Faculty Of Computers and Artificial Intelligence

Cairo University

Image processing

Phase (3)

AYA SABRY 2018035
Gehad Mostafa 20180080
Rana Mohamed 20180104

Submitted to
Eng.Ibrahim Zedan

May 2022

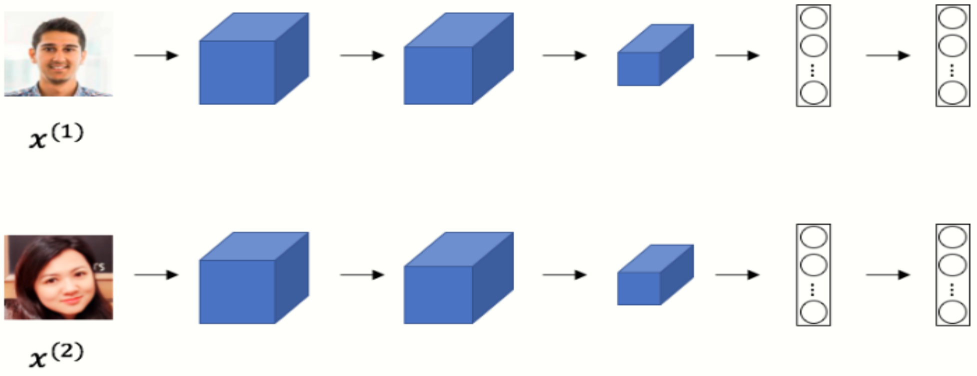
Abstract

- Face recognition identifies persons on face images or video frames. In a nutshell, a face recognition system extracts features from an input face image and compares them to the features of labeled faces in a database.
- Comparison is based on a **feature similarity** metric and the label of the **most similar database** entry is used to label the input image.
- If the similarity value is below a certain threshold the input image is labeled as ***unknown*****.**
- Comparing two face images to determine if they show the same person is known as face verification.

Explanation of the proposed system

- Detect, transform, and crop faces on input images. This ensures that faces are **aligned** before feeding them into the CNN.
- This preprocessing step is very important for the performance of the neural network.
- Use the CNN to extract **128-dimensional** representations, or *embeddings*, of faces from the aligned input images.
- In embedding space, **Euclidean distance** directly corresponds to a measure of **face similarity.**
- Compare input embedding vectors to labeled embedding vectors in a database. Here, a **support vector machine (SVM) and a KNN classifier,** trained on labeled embedding vectors, play the role of a database.
- The CNN architecture used here is a variant of the inception architecture
- More precisely, it is a variant of the NN4 architecture and identified as nn4.small2 model in the OpenFace project with **128 hidden units** followed by an **L2** normalization layer on top of the convolutional base.*(Siamese Network)*
- These two top layers are referred to as the ***embedding laye***r from which the 128-dimensional embedding vectors can be obtained.

Siamese Network



$$\mathcal{L} = max(d(a,p) - d(a,n) + margin, 0)$$

Triplet Loss Funciton

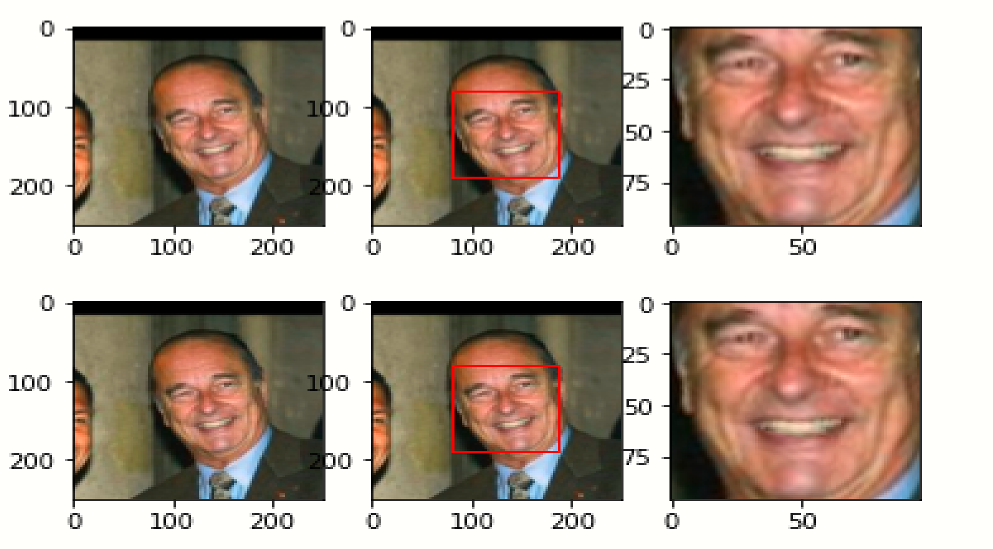
- In Siamese networks, we take an input image of a person and find out the encodings of that image, then, we take the same network without performing any updates on weights or biases and input an image of a different person and again predict it’s encodings.
- We compare these two encodings to check whether there is a **similarity** between the two images. T
- hese two encodings act as a latent feature representation of the images. Images with the same person have similar features/encodings. Using this, we compare and tell if the two images have the same person or not.
- We train the network by taking an **anchor** image and comparing it with both a **positive** sample and a **negative** sample.
- The **dissimilarity** between the **anchor image and positive** image must **low** and the **dissimilarity** between the **anchor image and the negative** image must be **high**.

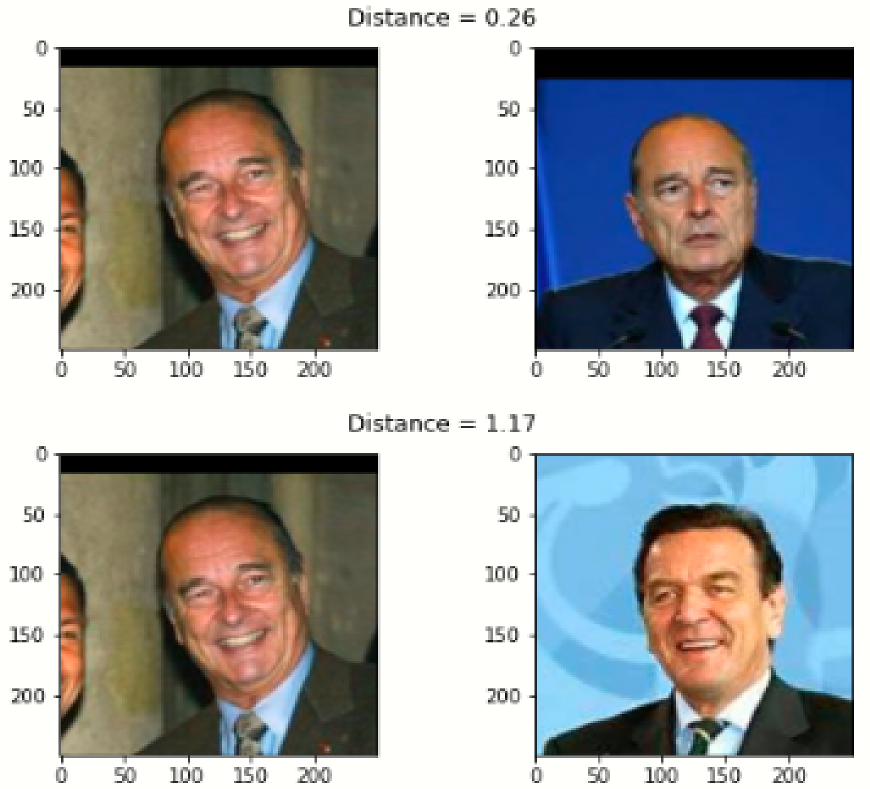
Experimental result

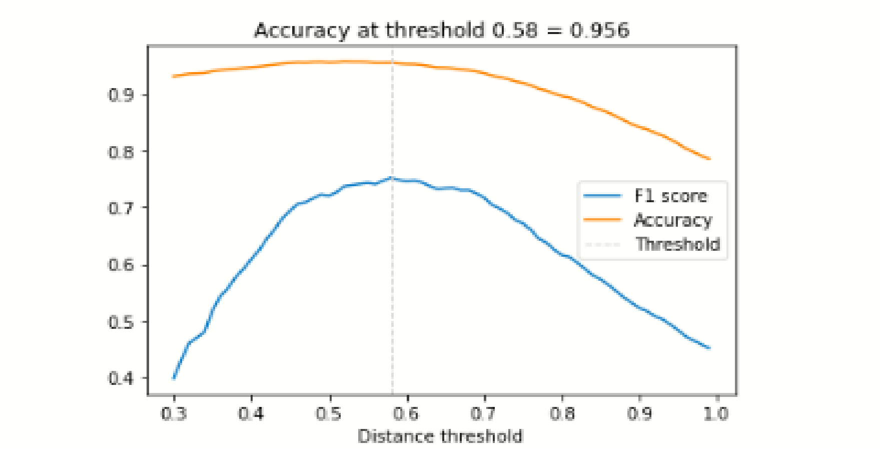
- The **KNN** classifier achieves an accuracy of **96%** on the test set, the **SVM classifier 98%.**
- Let’s use the SVM classifier to illustrate face recognition on a single example.
- Classification results should actually be checked whether (a subset of) the database entries of the predicted identity have a distance **less than τ** otherwise one should assign an **unknown** label.

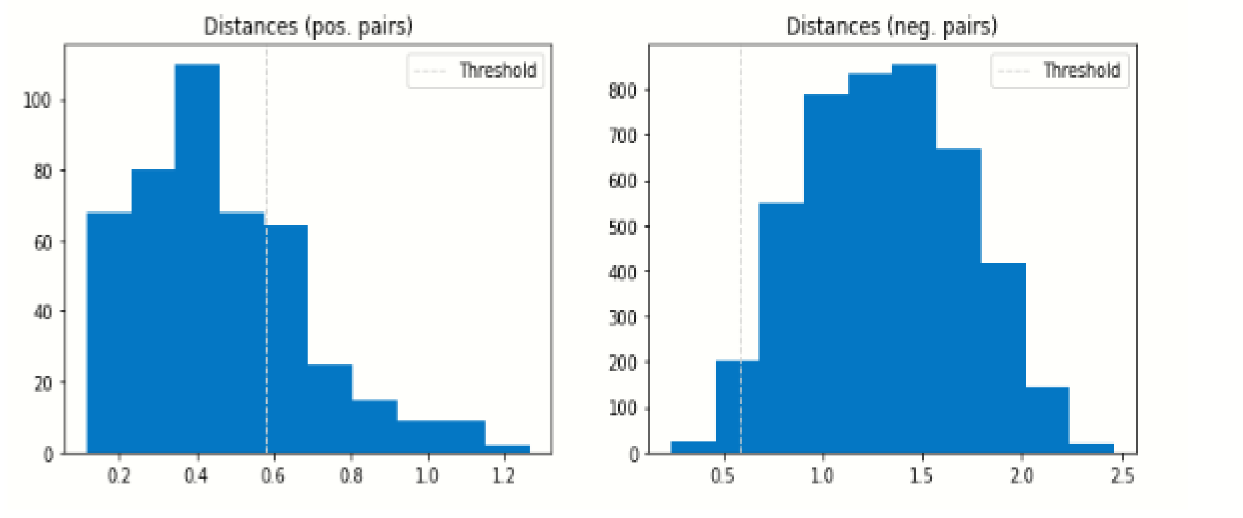
Screenshots from the program running

```
Epoch 1/10
100/100 [==============================] - 74s 579ms/step - loss: 0.7989

Epoch 2/10
100/100 [==============================] - 52s 519ms/step - loss: 0.8012

Epoch 3/10
100/100 [==============================] - 49s 487ms/step - loss: 0.8011


Epoch 4/10
100/100 [==============================] - 45s 449ms/step - loss: 0.8003

Epoch 5/10
100/100 [==============================] - 45s 448ms/step - loss: 0.7999

Epoch 6/10
100/100 [==============================] - 45s 454ms/step - loss: 0.8004

Epoch 7/10
100/100 [==============================] - 45s 453ms/step - loss: 0.8000

Epoch 8/10
100/100 [==============================] - 45s 454ms/step - loss: 0.8003

Epoch 9/10
100/100 [==============================] - 45s 449ms/step - loss: 0.8001

Epoch 10/10
100/100 [==============================] - 45s 450ms/step - loss: 0.8001
```

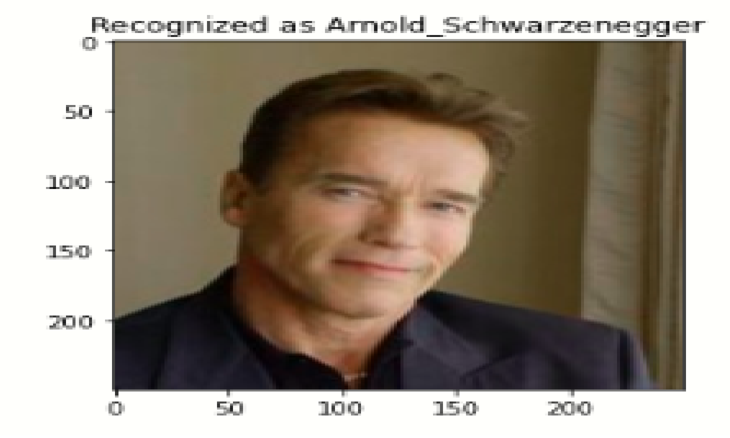

0
100
200
0
100
200
0
25
50
75
50

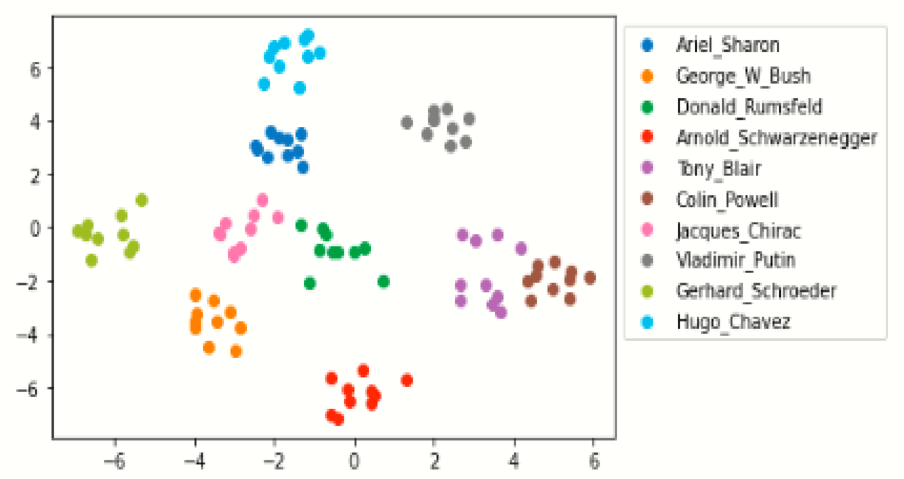

Distance = 0.26
Distance = 1.17
0
50
100
150
200





- KNN accuracy = 0.96, SVM accuracy = 0.98


Recognized as Arnold_Schwarzenegger
0
50
100
150
200
0
50
100
150
200


6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
● Ariel_Sharon
● George_W_Bush
● Donald_Rumsfeld
● Arnold_Schwarzenegger
● Tony_Blair
● Colin_Powell
● Jacques_Chirac
● Vladimir_Putin
● Gerhard_Schroeder
● Hugo_Chavez