Face Verification
Final Project – CSCE 4604

By Abdullah Kasem and Rafik Yacoub

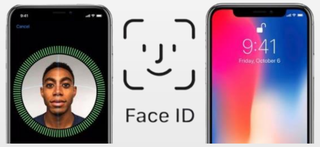
Under the Supervision of Prof. Moustafa Youssef and Eng. Sherif Moustafa

School of Science and Engineering

Computer Science and Engineering Department

Problem Statement

- **The Face verification** is the process of **extracting facial features** from an image and then **comparing** them to the facial features of another image to **verify** the identity of the person in the image.
  Real-life example: **face ID feature** of cell phones.

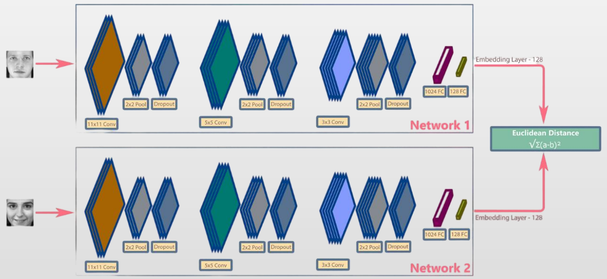


Dataset Description

- Name: **Extracted Faces, face-recognition-dataset**, derived from the **LFW Dataset**.
- Size: 190 MB
- Description: Each picture is **centered** on a **single face** and encoded in **RGB** and of size **128x128x3**. **1680 celebrity** directories, with **2-50 images** per directory.
- Faces extracted from the original image using **Haar-Cascade Classifier (cv2)**.

Original chosen model

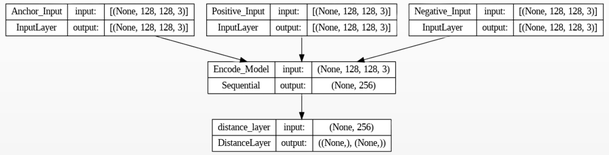
- **Siamese Network:** It **does not classify** the images into categories, rather, it finds out the **distance** between any two given images. If the images have the **same label**, then the network should **learn the parameters**, in such a way that it should produce a **smaller distance** between the two images, and if they belong to **different** labels, then the **distance** should be **larger.**



- **Encoder**: is responsible for **converting** the images into their **feature vectors**.
- A pre-trained model, **Xception** is used.
- It is connected to **Dense layers** and the last layer uses **L2 Normalization**.

How it works

- The **Siamese Network** takes 3 input images (**anchor, positive, negative**) and uses the **encoder** to get the **feature vectors**.
- Those features are passed to a **distance layer** which computes the **distance between (anchor, positive) and (anchor, negative) pairs.**

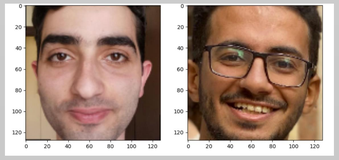


- After training (using the **triplet loss function**), **Encoder** is **extracted** and given two pictures to get their **feature vectors** and calculate the **distance** between them.
- Given a **threshold**, it's determined **whether** the images are the **same** person or a **different** person.

Updates to the model

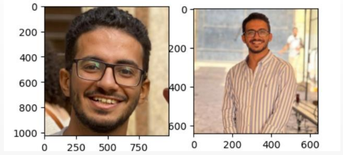
- Data is split into correct and fixed sets and triplets.
- Adjusted the number of frozen layers in the Xception model to 35 layers.
- Data augmentation was used to compensate for the small dataset and as a way of regularization.
- Shuffled the training data.
- Added function to do face extraction and alignment.
- Added more evaluation metrics besides accuracy.
- Live face verification.
- Hyperparameter tuning:
  - Tanh instead of Relu
  - Max pooling instead of Average
  - Batch sizes 64, 128, 256
  - Augmentation percentages of 30%, 50%, 60%, 100%

Input / output example



```
Distance: [1.7238022]
prediction is Different
```

Input Output



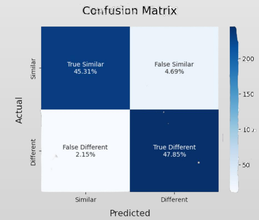
```
Distance: [0.44238418]
prediction is Same Person
```

Input Output

Initial Results Vs Final Results

- **Initial Accuracy: 84.375%**

- **Final Accuracy: 93.164%**
- **Recall: 95.7%**
- **Precision: 91.1%**
- **F1 score: 93.3%**
- **AUC: 93.16%**
- **Categorical Cross-entropy: 2.46**



Conclusion

- Learned to work with limited resources (colab + small dataset) by using the pre-trained model and data augmentation.
- One-Shot Learning.
- Model is biased.
- Results are still far from SOTA or product.

Acknowledgements

We would like to thank Prof. Moustafa Youssef and Eng. Sherif Mostafa for their continuous support.