Face Matching Project Report

**Executive Summary:**

This report presents the findings and outcomes of a Face Matching Project that leveraged various models available in the DeepFace library and Siamese network with triplet loss to achieve accurate face recognition. Among the models evaluated, FaceNet emerged as the champion model due to its exceptional performance in face matching. The project aimed to develop a robust face-matching system with applications in identity verification, security, and access control.

1. **Introduction**

Face matching is a critical technology in numerous domains, including identity verification, law enforcement, access control, and personalized services. This project aimed to assess and select the most effective face-matching model among several available in the DeepFace library and Siamese with triplet loss.

1. **Dataset**

The project used a diverse dataset of facial images, containing a wide range of lighting conditions, poses, and expressions.

1. **Methodology**

Face matching is a complex task with numerous approaches and models. In our pursuit of an accurate and robust face matching system, I adopted a comprehensive methodology that involved the use of DeepFace and Siamese with triplet loss. Our objective was to identify the most suitable model for achieving high precision in matching faces from different images.

* 1. **DeepFace**

DeepFace is the most lightweight face recognition and facial attribute analysis library for Python. The open-sourced DeepFace library includes all leading-edge AI models for face recognition and automatically handles all procedures for facial recognition in the background.

* **DeepFace has set of features:**
* **Face Verification:** The task of face verification refers to comparing a face with another to verify if it is a match or not. Hence, face verification is commonly used to compare a candidate’s face to another. This can be used to confirm that a physical face matches the one in an ID document.
* **Face Recognition:** The task refers to finding a face in an image database. Performing face recognition requires running face verification many times.
* **Facial Attribute Analysis:** The task of facial attribute analysis refers to describing the visual properties of face images. Accordingly, facial attributes analysis is used to extract attributes such as age, gender classification, emotion analysis, or race/ethnicity prediction.
* **Real-Time Face Analysis:** This feature includes testing face recognition and facial attribute analysis with the real-time video feed of your webcam.
* **Implementation**
* **Requirement installing libraries**

!pip install deepface

!pip install opencv-python

!pip install matplotlib

* **Training phase**

Several models from DeepFace are used which are VGG-Face, Facenet, Facenet512, OpenFace, DeepFace and, ArcFace. For each model different distances metrices are used which are cosine, Euclidean, Euclidean -l2.

1. **FaceNet:** FaceNet, developed by Google researchers, introduced the concept of using a siamese network architecture for face recognition. It learns a mapping from face images to a high-dimensional feature space where faces of the same identity are close together and faces of different identities are far apart. FaceNet achieved remarkable performance on benchmark face recognition datasets, such as Labeled Faces in the Wild (LFW) and MegaFace.
2. **VGGFace:** VGGFace is a deep CNN model based on the VGGNet architecture. It is trained on a large-scale dataset of faces and has demonstrated strong performance in face recognition tasks. VGGFace has been widely used and benchmarked on various face recognition datasets.
3. **DeepFace:** DeepFace is a deep learning model developed by Facebook AI Research. It utilizes a deep CNN architecture trained on a large-scale dataset to learn discriminative features for face matching. DeepFace achieved impressive performance on the LFW dataset and introduced a novel alignment technique to improve accuracy.
4. **ArcFace:** ArcFace is a face recognition model that incorporates an additive angular margin loss into the softmax loss function. This loss function enhances the discriminative power of the learned face embeddings. ArcFace has shown state-of-the-art performance on face recognition benchmarks like LFW, MegaFace, and IJB-C.

* **Evaluation phase**

1. **Accuracy**: Accuracy measures the proportion of correctly matched faces out of the total number of face pairs. It is calculated as the number of true positive matches divided by the sum of true positive and false negative matches.
2. **False Acceptance Rate (FAR):** FAR measures the percentage of incorrect matches where the system falsely accepts a non-matching pair as a match. It is calculated as the number of false positive matches divided by the sum of false positive and true negative matches.
3. **False Rejection Rate (FRR):** FRR measures the percentage of incorrect matches where the system falsely rejects a matching pair. It is calculated as the number of false negative matches divided by the sum of false negative and true positive matches.
4. **Equal Error Rate (EER):** EER is the point on the ROC curve where the FAR and FRR are equal. It represents the threshold at which the system achieves an equal balance between false acceptances and false rejections. A lower EER indicates better performance.
5. **Precision and Recall:** Precision measures the proportion of correctly matched face pairs out of the total number of pairs identified as matches. Recall, also known as sensitivity or true positive rate, measures the proportion of correctly matched face pairs out of the total number of actual matches. These metrics are useful for analyzing the system's performance at different similarity thresholds.
6. **F1 Score:** The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances precision and recall, allowing for a comprehensive evaluation of the system's performance.

A graph of different colored lines

Description automatically generated A graph of different colored bars

Description automatically generated

A graph of different colored lines

Description automatically generated with medium confidence A graph of different colored bars

Description automatically generated

A graph of different colored bars

Description automatically generated A graph of different colored bars

Description automatically generated

A graph of different colored bars

Description automatically generated

* 1. **Siamese with triplet loss**
* **Siamese Network**

Siamese networks are popular in image recognition tasks such as face verification, facial recognition, and signature verifications. In these tasks, Siamese networks compare two images and determine if they are similar or dissimilar. One of the key advantages of Siamese networks is their ability to work with small datasets

In the Siamese network, we take two images and pass them through the same network to get the feature embedding for the corresponding image. After we have the feature embeddings, we compare the feature embeddings, if they are from the same, we should have similar embeddings. If we have images from two different classes, then the embeddings should be far away.

* **Triplet loss**

The basic idea behind triplet loss is to learn embeddings for each input, such that the distance between embeddings for similar inputs is minimized, while the distance between embeddings for dissimilar inputs is maximized. This is typically achieved by defining triplets of inputs, consisting of an anchor input, a positive input (i.e., a similar input to the anchor class), and a negative input (i.e., a dissimilar input to the anchor class).

The loss function calculates the squared Euclidean distance between the embeddings of the anchor and positive inputs and subtracts it from the squared Euclidean distance between the embeddings of the anchor and negative inputs. The resulting difference is then compared to the margin hyperparameter. If the difference is less than the margin, the loss function returns zero, otherwise, it returns the difference between the two distances. The margin hyperparameter controls the degree of separation between the embeddings of the anchor and negative inputs.

* **Implementation**

**- Training phase**

Firstly, data preprocessing was performed to standardize image sizes and pixel values, and to enhance data quality. I convert the image to grey scale and resized it. After that, I build Siamese network using convolution neural network and fully connection layer, then build a triplet loss function. During training, the model was optimized using Adam optimizer with learning rate 0.0001

- **Evaluation phase**

I used the same evaluation that used in DeepFace

**4. Result:**

Multiple face-matching models available in the DeepFace library were evaluated using a rigorous evaluation process. The evaluation criteria included accuracy, precision, recall, F1 score, and efficiency. Each model was trained and tested on the dataset to assess its suitability for the face-matching task. FaceNet, a deep convolutional neural network model, consistently outperformed other models in all evaluation metrics with accuracy 99.6% and equal error rate 1%. It demonstrated exceptional accuracy and robustness, making it the champion model for this project.

**5. Discussion:**

The project's evaluation process revealed that FaceNet is the most suitable model for face-matching applications, offering exceptional accuracy and real-time performance. Its deep architecture, trained on a large-scale dataset, proved highly effective.

**6. Conclusion:**

In conclusion, this Face Matching Project successfully identified FaceNet as the champion model for face matching. The project provides a solid foundation for implementing FaceNet in real-world scenarios, such as identity verification, access control, and security applications.