Biometric-Based Office Access Control

A. Elsaadany, A. Hassan, and M. Abada, M. Elebody,

1

***Abstract*—**Security has been considered a significant concern in the last few decades. Providing reasonable security in modern technology and companies is a must because valuable information about the workers and the customers must be secured against any attacks. Thus, improved tools for access and personality identification should be invented. In the past, identification was based on passwords and usernames that could be easily broken. Nowadays, many companies require high technologies for their workers to login into their systems, such as fingerprint recognition, face recognition, and voice recognition. These tools have higher security than the old standard tools. As a result, a biometric-based office access controller is designed in this project using a face identification system. The system is based on four main layers: Face detection, face Alignment, feature extraction, and matching features. The first two layers are implemented using the libraries Dlib and OpenCV. The model "nn4_small.v1" extracts features from the input faces. The model consists of a pre-trained CNN network, and the last layer is modified to use the triplet loss calculation on our dataset split into the training and test sets. The test set is tested on a classification algorithm, clustering, and similarity calculations using the triplet loss.

I. Introduction

Big companies such as Apple, Google, Amazon spend too much money on research to improve their security mechanisms to secure their laptops or mobiles. An example of these security mechanisms is the Face recognition system that protects us against fake logins or data leakage. The system is based on these given layers:

1. Detection of Faces using Dlib and OpenCV
2. Face Alignment
3. We extract features such as the distances between noses and eyes or the shape of mouth and forehead.
4. We are matching these features to a known dataset.

The figures below show the flow chart of the face recognition systems:

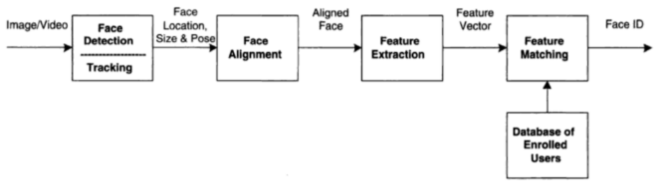


Fig. 1. Face recognition processing flow.

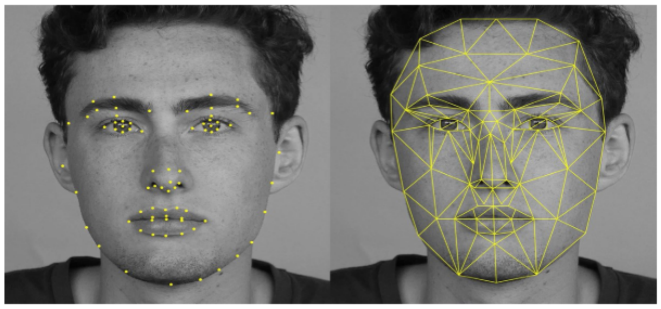


Fig. 2. Feature Extraction

I. Methods

The project consists of three main parts. In the first part, our dataset of the public figures is used, and faces are detected using the libraries Dlib and OpenCV. The library manages to detect the eyes, noses, and mouth and draw a rectangle around the face. Then, faces are aligned to give better accuracy. Finally, the output of the alignment is fed into the CNN network.

The pre-trained model "nn4_small2_v1" is used in the second part of the project. It is trained with almost 600,00 pictures and based on a convolutional neural network. It has 128 fully connected layers, followed by L2 normalization and convolutional base layer. Keras is used to implement the model. Embedding vectors are the output of the model. They are used in face recognition.

1

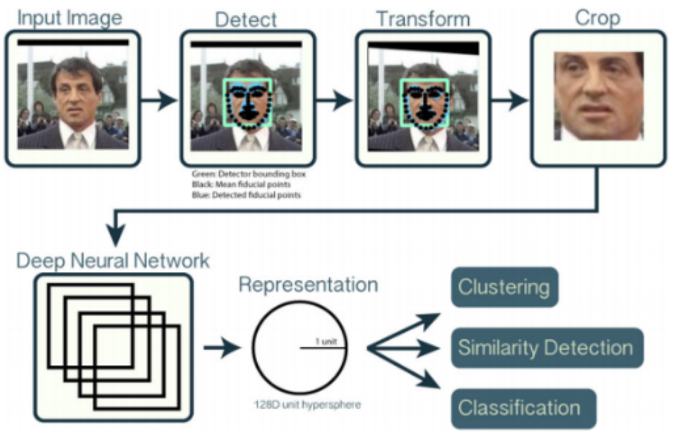


Fig. 3. Flow of the project

As shown in figure 3, in the last part, clustering, classification, and similarity detection using the triplet loss are done on the embedding vectors. Again, we used our dataset with 40 to 50 pictures of ten public figures.

a) As shown in figure 4, the triplet loss law compares three main pictures: The anchor picture with a positive picture for the same person and the same anchor picture with a negative picture for another person. All distances are calculated, and a threshold is calculated based on F1 and accuracy scores to determine the limit that differentiates and defines different people.

$$L = \sum_{i=1}^{m} \left[\| f(x_i^a) - f(x_i^p) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + \alpha \right]_+$$

Fig. 4. Law of Triplet Loss

b) The Dataset is split into a training set and a test set. A classification algorithm is used to classify people, and the accuracy is measured. Unknown faces are also tested.

c) Clustering is used to group the embedding vectors for the same person together to better visualize and differentiate between different people. Unknown faces are also tested.

d) Only the most prominent face is detected; however, all faces above a specific threshold are detected and recognized in testing the model.

II. Results

1. Examples from the detection and the alignment in the dataset used are shown below:

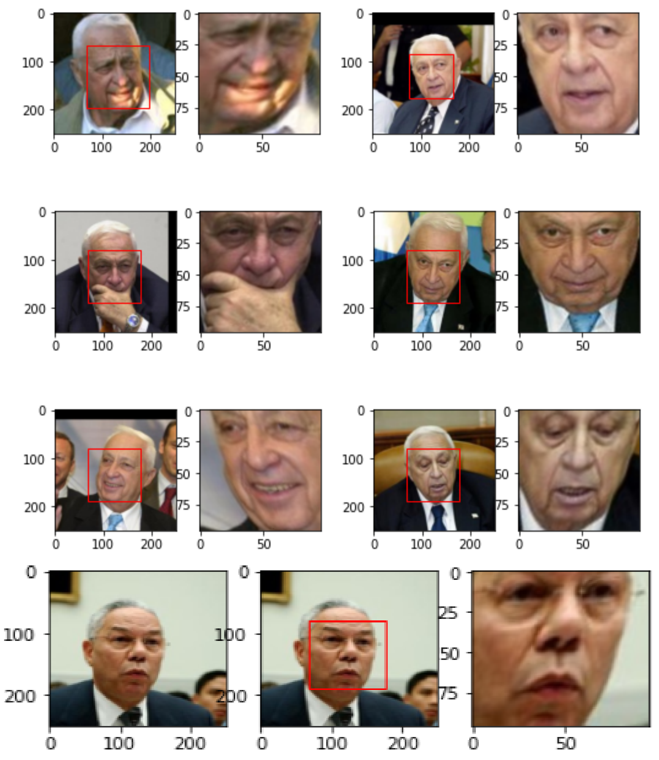


Fig. 5. Detected and Alignment Faces

2. Examples from the similarity detection and triplet loss calculation are shown in figure 6. The triplet loss distance between similar faces is small. On the other hand, it is significant in the case of different faces.

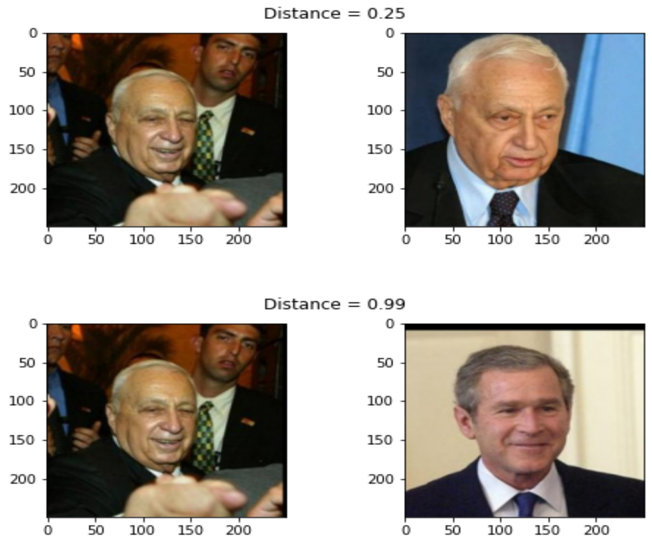


Fig. 6. Output Comparison between same faces and different faces

3. The threshold is determined through F1 Scores and accuracy scores, and the result is shown in figure 7. It is calculated by trying different thresholds and measuring the model's accuracy in each threshold, and the best accuracy is when the threshold is 0.64. Therefore, in the case of the previous comparison in figure 6, the distance 0.17 is smaller than the threshold. Therefore both pictures are for the same person. However, The distance in 1.70 is larger than the threshold. Thus, the two pictures are not for the same person.

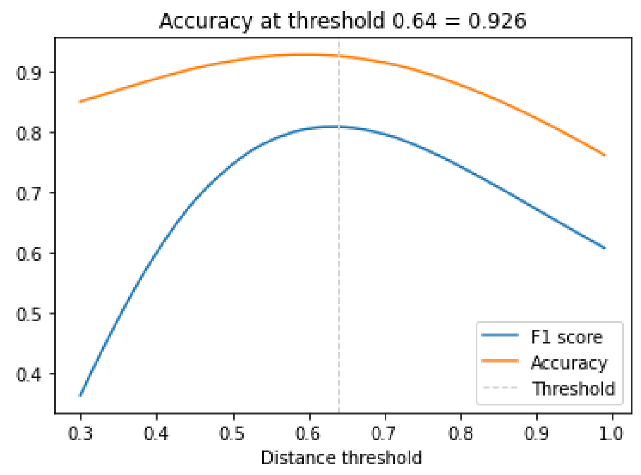
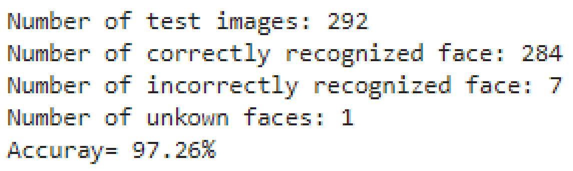


Fig. 7. Threshold Graph measured using F1 Scores and Accuracy Calculation

4. The Dataset is divided into labeled training sets used in training our model, while the test set is used to test our model. Classification is done based on the triplet loss calculations. The test picture is compared to all pictures of each person, and we get the minimum triplot loss distance. If the distance is less than the threshold, then the test picture is known and predicted for that given person. If the distance is larger than the threshold, it is an unknown person, and we should compare the test picture to another person. The accuracy of the model is 97.26 %. It is calculated by knowing the number of correct predictions from the total number of predictions, as figure 8 shows. Also, if the model could not classify an image, it labels the picture as unknown.



```
Number of test images: 292
Number of correctly recognized face: 284
Number of incorrectly recognized face: 7
Number of unkown faces: 1
Accuray= 97.26%
```

Fig. 8. Result of Prediction

5. Dataset visualization is made both in the training and the validation set to group the embedding vectors to visualize the features better. Also, unknown faces are tested.

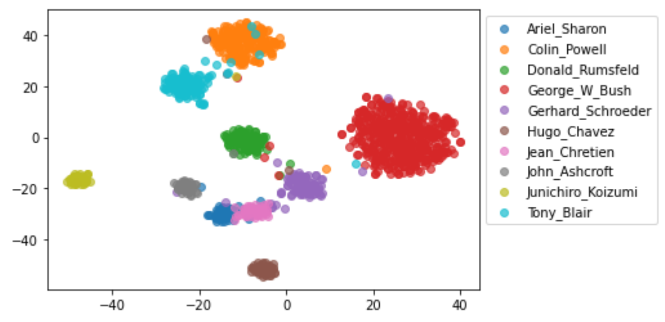


Fig. 9. Result of visualization of training set

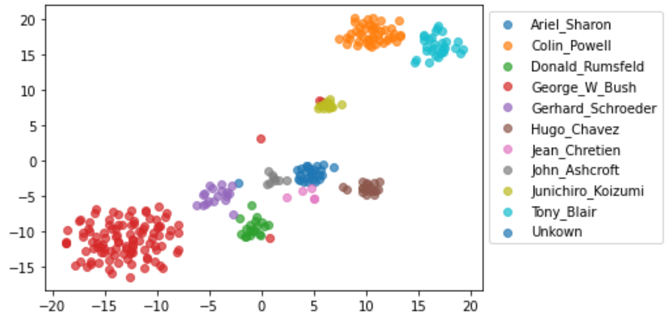
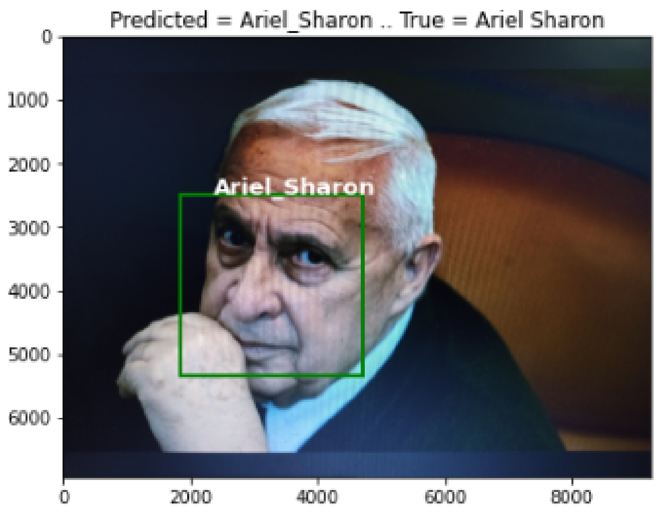
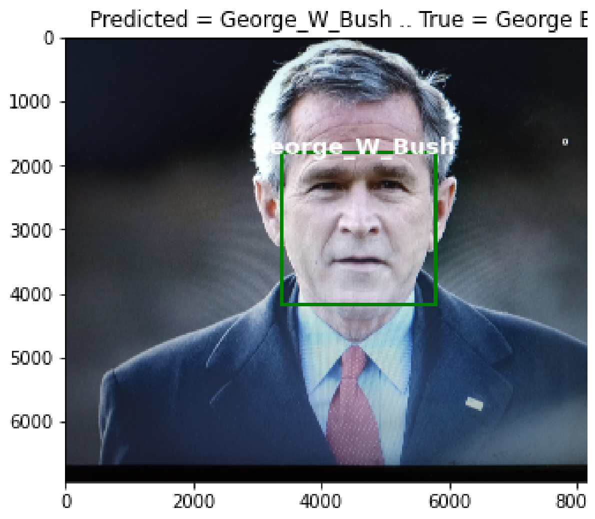
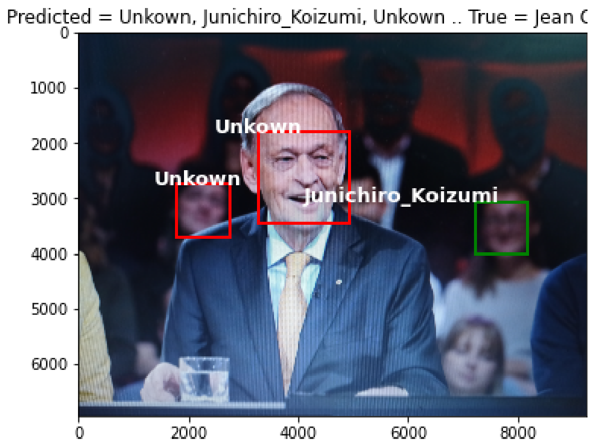


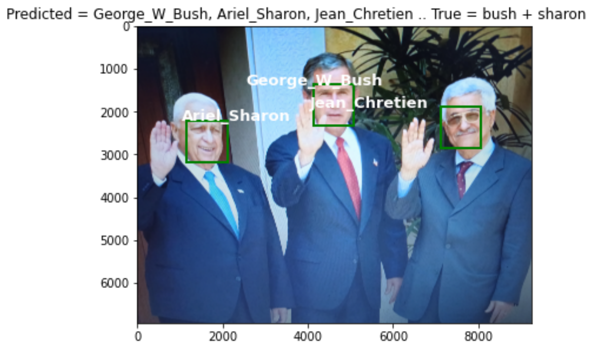
Fig. 10. Result of visualization of test set (Unknown faces are tested)

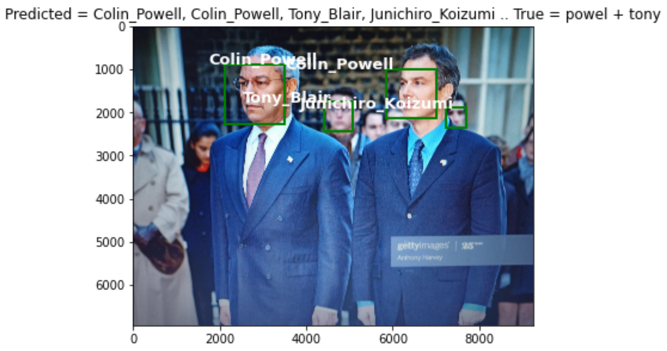
6. Examples of predictions are shown in figure 11. Pictures are taken from mobile to visualize real-life security cameras. The model also compares the prediction to the actual value and detects and recognizes many faces in the picture. Unknown faces are detected, but the model does not recognize them.






Predicted = Unkown, Junichiro_Koizumi, Unkown .. True = Jean C
Unkown
Unkown
Junichiro_Koizumi


Predicted = George_W_Bush, Ariel_Sharon, Jean_Chretien .. True = bush + sharon
George_W_Bush
Jean_Chretien
Ariel_Sharon


Predicted = Colin_Powell, Colin_Powell, Tony_Blair, Junichiro_Koizumi .. True = powel + tony
Colin_Powell
Colin_Powell
Tony_Blair
Junichiro_Koizumi

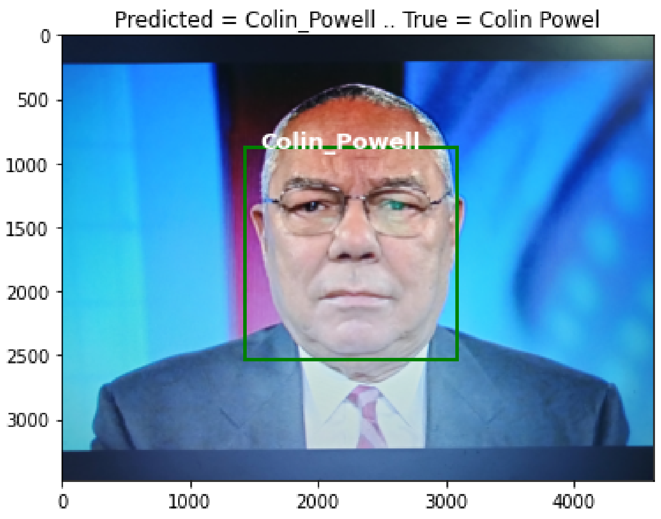


Fig. 11. Result of the test set

As shown in Figure 11, the model is correct. predicted most of the individual faces while in the multi-faces pictures, the model predicted some of They are right and one or two wrong.

III. Conclusion

To sum up, our face recognition systems have an accuracy of 97.26% and can detect and recognize many faces in one picture. Moreover, the system can control access in a given login system.

IV. STRENGTHS

1. The model can detect and align faces.
2. The model can get the embedding vectors and calculate the triplet loss distances to compare pictures for the same person and a different person.
3. The model can visualize and group similar faces if we do not have any labels or names.

4. The model can detect many faces in one picture.
5. The model can detect and recognize pictures from the mobile phone camera.
6. The model can give permission or access to genuine people.

V. Weaknesses

1. The state of the art accuracy in this model is above 99%, but our accuracy is 97.26%
2. The inference time is long and does not fit real-time applications like security camera checking.

VI. Possible improvements:

1. When we tried to use transfer learning to adjust the pre-trained model on a subset (1000 images) of our dataset, it took hours to train, and we assumed it would take days to train on the whole dataset. With some higher specs or more free time, we can train the model on the dataset and save the weights once and for all.
2. Try it in a real camera and get successive frames as input pictures.
3. Train the model on images of various brightness levels for every person.
4. Train the model on the very recent images of the person as we noticed that most of its mistakes were on people that have young and old images.
5. Get the average embeddings for each class and use them instead of the embeddings of all images when calculating the distance of a test image, which should significantly decrease the inference time.

VII. Acknowledgment

Special thanks to Dr. Sayed Hemayed and Dr. Mohamed Elshenawy for their helpful feedback in the demo and the first draft of the report.

VIII. References

[1] Eng, D., & Hong, J. (2015). Convolutional Neural Networks for Visual Recognition Course (Cs 231n). Convolutional Neural Networks for Visual Recognition Course (CS 231n).

[2] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 815–823, 2015.

[3] Li, S. Z., & Jain, A. K. (2011). Handbook of face recognition. London: Springer.

[4] B. Amos, B. Ludwiczuk, J. Harkes, P. Pillai, K. Elgazzar, and M. Satyanarayanan. OpenFace: Face Recognition with Deep Neural Networks

[5] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. Procedings of the British Machine Vision Conference 2015. doi: 10.5244/c.29.41

[6] Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. 2014 IEEE Conference on Computer Vision and Pattern Recognition. doi: 10.1109/cvpr.2014.220

[7] Oravec Miloš. (2010). Face recognition. Vienna: In-Tech.