# Face Matching Project

Koren Abitbul

## 1 Introduction

The primary objective of the project is to develop a system that can identify all the images in a database where a specific person appears, given a reference face image of that individual. For example, consider a company that hosts an event and captures numerous photographs during the occasion. The goal is to enable the company to efficiently identify and provide the relevant images to each employee, under the assumption that the company has access to the employees' phone numbers and face images.

## 2 Data Preprocessing

### 2.1 Directory with Images from an Event

Data preprocessing of the dataset involved the following steps:

### 2.2 Step 1: Face Extraction

Utilized YOLO-V8 with the following CLI command to extract faces from all the images:

```
yolo task=detect mode=predict model=yolov8m-face.pt conf=0.85 imgsz=1280
line_width=1 max_det=1000 source=event_images save_crop=True
```

Saved the extracted face images in a different directory.

### 2.3 Step 2: Face Embedding with VGG-Face Model

Employed the VGG-Face model to generate embeddings from the extracted face images. The model was trained to determine the probability that two face embeddings are from the same person. Created clusters of persons' faces using this model.

### 2.4 Step 3: Manual Verification of Clusters

Verified and corrected the clusters manually, acknowledging the VGG-Face model's less than 100% accuracy. more details in Figure 1.

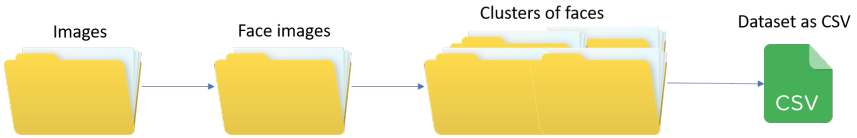


Figure 1: The data processing stages

# 3 Creating the Dataset

After generating the clusters, it is possible to construct the dataset. The dataset comprises negative and positive samples, with each sample being an 8-length vector that includes the distances between two face images. Figure 2 illustrate it. These distances are computed using cosine-distance and Euclidean-distance metrics from four distinct face detection models. The dataset is automatically very big. An event with 500 images easily contained 2000 faces and generating samples from a pair of faces will be approximately 2,000,000 samples. Most of the samples are negative (around 97%), yet we keep the dataset unbalanced since the real-world data is also unbalanced.

# 4 Training the Model

Since our dataset is a simple tabular dataset where each sample is composed of 8 features, we pick a simple ML model – the XGboost. We employed the 'xgboost' library for our model, utilizing the squared error logistic function as the objective function. However, this choice posed a challenge in our context. While maximizing accuracy is generally desirable, our specific scenario demanded a different approach. Misclassifying two images as a 'match' when they do not belong to the same person is worse than classifying a genuine match as 'not a match.' In our case, the consequence of sending images of a different individual to a client outweighs the cost of omitting a valid image.

There are two approaches to address the issue:

1. Opting for a different objective function that prioritizes precision over accuracy.
2. Instead of using argmax for the classification, we can determine the prediction of the model as 'match' only if the certainty is higher than some threshold (higher than 0.5 obviously).

We choose option 2 since there is no built-in such objective function in the library, and creating a custom objective function would involve defining its gradient and Hessian, which could complicate matters unnecessarily, and we want to keep it simple. To identify the optimal threshold, we needed a metric other than accuracy. We chose $F_\beta$ as our metric of choice, defined as:

$$F_\beta = \frac{(1+\beta^2) \cdot (Precision \cdot Recall)}{\beta^2 \cdot Precision + Recall}$$

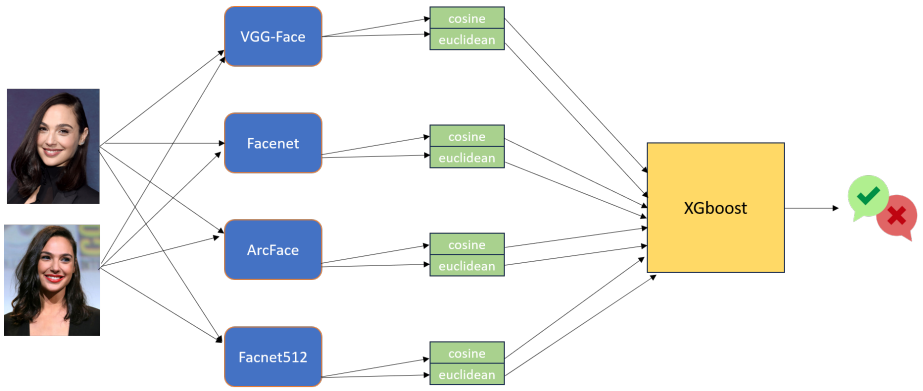


Figure 2: The ensemble method

After experimenting with different values, we settled on $\beta = 0.15$, finding it well-suited to our case based on theoretical precision and recall values. Following a grid search on the validation set, we determined that a threshold of 0.94 yielded promising results:

- Precision: 99.16%
- Recall: 77.68%