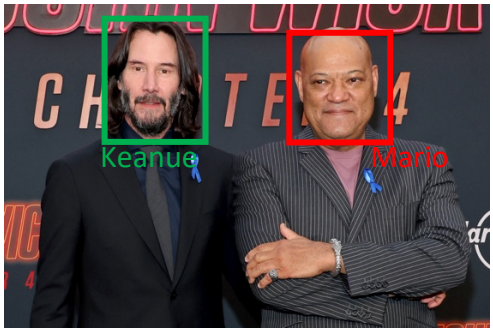
Assignment 2: Classical Face Recognition
Deadline: 30/04/2023

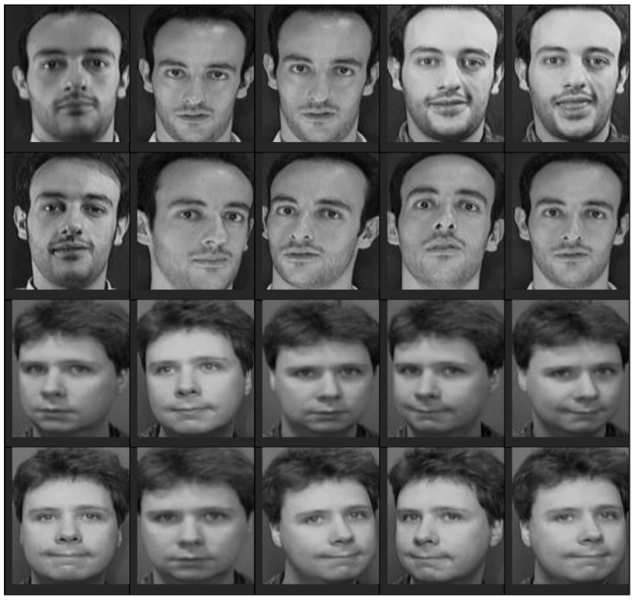
In this assignment, you will implement a "closed-world" face recognition system based on eigenfaces. By using **one** of the Hikvision camera in the CVIP lab, you will **produce a video** of persons moving in the lab. At each frame, close to the detected faces, you will show labels with the names of the recognized persons. For instance:



Here some details on how to solve the problem:

Resize Images to a size 64 x 64.

a. To extract the eigenfaces-based descriptor (Turk-Pentland), you will need to **estimate** a good **Face Space**. You will download and use the "**Labeled faces in the wild**" dataset (Huang et al. 2007) to estimate the **mean face** and the **relevant eigenfaces**. I suggest using the dataset "*All images aligned with funneling*" at the link http://vis-www.cs.umass.edu/lfw/. These images are already aligned. It is not required that you use all the images in the dataset. However, I would use at least 2000 **face** images to get a good face space (run your face detector on the images to get face images!). Building the face space can be done *just once*. If you need to save the results, use the pickle package.
b. Since the system is "closed", you need to **create a gallery** of the identity to recognize. You can imagine the gallery as **a set of labeled face images**. It is advisable that several face images are collected for each identity, and that the face images are taken with different face expression and head orientation. The gallery should contain **at least 4 persons** but you can include more identities if you like. The gallery is built once. You can store the gallery on disk by the pickle package.
Example of face gallery:



c. To detect the faces to build the gallery and implement a (almost) real-time identity recognition, you will use the OpenCV implementation of the **Viola-Jones face detector**. You'll need to download from the OpenCV public repository on github the file '*haarcascade_frontalface_alt.xml*'.
d. You will need a classifier to associate the face descriptor of a face to a known identity. You can use a **K-NN classifier** (if you like, you can include results with other classifiers, but you will need more data).
e. You will need a manually annotated validation set for model selection. A video of 20-30 frames of the persons moving in the lab will suffice. This set of images must not overlap with the training set (images in the gallery) nor with the test set.

Expected Results:
You will have to prepare a Jupiter notebook to solve the problem. The code must be properly commented, and implementation details clearly stated.

It is mandatory to include, in the proposed solution, an analysis of the achieved results:

1. Results achieved by varying the number of eigenfaces. The number of eigenfaces should be selected in order to cover at least 95% and 99.99% of the total variance (remember that the covered variance is obtained by analyzing the normalized singular values!).
2. Results achieved by varying the value of K in {1, 3, 5}.

Choose the value of K and of the covered variance on a validation set in order to **select the best model**. Report the number of errors made by the system on the validation set in a table and choose the model accordingly.
Then, use the selected model to test the system on a video of at least 2000 frames and produce a video with the results. The test video must not overlap with the training or the validation set. It is not required to present quantitative results on the test set.

***Non mandatory**:*

- *if you want, you may try to align the faces you collect to the mean face by using the eyes. Eyes can be found, for instance, by another cascaded classifier (see the slides), which is not perfect. Alignment requires a 2D roto-translation. Alignment of the faces should improve the face recognition system. You are free to explore also other tools.*

- *If you want, you may decouple the acquisition of the frames and the processing by using a multi-thread approach (this is useful if you want to directly process the frames from the camera).*