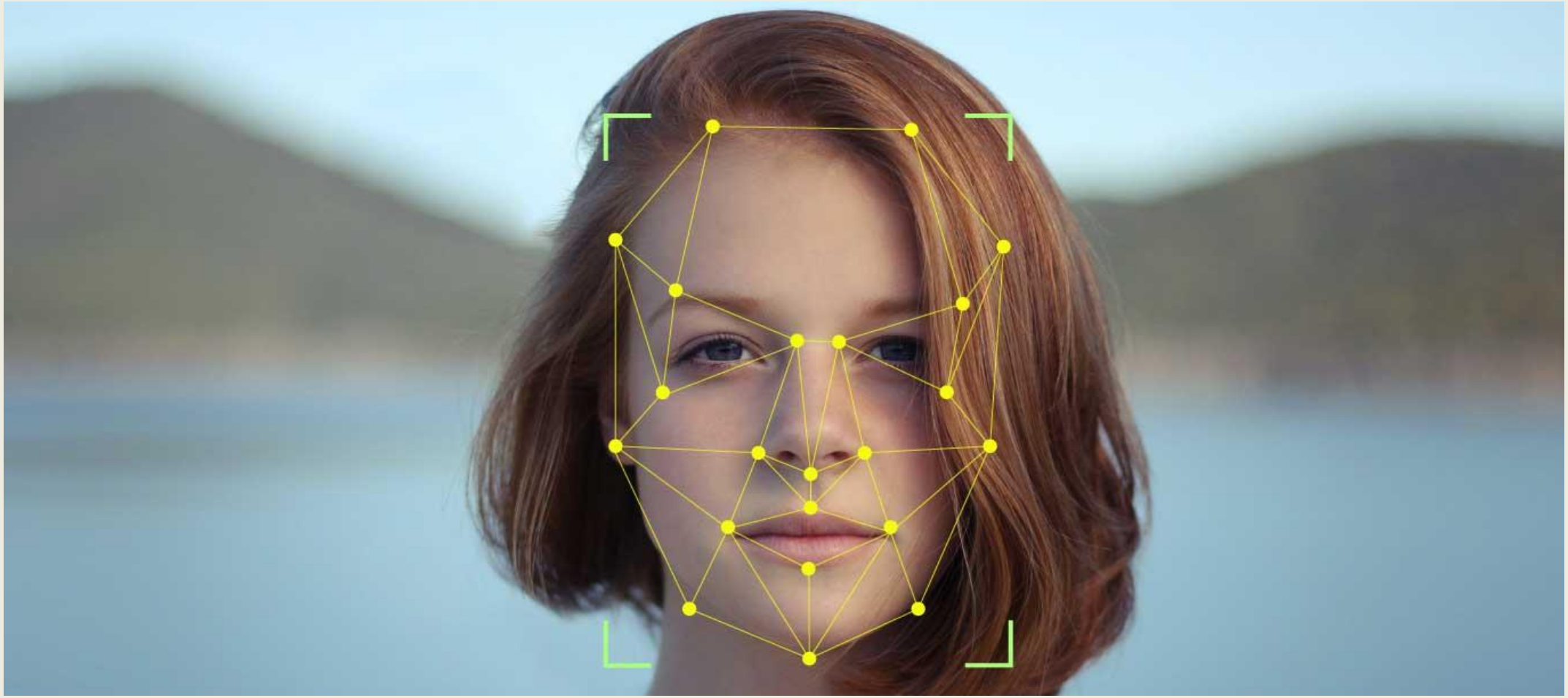
Face Recognition

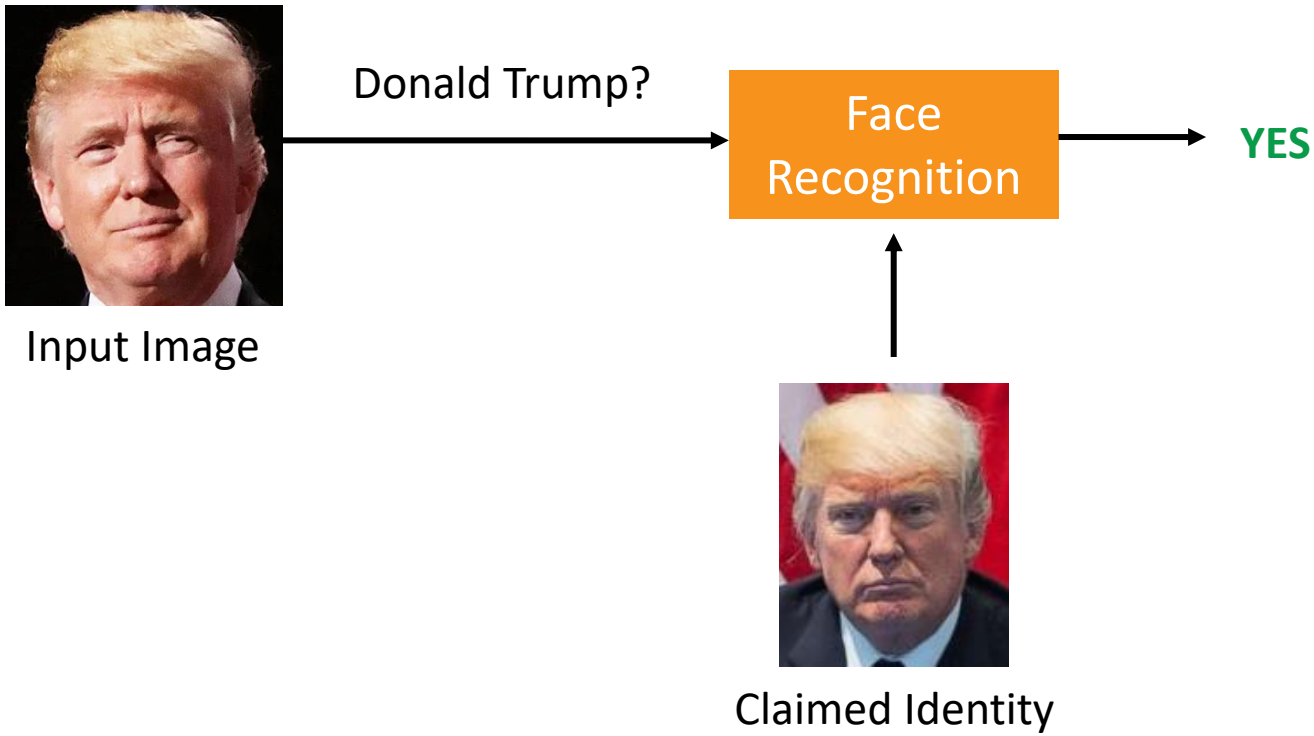


Ruben Tolosana
ruben.tolosana@uam.es

BiDA Lab
Biometrics & Data Pattern Analytics Lab

UAM
Universidad Autónoma de Madrid

Face Verification



Automatic Face Recognition: Architecture

Donald Trump?

Input Image

Automatic Face Recognition: Architecture

Donald Trump?

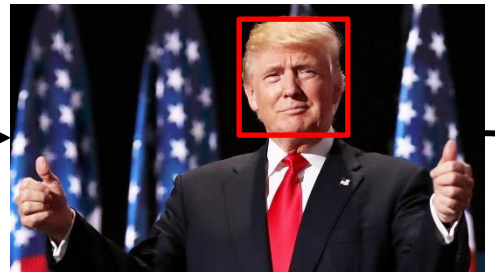
Input Image

Automatic Face Recognition: Architecture

Donald Trump?

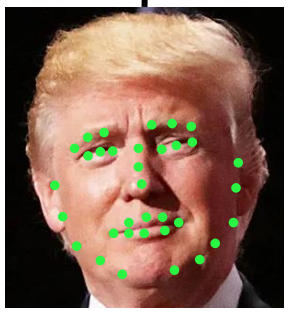
Input Image

Face Detection



Face Alignment

Landmark Detection

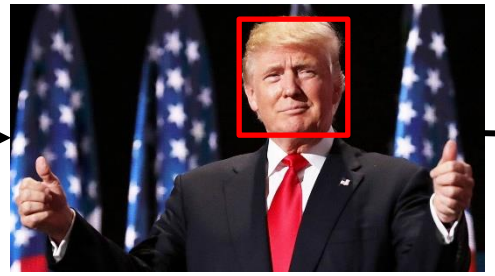


Automatic Face Recognition: Architecture

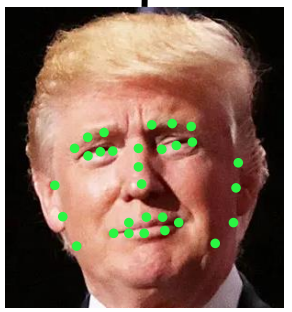
Donald Trump?

Input Image

Face Detection



Face Alignment



Landmark Detection

Presentation Attack Detection

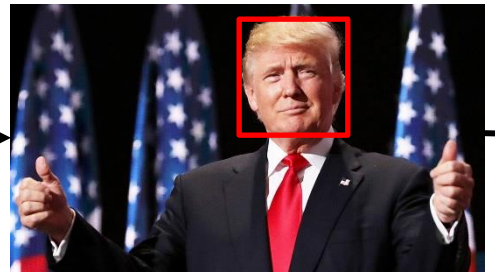
Real or **Fake**?

Automatic Face Recognition: Architecture

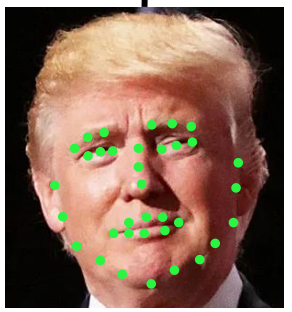
Donald Trump?

Input Image

Face Detection



Face Alignment



Landmark Detection

Presentation Attack Detection

Real or **Fake**?

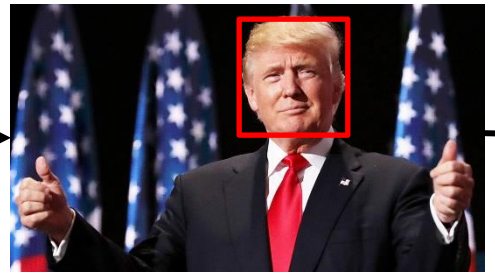
Face Recognition

Automatic Face Recognition: Architecture

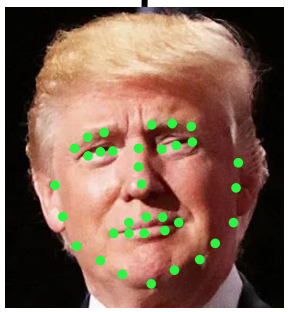
Donald Trump?

Input Image

Face Detection



Face Alignment



Landmark Detection

Presentation Attack Detection

Real or **Fake**?

Face Processing

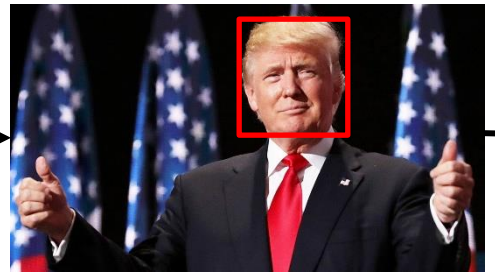
Face Recognition

Automatic Face Recognition: Architecture

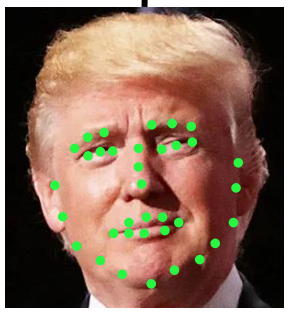
Donald Trump?

Input Image

Face Detection



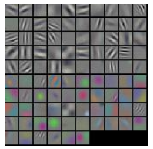
Face Alignment



Landmark Detection

Presentation Attack Detection

Face Recognition



Feature Extraction

Face Processing

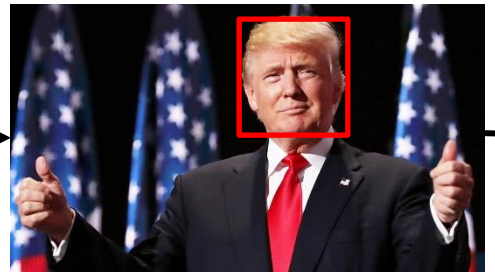
Real or **Fake**?

Automatic Face Recognition: Architecture

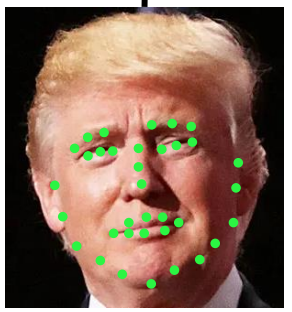
Donald Trump?

Input Image

Face Detection



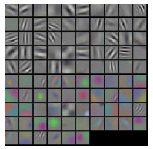
Face Alignment



Landmark Detection

Presentation Attack Detection

Face Recognition

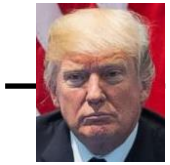


Face Processing

Feature Extraction

Face Matching

YES



Claimed Identity: Donald Trump

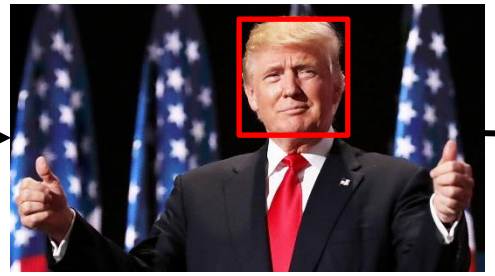
Real or **Fake**?

Automatic Face Recognition: Architecture

Donald Trump?

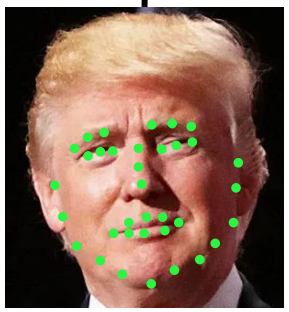
Input Image

Face Detection



Face Alignment

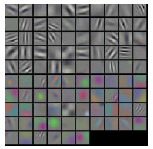
Landmark Detection



Presentation Attack Detection

Real or **Fake**?

Face Recognition

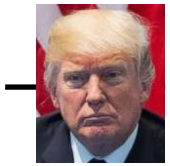


Face Processing

Feature Extraction

Face Matching

YES

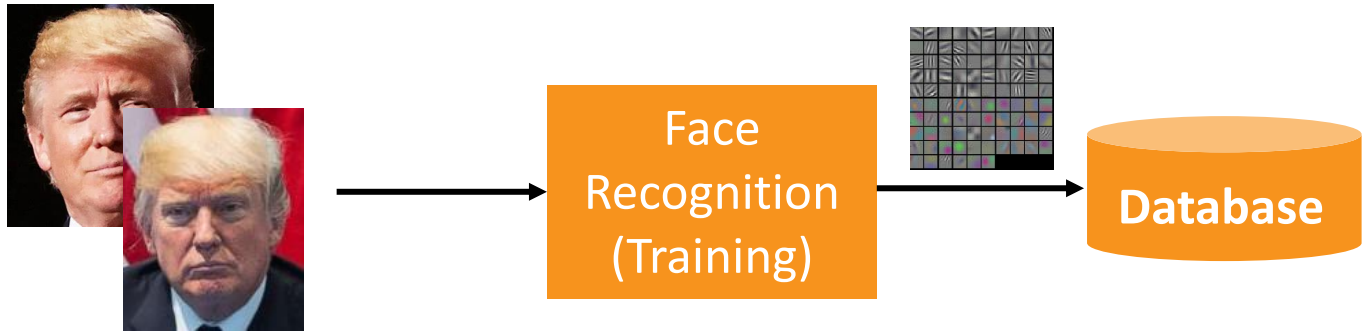


Claimed Identity:
Donald Trump

Automatic Face Recognition: Phases

Biometric recognition systems comprise 2 phases:

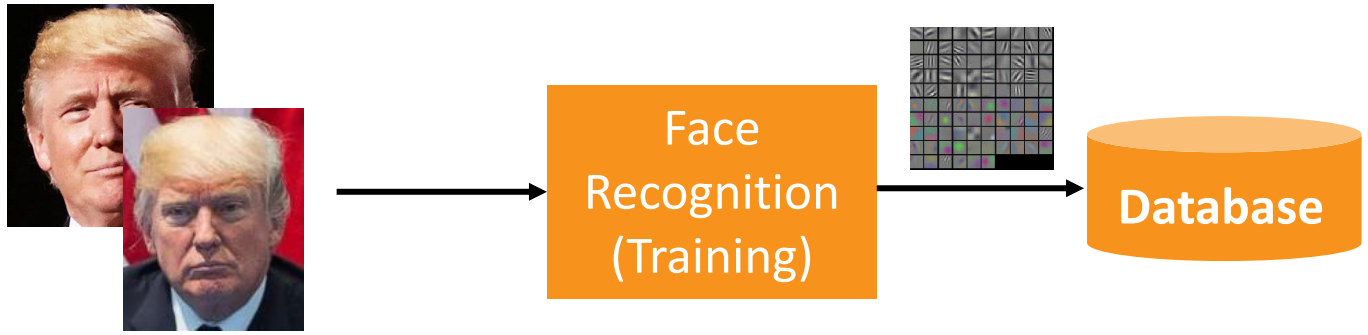
- Training (modeling the users): the system acquires images from the user. Each image is processed to detect the face, normalize it, extract the features and store them into the database.



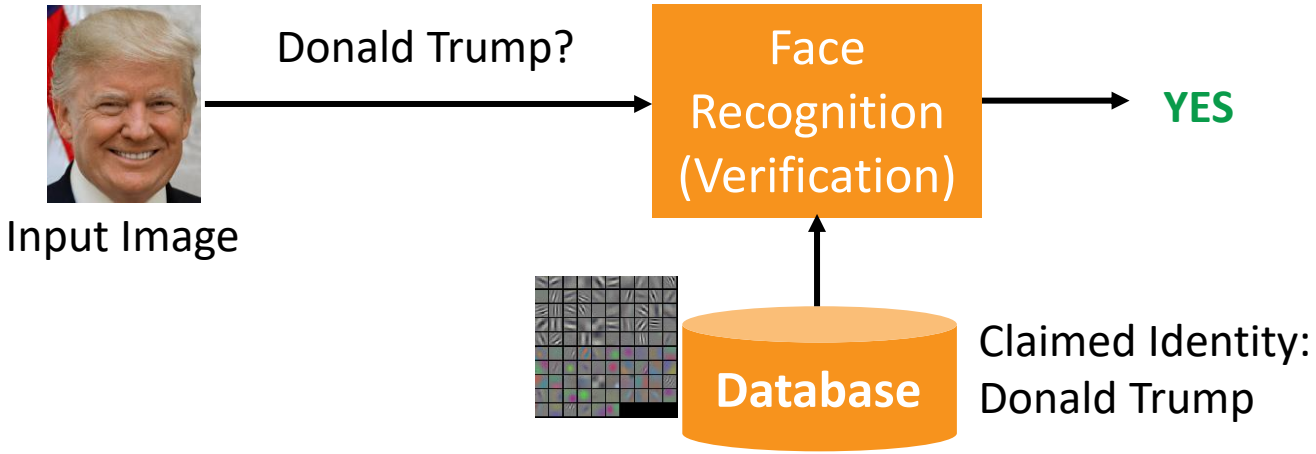
Automatic Face Recognition: Phases

Biometric recognition systems comprise 2 phases:

- Training (modeling the users): the system acquires images from the user. Each image is processed to detect the face, normalize it, extract the features and store them into the database.



- Verification (recognizing the users): a new image is compared with the template stored in the database.



Database

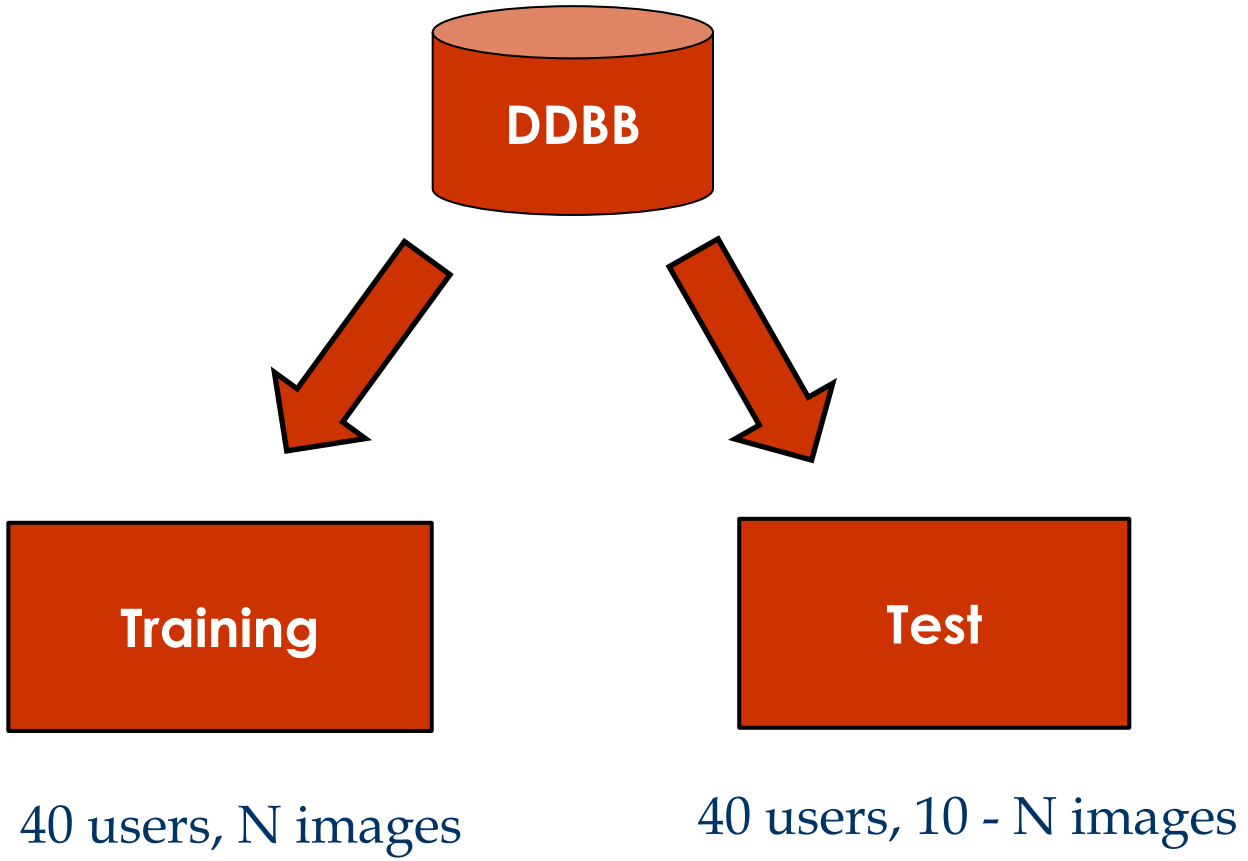
ATT Face Database:

- 40 users, 10 images/user, 1 session.

Link: https://git-disl.github.io/GTDLBench/datasets/att_face_dataset/

Database

ATT Face Database:



Feature Extraction and Matching

Feature Extraction:

- Simple approach based on 2D DCT (Discrete Cosine Transform) coefficients:
- DCT: represents an image as a sum of sinusoids of varying magnitudes and frequencies. For an image, most of the visually significant information is concentrated in just a few coefficients of the DCT (Energy Compaction).
- Matlab code: feature_extraction.m

Feature Extraction and Matching

Matching:

- The features of the Training phase are compared with the input features of the Test.
- Matlab code:

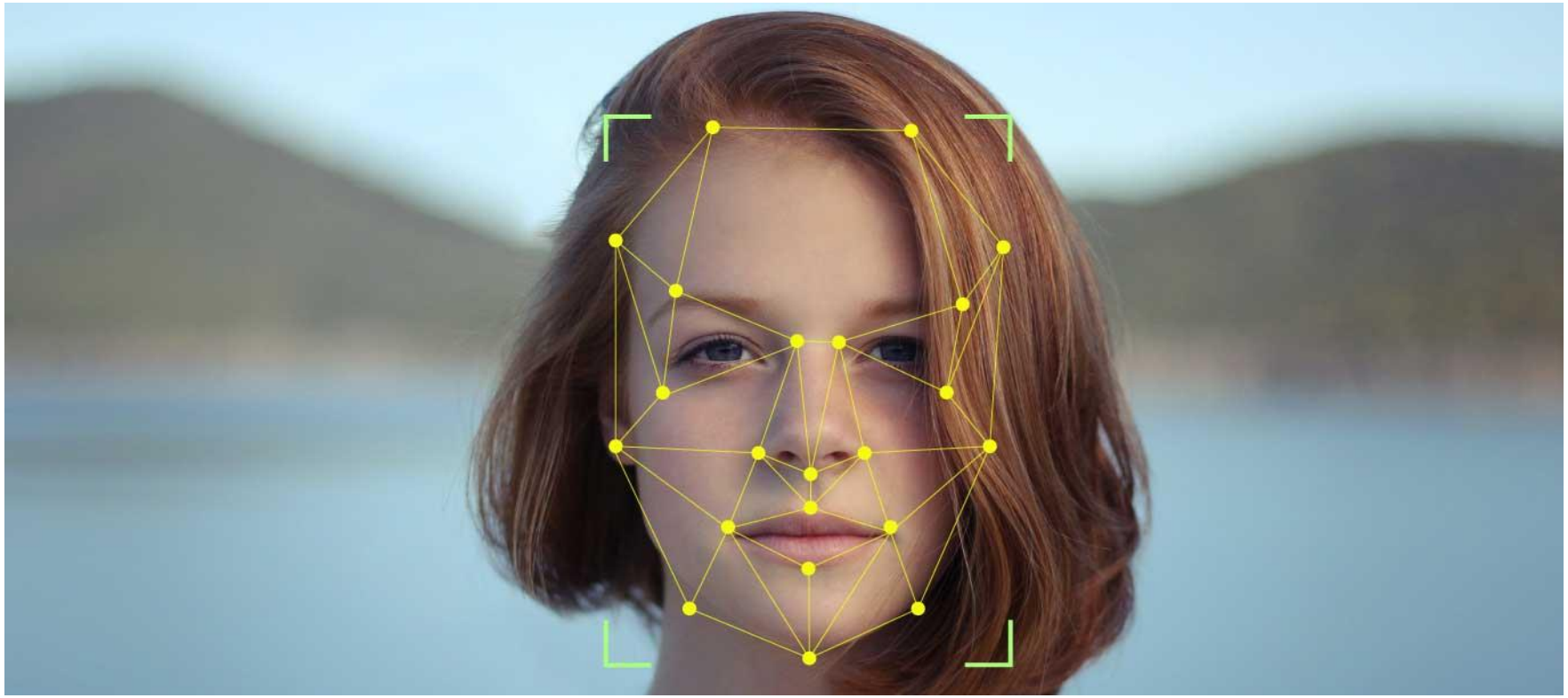
```
my_distance(contTest)=mean(abs(MatrizTest(i,:)-MatrizEnrolment(j,:)));
```

Face Recognition System:

- Matlab code: FaceRecognition.m

TASK 1

UNDERSTAND AND PLAY WITH THE PARAMETERS OF THE SYSTEM

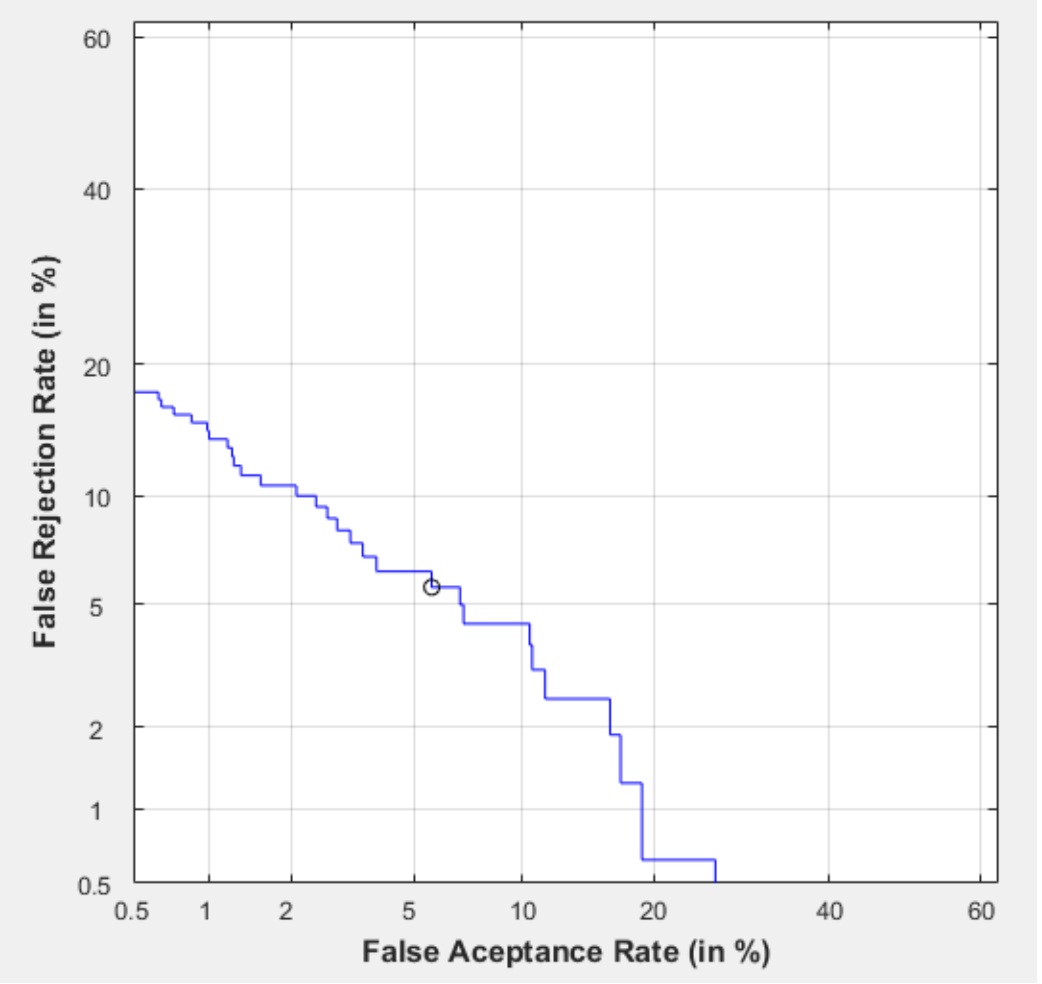


Baseline Results

Parameters:

- Number of DCT coefficients: 10
- Number of Training images: 6
- Number of Test images: 4

Equal Error Rate (EER) = 5.6%

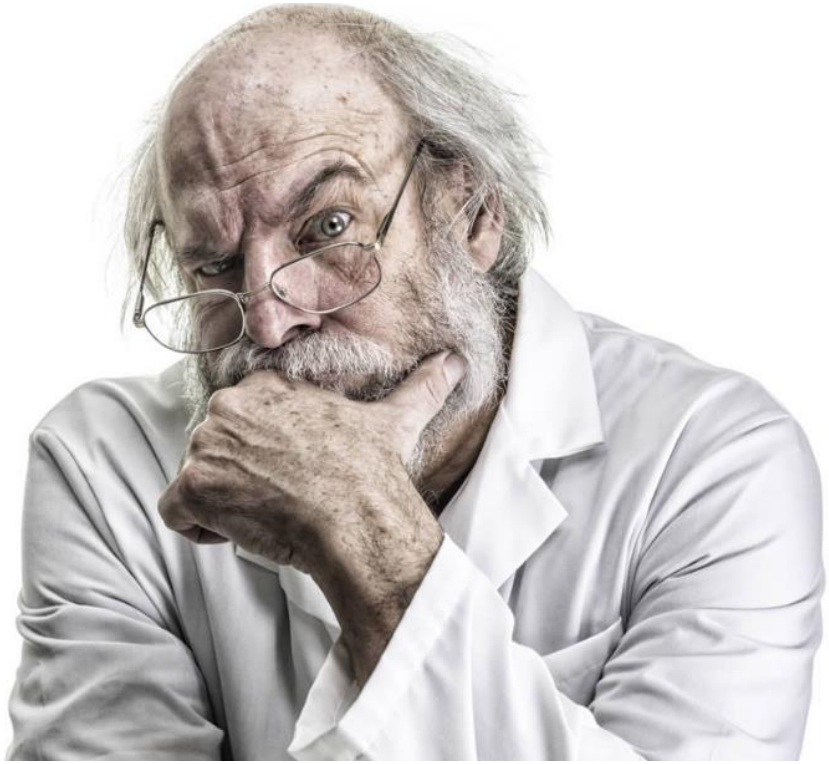


Tasks:

- Understand the code.
- Which parameters provide the best results?

TASK 2

IMPROVE THE FEATURE EXTRACTOR: PRINCIPAL COMPONENT ANALYSIS



Baseline Results

Task 2: Use Principal Component Analysis (PCA) instead of DCT coefficients.

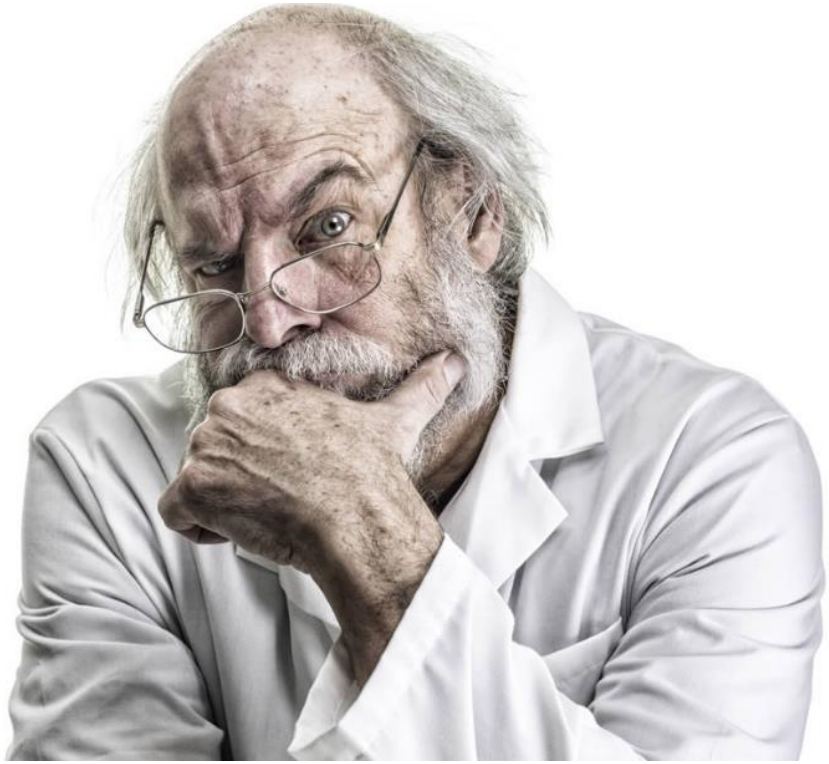
- Matlab code: pca.m
- With the Training data:

```
[coeff_PCA,MatrixTrainPCAFeats,latent] = pca(MatrixTrainFeats);
meanTrainMatrix=mean(MatrixTrainFeats);
```

- With the Test data:
  - For each test, subtract the meanTrainMatrix, and multiply by the coeff_PCA transformation matrix.

TASK 3

IMPROVE THE MATCHING MODULE: SUPPORT VECTOR MACHINES



Baseline Results

Task 3: Change the simple similarity distance by a more sophisticated method, such as Support Vector Machines (SVM) classifier. You must continue using PCA features.

- If you choose SVM, then you have to Train model per user with the training data:

  See: SVMModel = fitcsvm(...)

- Then, for the Test you have to use:

  [label,score]= predict(SVMModel,MatrixTestFeats); %to obtain the scores for each user model