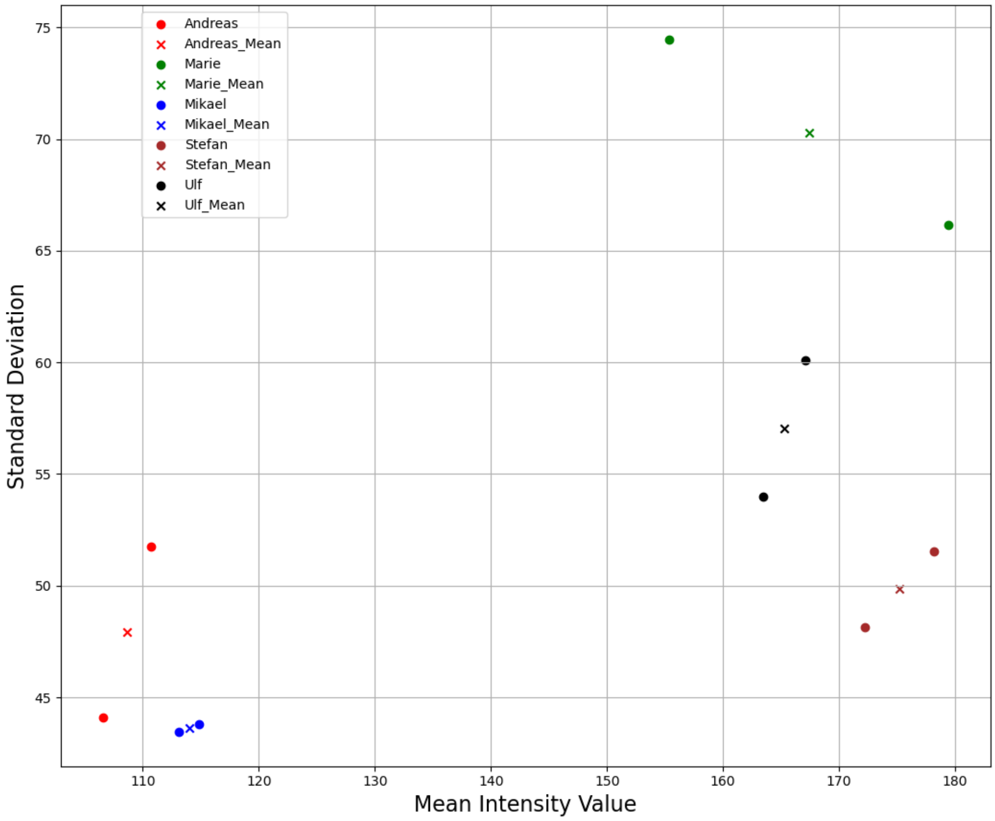
Where *t* denotes test folder, *p* denotes people folder and *k*, *j* are the indexes of each image. For example, when *k*=1 and *j*=1, we see that: $d_{t1p1} = 6.4041$, and when *k*=2 and *j*=4, we see that: $d_{t2p4} = 31.4496$; as shown in the table in Table. 1.

You need to submit a code (written in Python or MATLAB) that when executed it displays the following two outputs:

1 A scattered plot of the results from step 1. To guide you, an example using only 2 images for each person is shown in the below figure, **but you need to submit the plot for all 9 images**. Make sure to have the *x*, *y* labels and the legends.

2 The distance table as shown in Table.1 using Eq (1).

3 **Optional task:** Discuss the most interesting pattern that you observed in the generated figure. Was it easy to spot such pattern in the table?


Andreas
Andreas_Mean
Marie
Marie_Mean
Mikael
Mikael_Mean
Stefan
Stefan_Mean
Ulf
Ulf_Mean
Standard Deviation
Mean Intensity Value