Team 0

# Facial Recognition

Trịnh Minh Tâm - 11215226
Phan Anh Khôi - 11212902
Hoàng Thị Tú Anh - 11219255
Kiều Thanh Tâm - 11219286
Nguyễn Thành Long - 11213549

Project Face Recognition

# Table of Contents

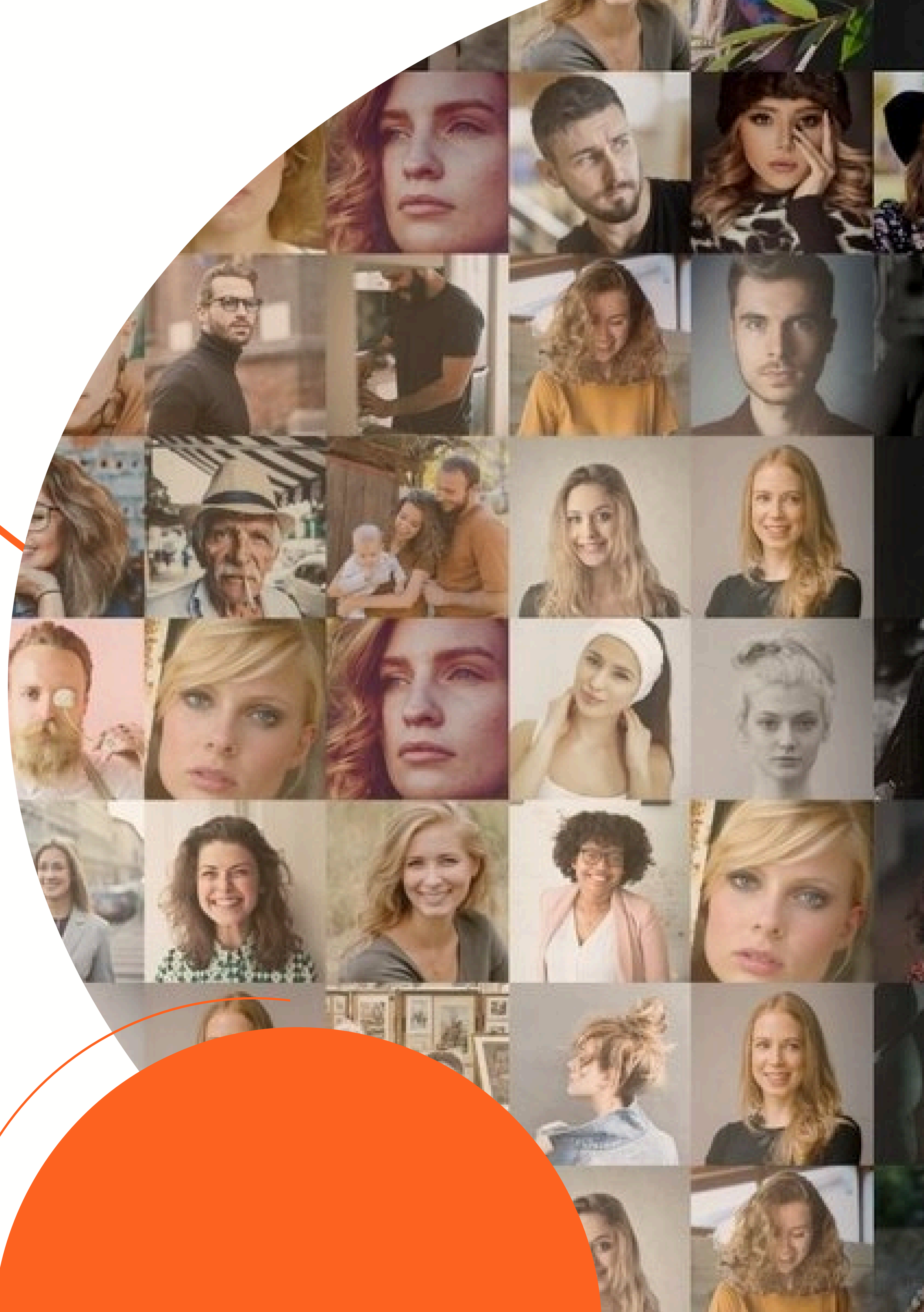
Face Recognition

# INTRODUCTION

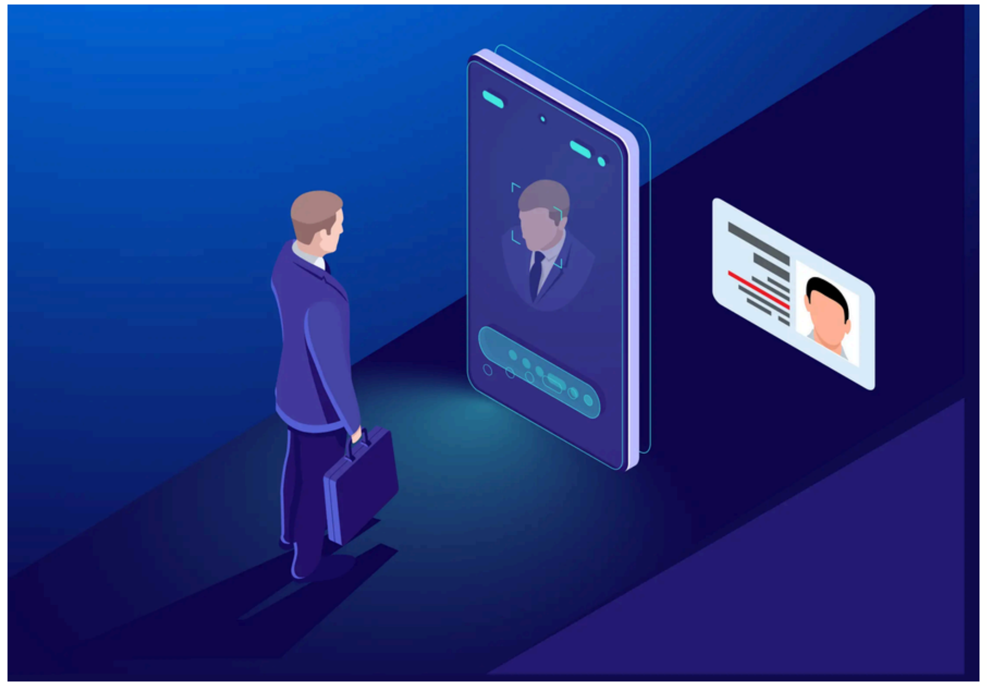
Face Recognition

# Definition

Face recognition is a computer vision task that involves identifying and verifying faces  by analyzing and comparing unique facial features captured in digital images and videos.



# Motivation



Attendance System

Law Enforcement & Surveillance


Smart Cards

A facial recognition project not only opens doors for innovative applications but also showcases the power of technology in addressing real-world challenges, creating modern and secure user experiences.

# Problem setting: Task 1 and 2

**Dataset AT&T:** There are 10 different images of each of 40 distinct subjects.


**Image Size: 92x112x1**

Images are various in lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses / no glasses) in dark background with upright, frontal position.

**The goal** is to implement a facial recognition system with a dual approach:

- classifying images to associate them with specific individuals
- identifying the closest match by comparing faces to a pre-existing database.

# Problem setting: Task 3

**Dataset Labelled Faces in the Wild (LFW):**

- **13,233 images** with **5749 people.**
- Various in **pose**, **lighting conditions**, **facial expressions**, **occlusions**, and **image quality.**

Due to **class imbalance**, we filtered data:

- **1387 images** with **100 people**.
- Each class has **10-20 images**

**The goal** is to do experiment with:

- different preprocessing methods data.
- different types of PCA for dimensional reduction and previous approaches.

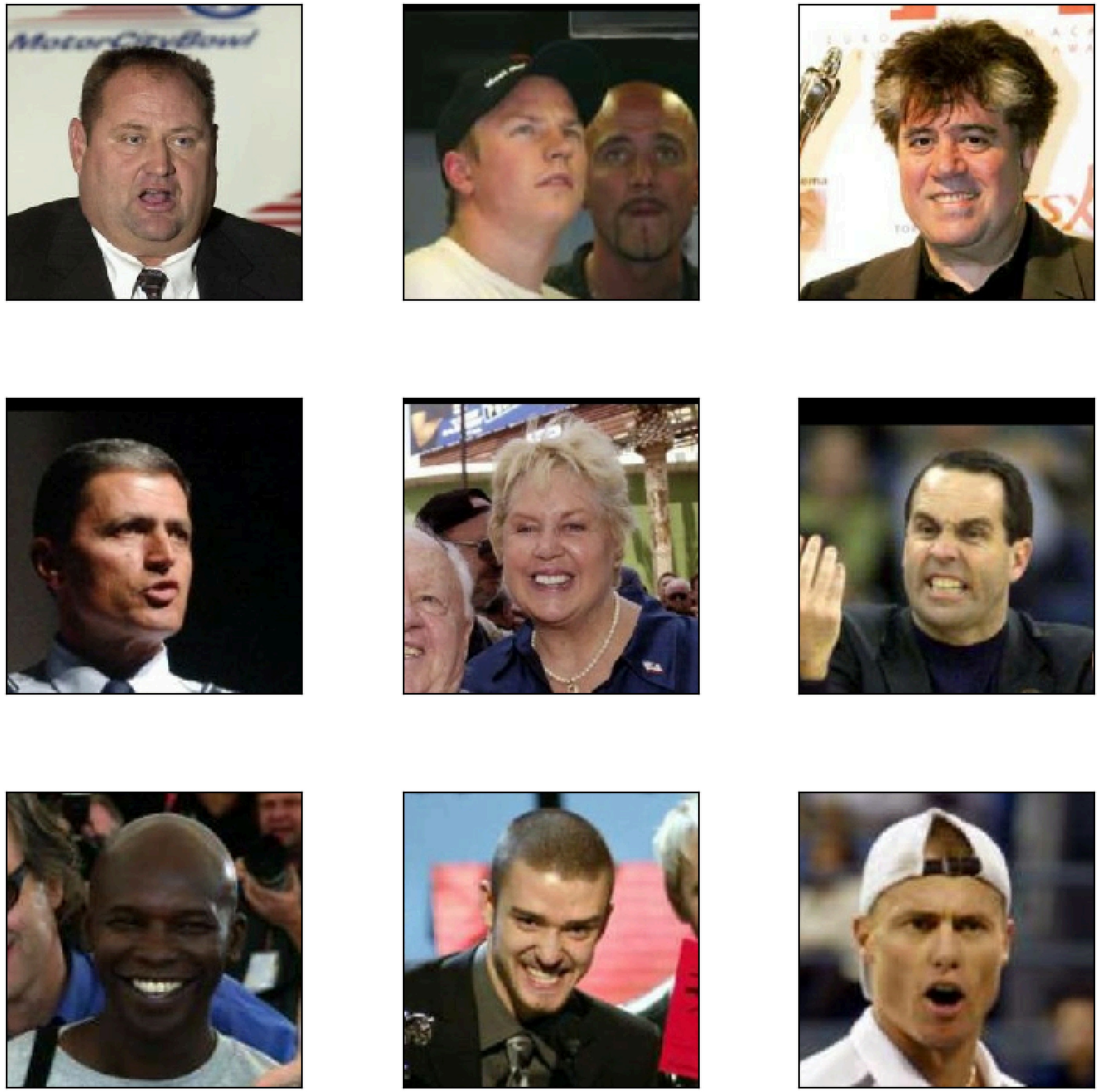


**Image Size: 250x250x3**

# Labelled Faces in the Wild (LFW)

# LITERATURE REVIEW

Face Recognition

# LITERATURE REVIEW

There are many state-of-the-art approaches have been introduced for face recognition. D.S. Trigueros et al. (2018) introduced Convolutional neural networks (CNNs) for this problem. CNNs are the most common type of deep learning method for face recognition.

Advantages:

- Can be trained with large amounts of data to learn a face representation that is robust to the variations present in the training data => Robust to different types of intra-class variations (e.g. illumination, pose, facial expression, age, etc.)

Drawbacks:

- Need to be trained with very large datasets that contain enough variations to generalize to unseen samples.

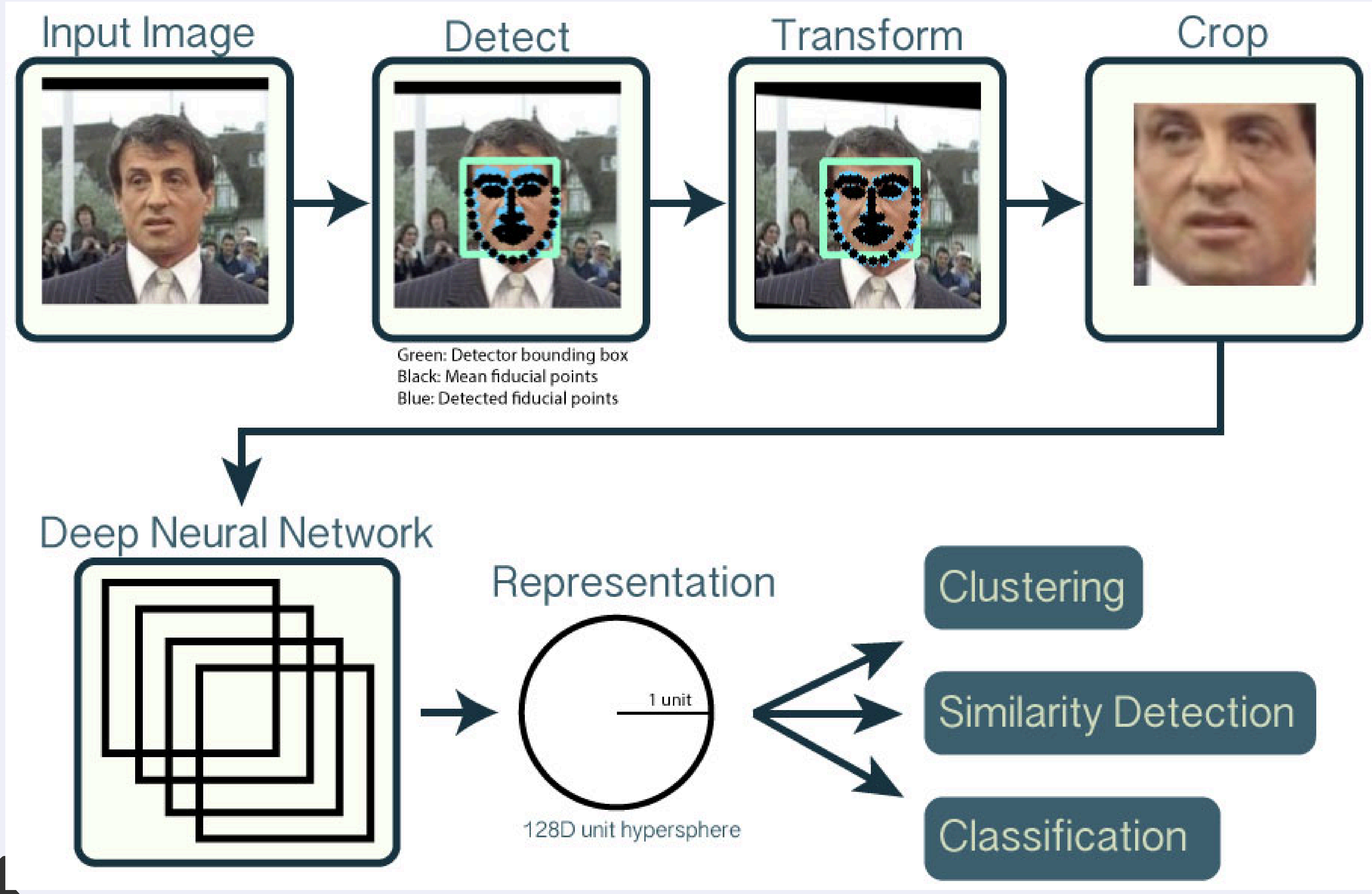



Figure: OpenFace

# Deep Neural Network(CNN)



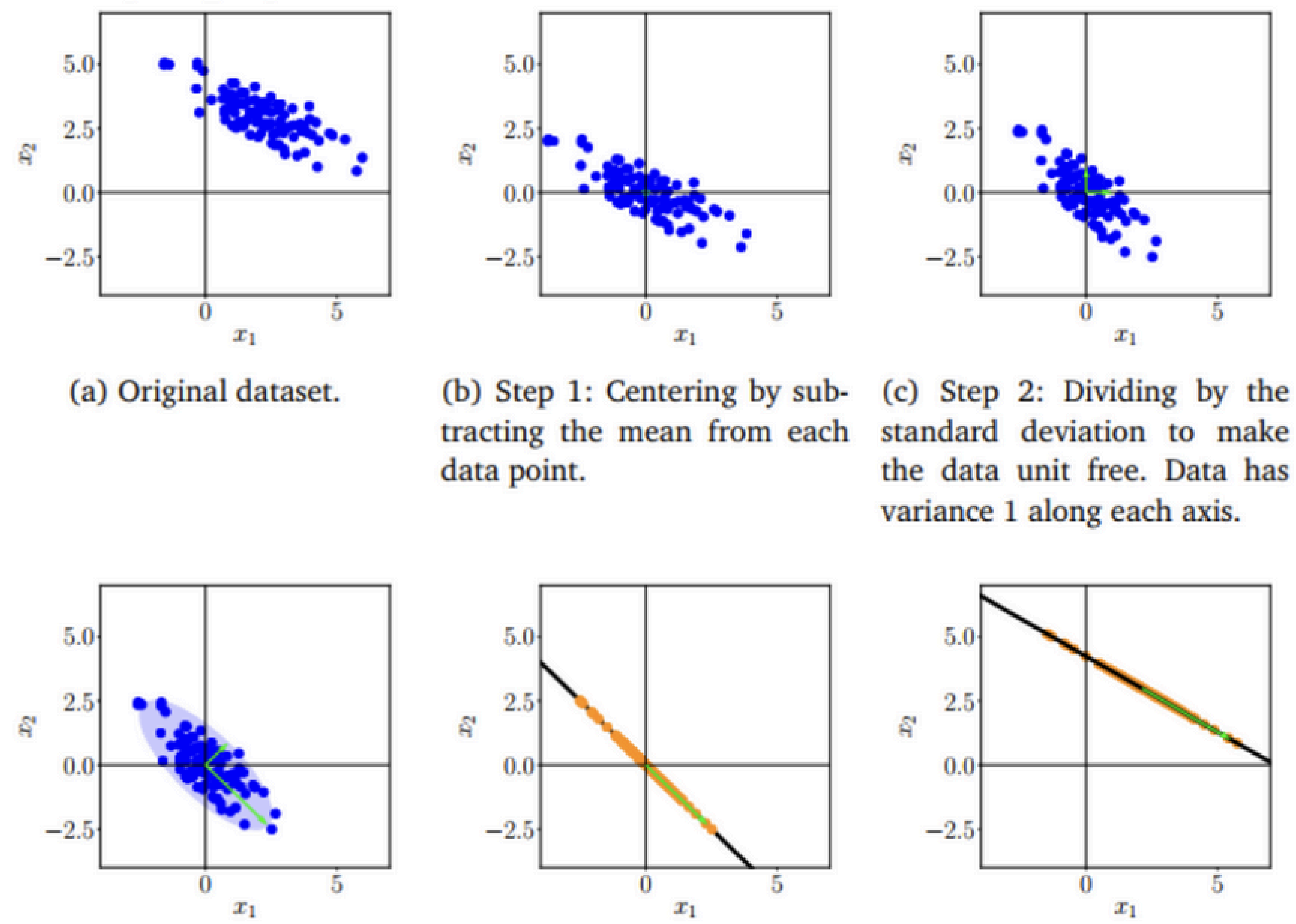
**→ We decided to do experiment with traditional approaches by using PCA with distance measurement methods and machine learning models in this project**

- Muge Carikci et. al. (2012) used PCA and Eigenfaces Method for face recognition problem and archived an 94.74% accuracy using Euclidean distance.
- Vytautas Perlibakas (2003) compared 14 different distance measures and their modifications with respect to the recognition performance of the principal component analysis (PCA)-based face recognition method and propose modified sum square error (SSE)-based distance.
- Omar Faruque et. al. (2009) shows that Polynomial and Radial Basis Function (RBF) SVMs performs better than Linear SVM on the ORL Face Dataset and the SVMs are a better learning algorithm than the standard eigenface approach using Multi-Layer Perceptron (MLP) Classification criterion for face recognition.

# Principal Component Analysis (PCA)

PCA is a dimensional reduction technique used in statistics and machine learning to transform high-dimensional data into a lower-dimensional representation, preserving the most important information

## How does PCA works?




(a) Original dataset.

(b) Step 1: Centering by subtracting the mean from each data point.

(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.

(d) Step 3: Compute eigenvalues and eigenvectors (arrows) of the data covariance matrix (ellipse).

(e) Step 4: Project data onto the principal subspace.

(f) Undo the standardization and move projected data back into the original data space from (a).

## Advantages

1. Dimensional Reduction
2. Feature Selection

## Disadvantages

1. Linearity Assumption
2. Sensitivity to Outliers

# Kernel Principal Component Analysis (KPCA)

- **Definition:** Kernel Principal Component Analysis (KPCA) is a technique used in machine learning for nonlinear dimensionality reduction

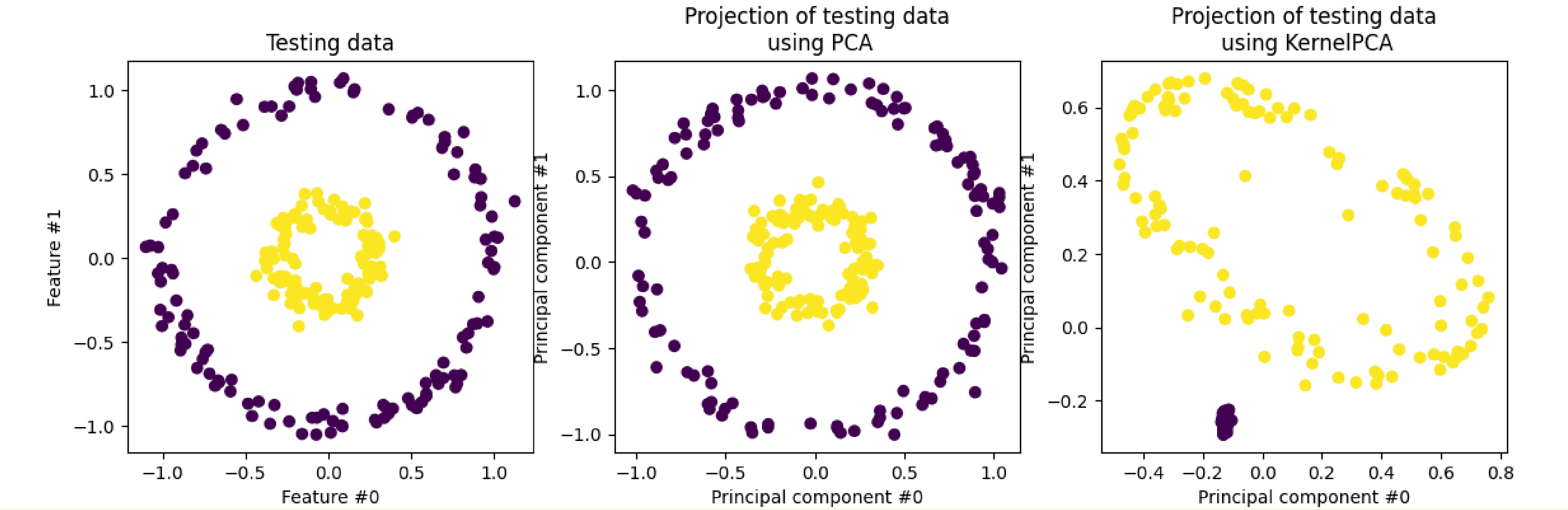
- KPCA applies a nonlinear mapping function to the data before applying PCA, allowing it to capture more complex and nonlinear data

**Advantages of KPCA:**

- Deal with non-linear data

**Disadvantages of KPCA:**

- Complexity
- Overfitting



# Randomized PCA

## Definition

- Randomized PCA is a method used to approximate the principal components of a dataset, rather than computing them exactly as in traditional PCA. This reduces the computational burden, especially for large datasets.

- Randomized PCA uses a random projection matrix to map the data to a lower-dimensional subspace.

## Advantages of Randomized PCA

- Scalability
- Memory Efficiency

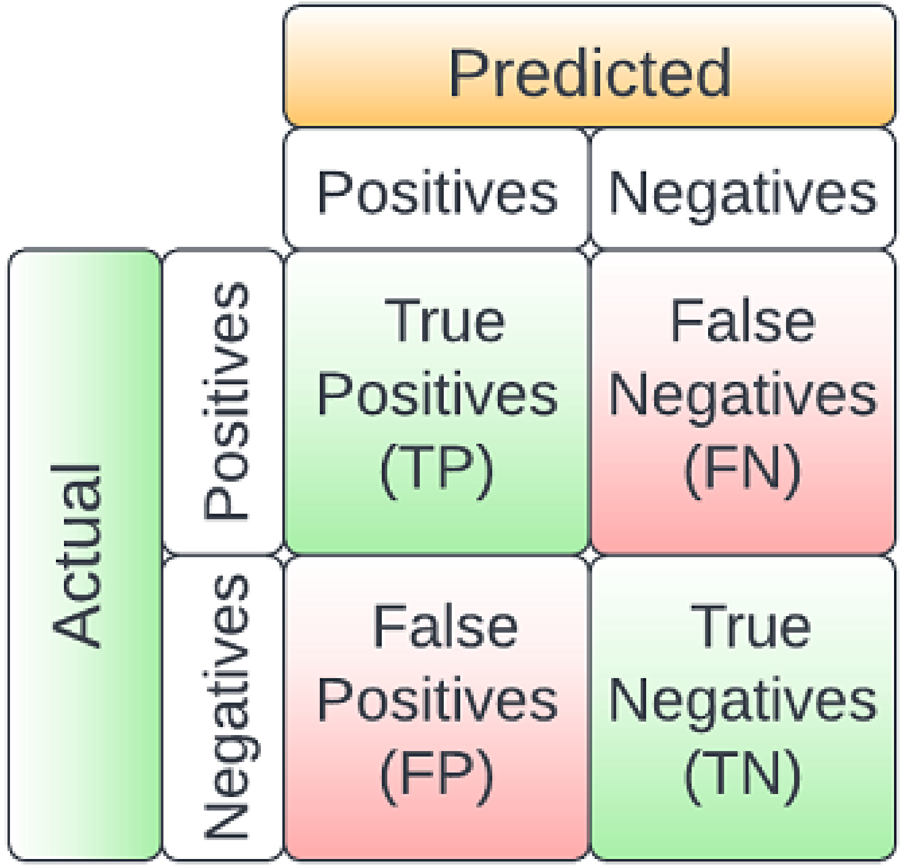
## Disadvantages of Randomized PCA

- Accuracy Trade-off
- Less Interpretability

# SOLUTION + RESULT

Face Recognition

# Evaluation Metrics




**Accuracy**

**Precision**

**Recall**

**F1-score**

## Why we use this 4 metrics:

- Assess the system's ability while considering both false positives and false negatives.
- Ensure not only a comprehensive evaluation but also specific insights such as positive prediction accuracy or sensitivity

# Evaluation Metrics

- **Accuracy:** measures the overall correctness of the model by quantifying the number of correct predictions made over the total number of predictions. Accuracy answers the question *"Out of all the predictions we made, how many were true?"*

$$accuracy = \frac{true\ positives + true\ negatives}{true\ positives + true\ negatives + false\ negatives + false\ positives}$$

**=> Accuracy provides a general idea of how well a recognition system is performing.**

- **Precision:** refers to the proportion of correct positive matches out of all the positive matches identified by the system. It answers the question *"Out of all the positive predictions we made, how many were true?"*

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

**=> High precision means that when system identifies someone, it is very likely to be correct.**

# Evaluation Metrics

- **Recall:** refer to the proportion of actual positive matches that were correctly identified by the system. It answers the question *"Out of all the data points that should be predicted as true, how many did we correctly predict as true?"*

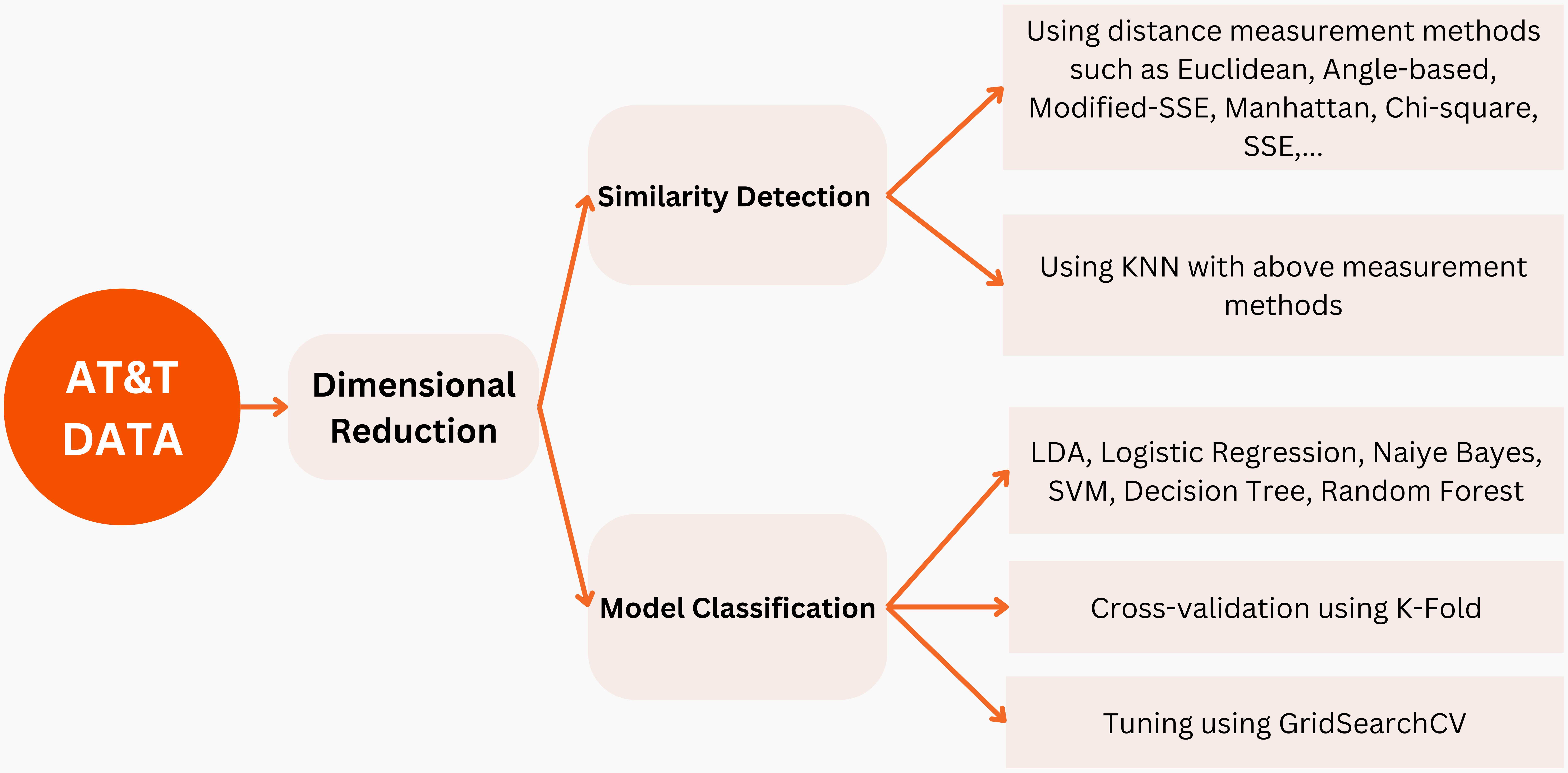
$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

**=> Recall focuses on minimizing false negatives, ensuring that the model doesn't miss any cases.**

- **F1-score:** is the harmonic mean of precision and recall.

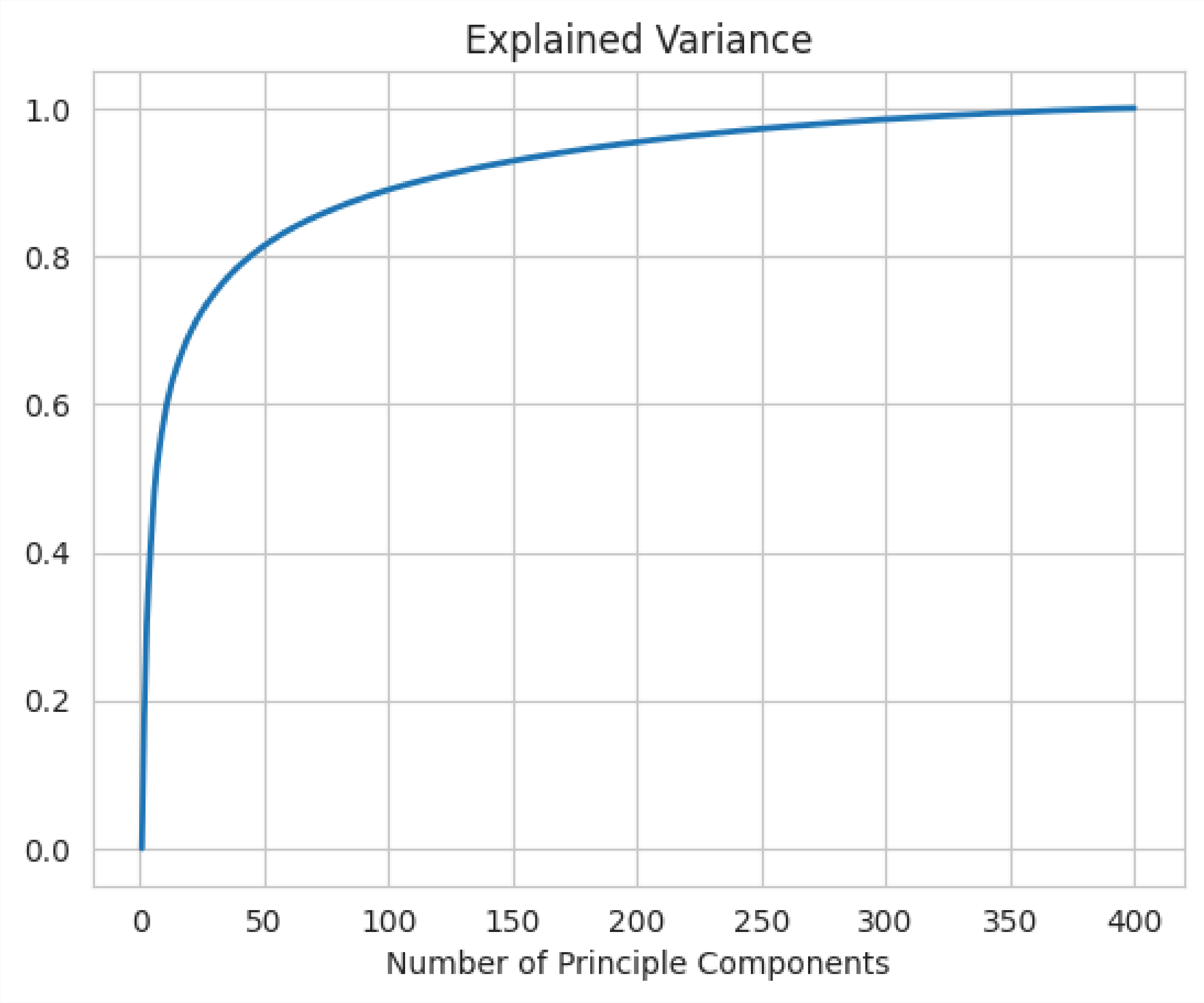
$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

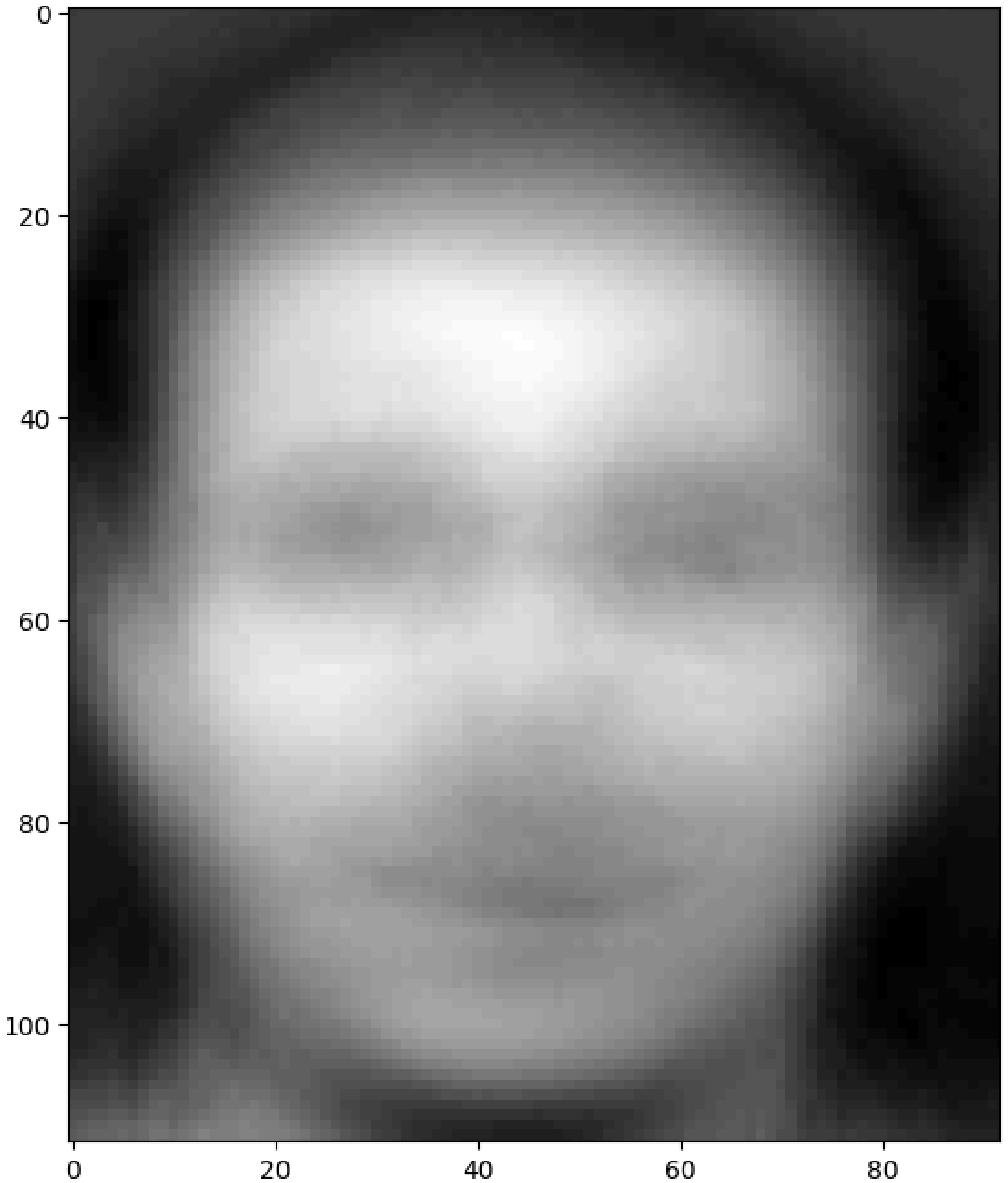
**=> F1-score considers both false positives and false negatives, making it a more reliable indicator of a model's performance**


AT&T DATA
Dimensional Reduction
Similarity Detection
Model Classification
Using distance measurement methods such as Euclidean, Angle-based, Modified-SSE, Manhattan, Chi-square, SSE,...
Using KNN with above measurement methods
LDA, Logistic Regression, Naiye Bayes, SVM, Decision Tree, Random Forest
Cross-validation using K-Fold
Tuning using GridSearchCV

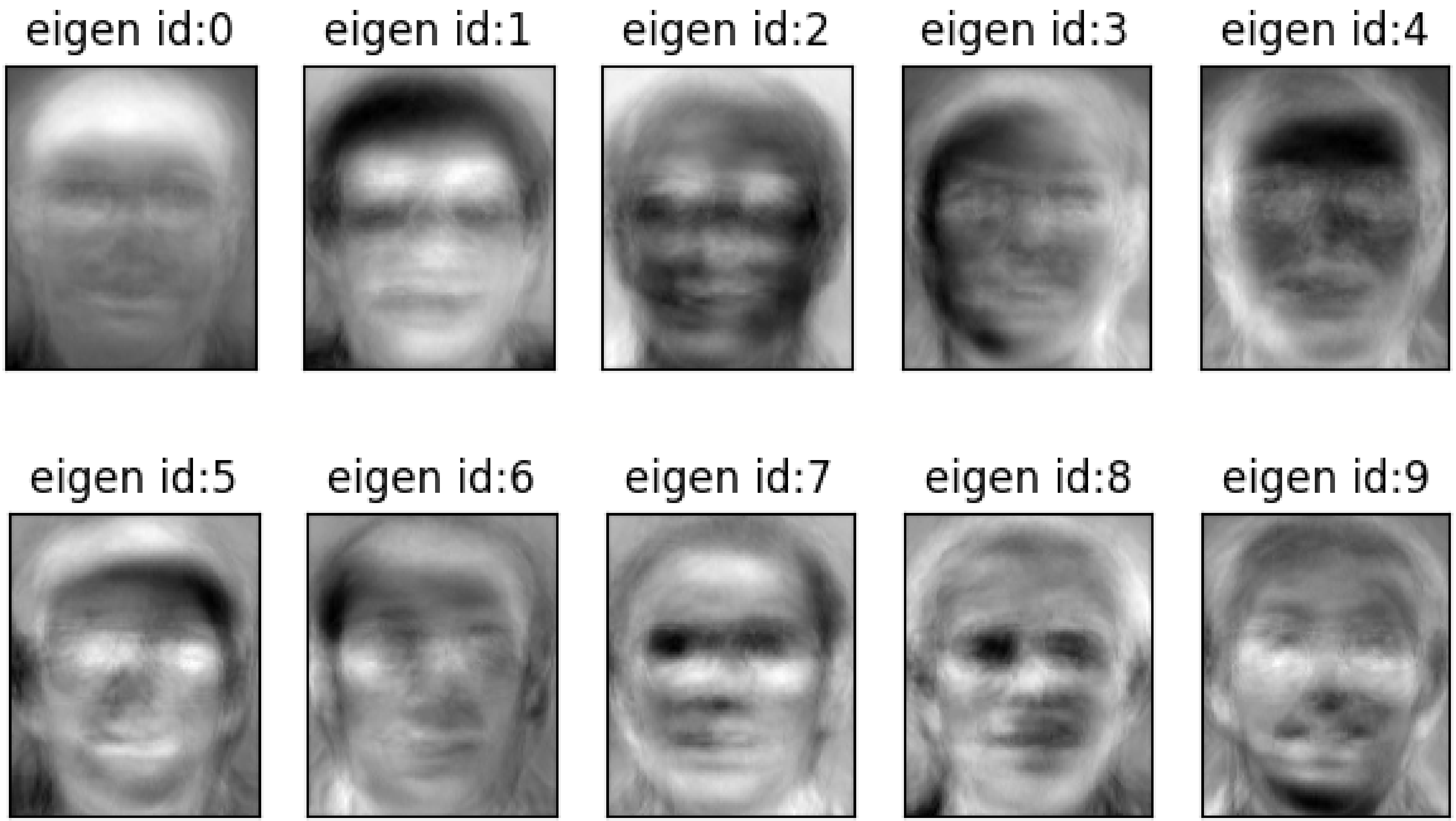
# Dimensional Reduction using PCA

In the figure, it can be seen that 50 PCA components represent 80% of the data. Now let's make the classification process using 50 PCA components






Average face




Ghost faces

The average face and ten eigenfaces with the highest eigenvalues (ghost faces) are calculated using the training set. It can be seen that the sharpness of the ghost faces is relatively low, which shows that the facial expressions and the lighting conditions are quite varying.

# Distance measures

To match the input images with individual folders, we calculate the distance between the eigenface of the image and the eigenfaces stored previously. The person in question is identified as the one whose distance is minimum in the eigenface database.

- Euclidean distance
- Angle-based distance (cosine similarity)
- Modified SSE distance
- Mahalanobis distance
- Manhattan distance (city block distance)
- SSE distance
- Chi-square distance
- Correlation coefficient-based distance
- Minkowski distance

# Distance measures: results

- Manhattan distance achieved the highest accuracy of 99.17%, while Chi-square distance yielded the lowest accuracy of 2.50%.
- Most distances improved with higher n_components, except Mahalanobis distance, which performed better with smaller n_components but worse as n_components increased.

| | **Accuracy** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|---|
| **Manhattan distance** | 99.17 | 99.38 | 99.17 | 99.14 |
| **Euclidean distance** | 95.83 | 96.67 | 95.83 | 95.74 |
| **Mahalanobis distance** | 91.67 | 93.83 | 91.67 | 91.4 |

# Distance measures: results

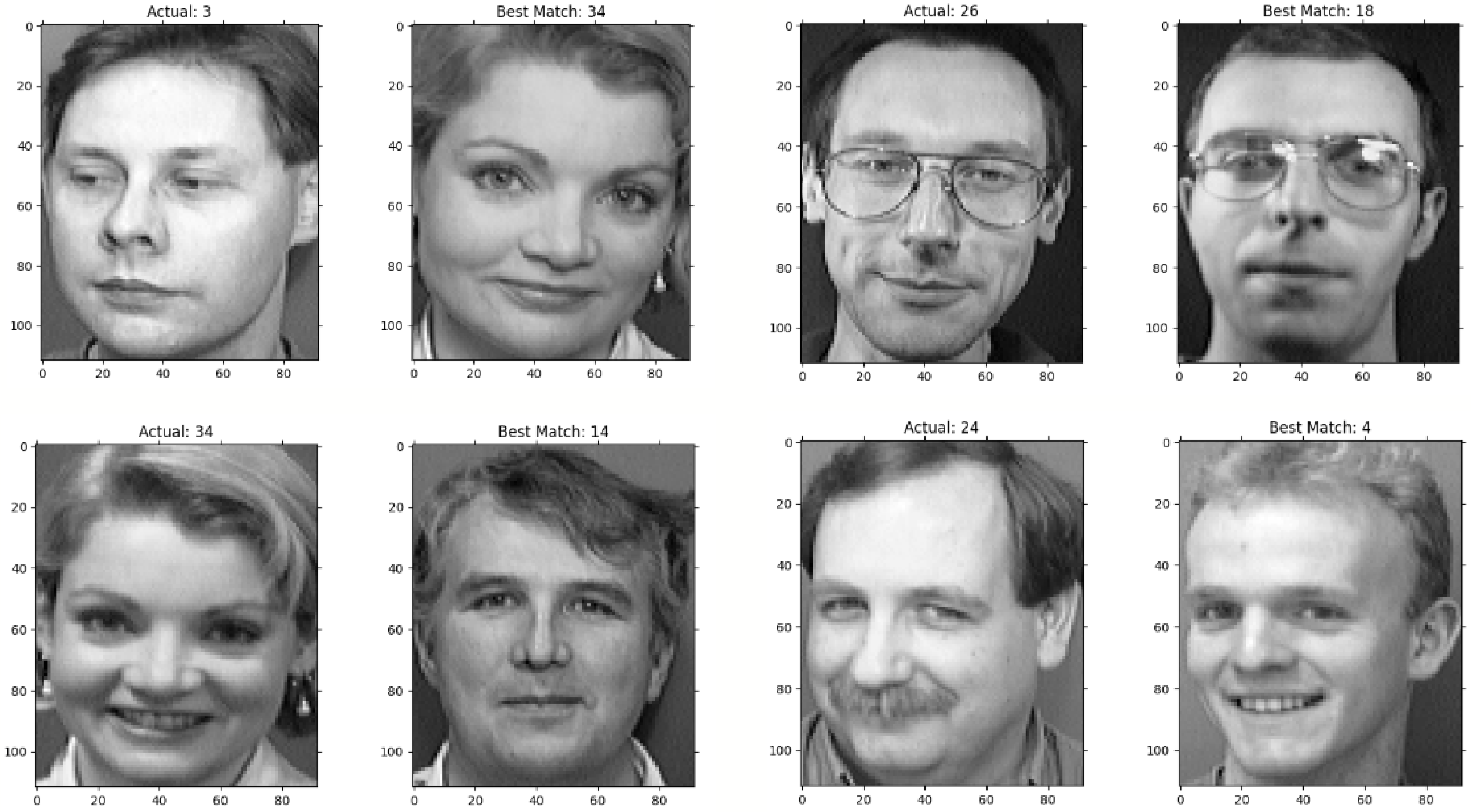
## KNN with top 3

- When using KNN, the model's performance has decreased more significantly.
- There might be various influencing factors, but one possibility is the way the data is split

| | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| **Manhattan distance** | 95.00 | 96.50 | 95.00 | 94.70 |
| **Euclidean distance** | 89.17 | 90.29 | 89.17 | 88.06 |
| **Mahalanobis distance** | 90.83 | 94.03 | 90.83 | 90.64 |

# Distance measures: results

## The most frequent incorrect cases are:




- The main challenges for distance measures in facial recognition are shooting angle, facial expressions, facial features (beard, hair, ...), glasses and illumination.
- Angle-based, modified-SSE, Correlation coefficient-based, SSE and Minkowski distance have many things in common in wrong cases, with SSE and Minkowski recognizing beard and hair better than the remaining distances.

# Classification models: results

## Decision Tree, Random Forest, GaussianNB, SVM, Logistic Regression, LDA

- LDA and Logistics Regression gave the best results with accuracy of 99.17% and 98.33% respectively (1 case and 2 cases wrong).
- Decision Tree is the only model with accuracy less than 90%.
- GaussianNB, Logistic Regression and LDA have very similar wrong cases.
- Except for Decision Tree, all models give better performance or no change when tuning with GridsearchCV.
- According to the cross-validation scores Linear Discriminant Analysis and Logistic Regression still have best performance

# Classification models: results

## Decision Tree, Random Forest, GaussianNB, SVM, Logistic Regression, LDA

**Model**

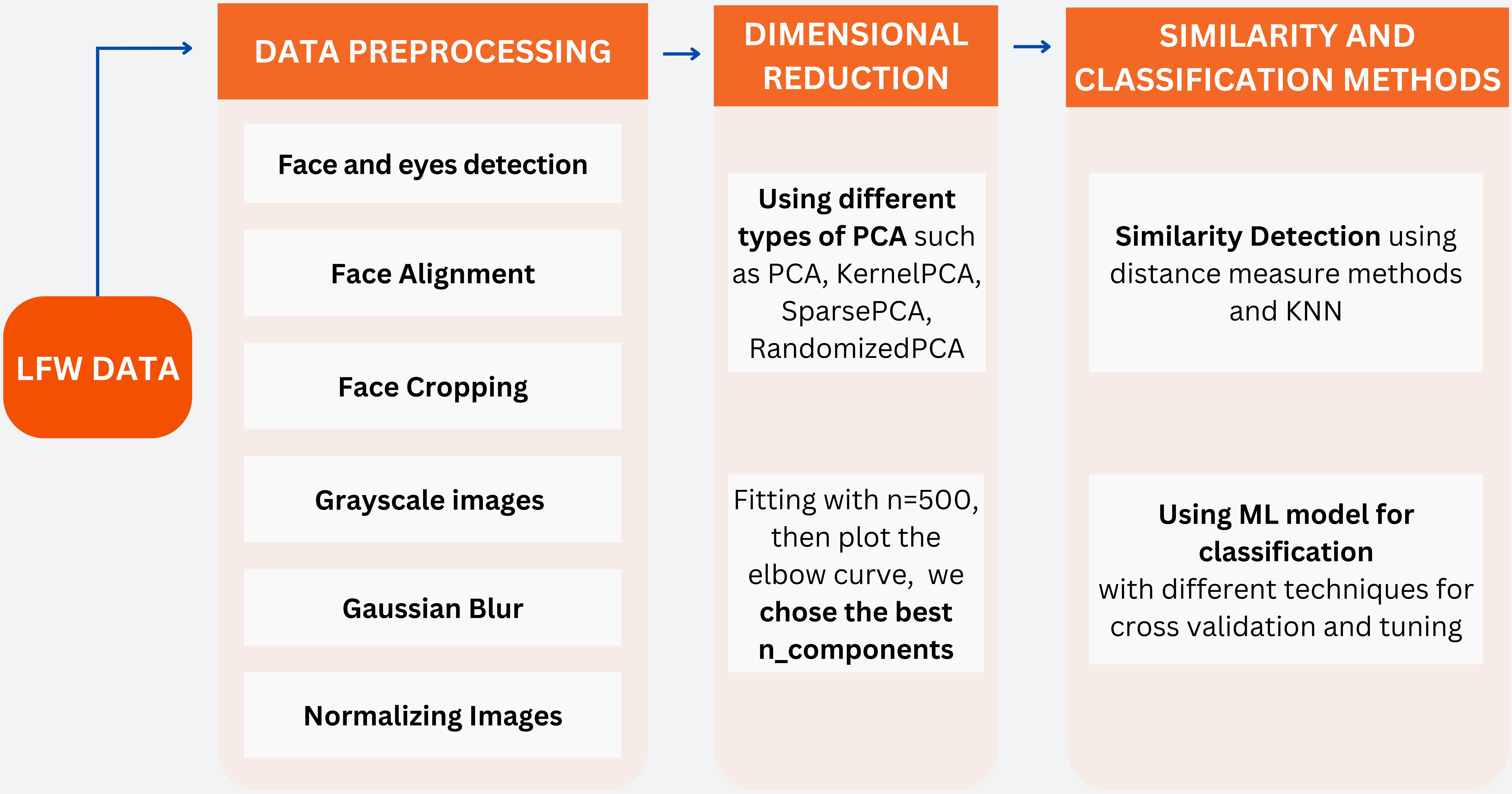
| | Accuracy | F1-score | Precision | Recall |
|---|---|---|---|---|
| LDA | 99.17 | 99.14 | 99.38 | 99.17 |
| Random Forest | 91.67 | 91.49 | 93.42 | 91.67 |
| SVM | 85 | 83.15 | 88.5 | 85 |
| Decision Tree | 65.83 | 64.07 | 68.38 | 65.83 |

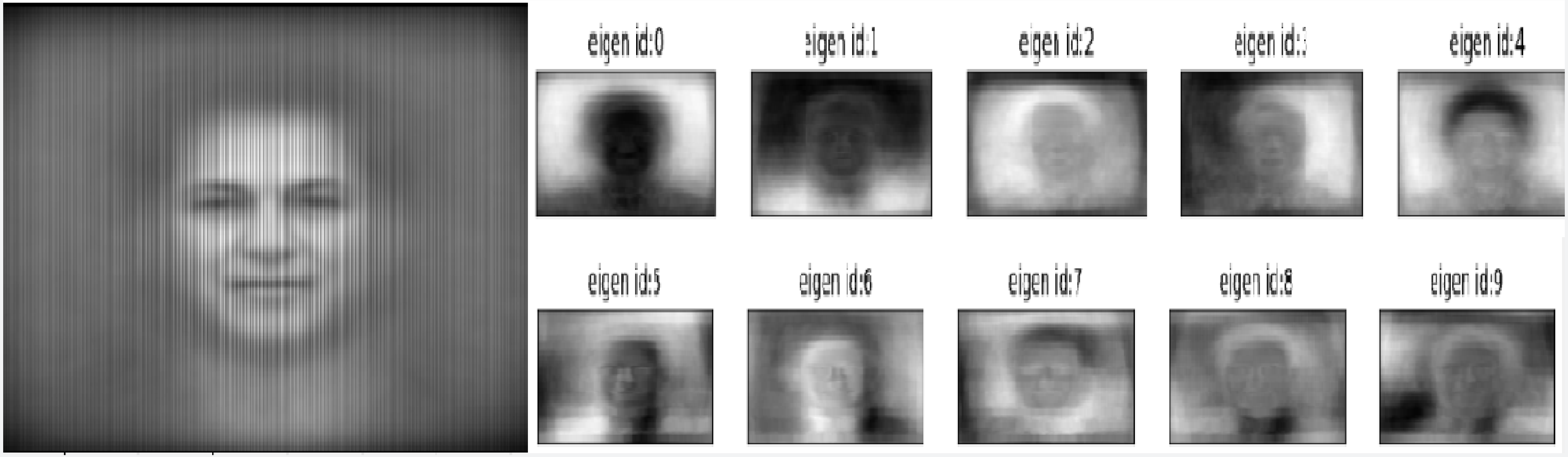
**Cross validation using K-fold**

| | Accuracy | F1-score | Precision | Recall |
|---|---|---|---|---|
| LDA | 99.17 | 99.14 | 99.38 | 99.17 |
| Random Forest | 94.17 | 91.49 | 93.42 | 91.67 |
| SVM | 97.5 | 97.43 | 98.12 | 97.5 |
| Decision Tree | 64.17 | 62.31 | 64.98 | 64.17 |

**Tuning with GridSearchCV**

| | Accuracy | F1-score | Precision | Recall |
|---|---|---|---|---|
| LDA | 99.17 | 99.14 | 99.38 | 99.17 |
| Random Forest | 95.83 | 95.8 | 97.12 | 95.83 |
| SVM | 97.5 | 97.43 | 98.12 | 97.5 |
| Decision Tree | 56.67 | 55.75 | 64.2 | 56.67 |


LFW DATA
DATA PREPROCESSING
Face and eyes detection
Face Alignment
Face Cropping
Grayscale images
Gaussian Blur
Normalizing Images
DIMENSIONAL REDUCTION
Using different types of PCA such as PCA, KernelPCA, SparsePCA, RandomizedPCA
Fitting with n=500, then plot the elbow curve, we chose the best n_components
SIMILARITY AND CLASSIFICATION METHODS
Similarity Detection using distance measure methods and KNN
Using ML model for classification with different techniques for cross validation and tuning




The average face and ghost faces of the LFW dataset. It can be seen that the sharpness of the ghost faces is very low, which shows that the background, the facial expressions, and the lighting conditions are more diverse than in the AT&T dataset.

# Data Preprocessing

## Face Detection by Haar Feature-based Cascade Classifier (Viola-Jones Algorithms)

### HAAR FEATURE SELECTION

Calculating Haar features by summing the pixel intensities in each region and calculating the differences between the sums

### ADABOOST TRAINING

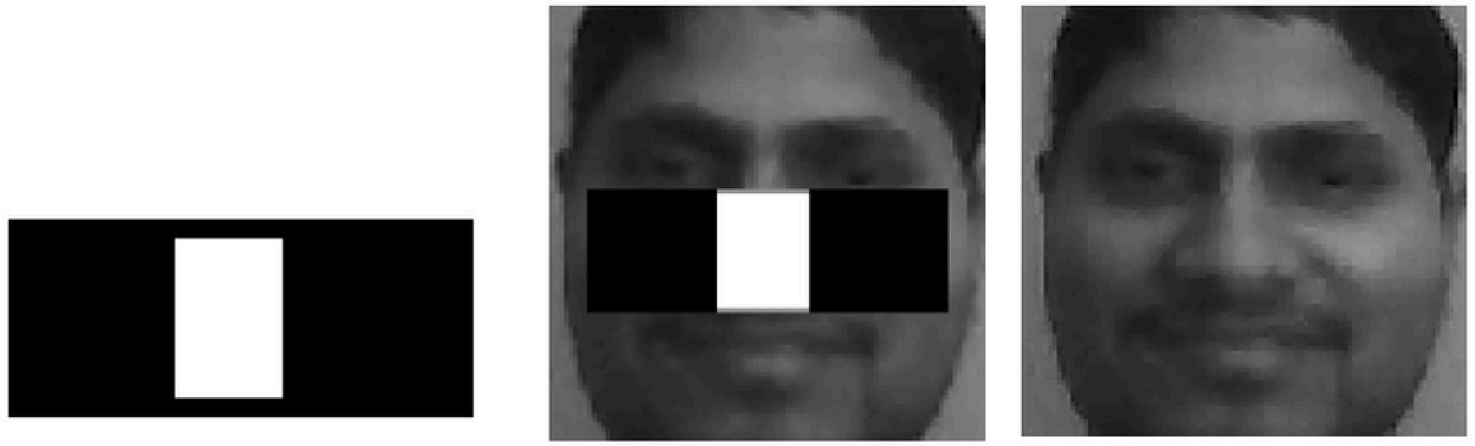
Adaboost essentially chooses the best features and trains the classifiers to use them

### CASCADING CLASSIFIERS

Distinguish whether a window contains a face or not.

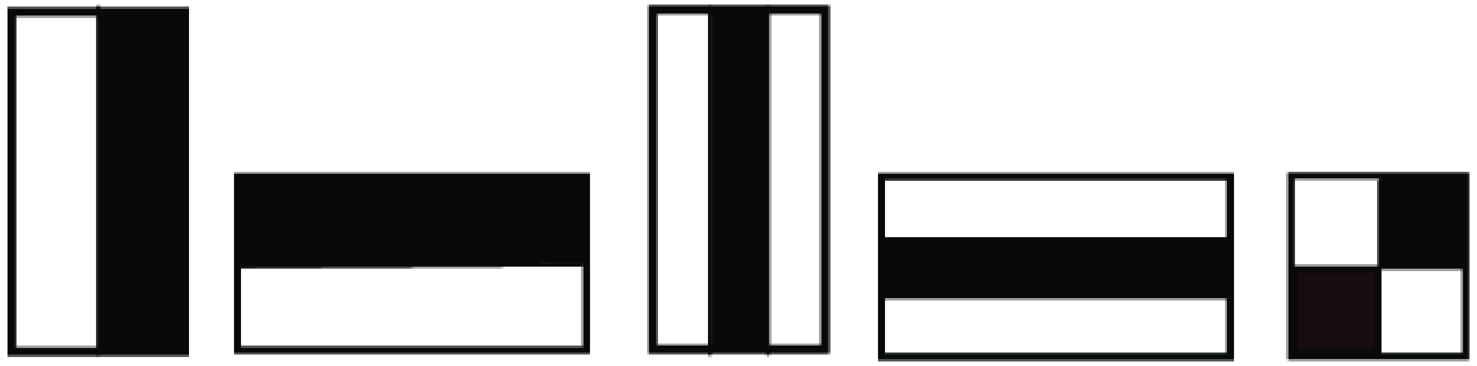
# Haar-like features

- Typical human faces share some similarities. For example, the eye region is darker than the upper cheeks, while the nose bridge region is brighter than the eyes.
- We can determine some features of the images such as lines and edges by comparing these pixel readings to readings in an adjacent area. To do this, we use what are called Haar-like filters, which are ideal clusters of pixels that could represent a feature.



This Haar-like feature identifies the nose

This Haar-like feature identifies the eye region

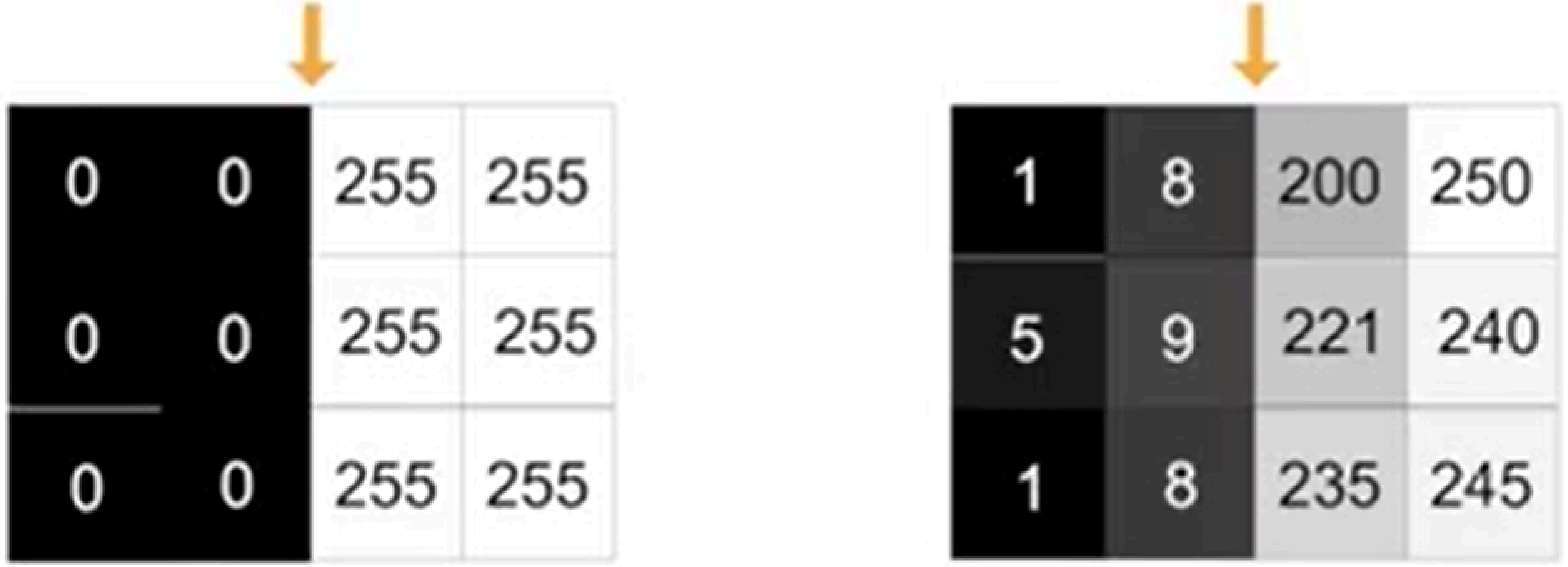


Types of Haar-like features

## Original Image

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10 | 5 | 25 | 82 | 42 | 13 | 63 | 46 |
| 88 | 64 | 35 | 39 | 66 | 21 | 15 | 44 |
| 2 | 45 | 78 | 24 | 42 | 54 | 90 | 62 |
| 4 | 12 | 46 | 73 | 36 | 25 | 85 | 46 |
| 35 | 66 | 3 | 56 | 42 | 73 | 85 | 25 |
| 56 | 98 | 24 | 1 | 50 | 53 | 42 | 14 |
| 78 | 46 | 62 | 85 | 53 | 27 | 26 | 58 |
| 34 | 52 | 26 | 52 | 29 | 47 | 24 | 34 |

Sum = 419
(computation of
8 numbers)




**Ideal**: There is <u>definitely</u> an edge here

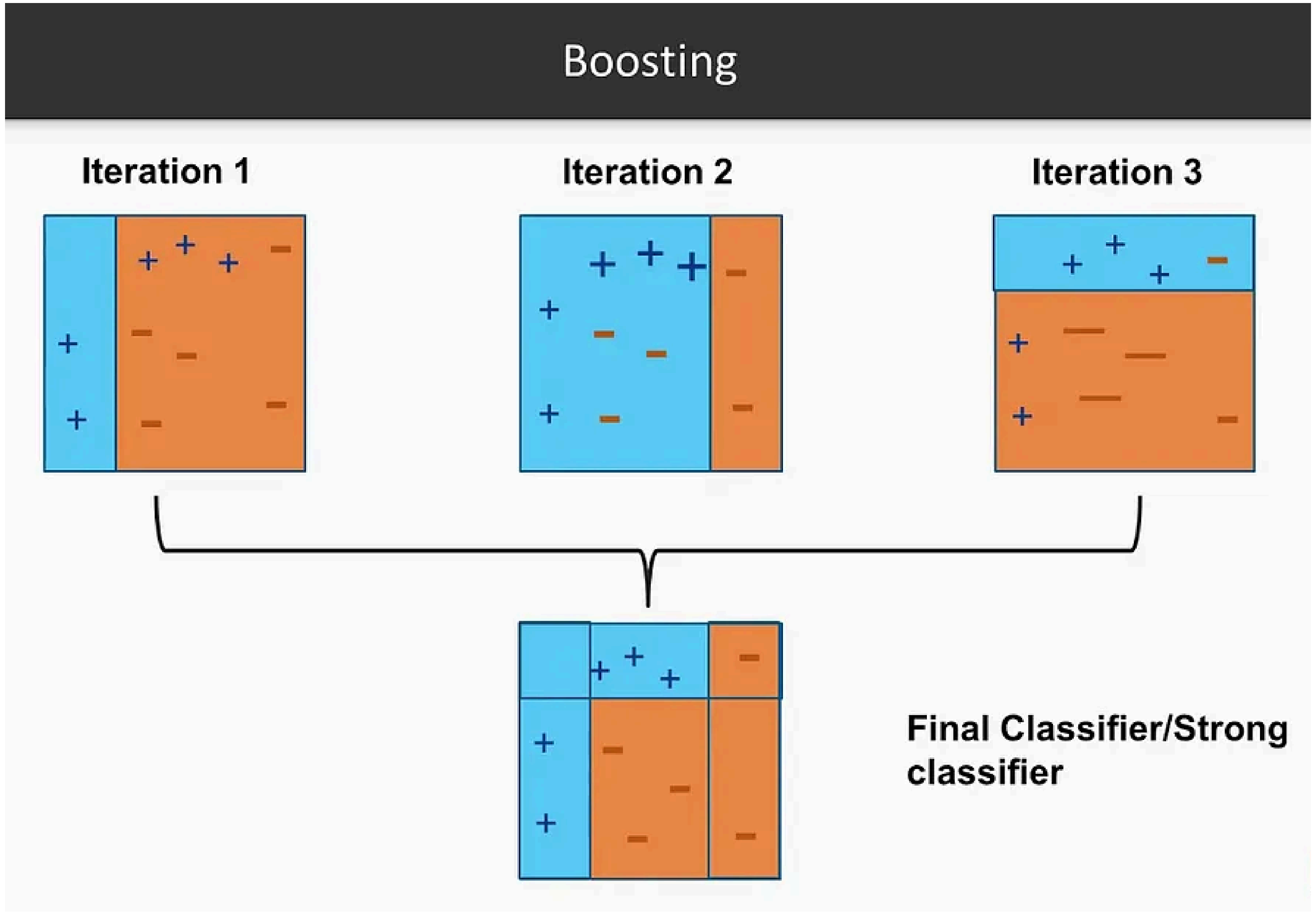
**Realistic**: There is <u>probably</u> an edge here

**Response to Filter $H_A$ at location $(i,j)$:**

$$V_A[i,j] = \sum \text{(pixel intensities in white area)}$$
$$- \sum \text{(pixels intensities in black area)}$$

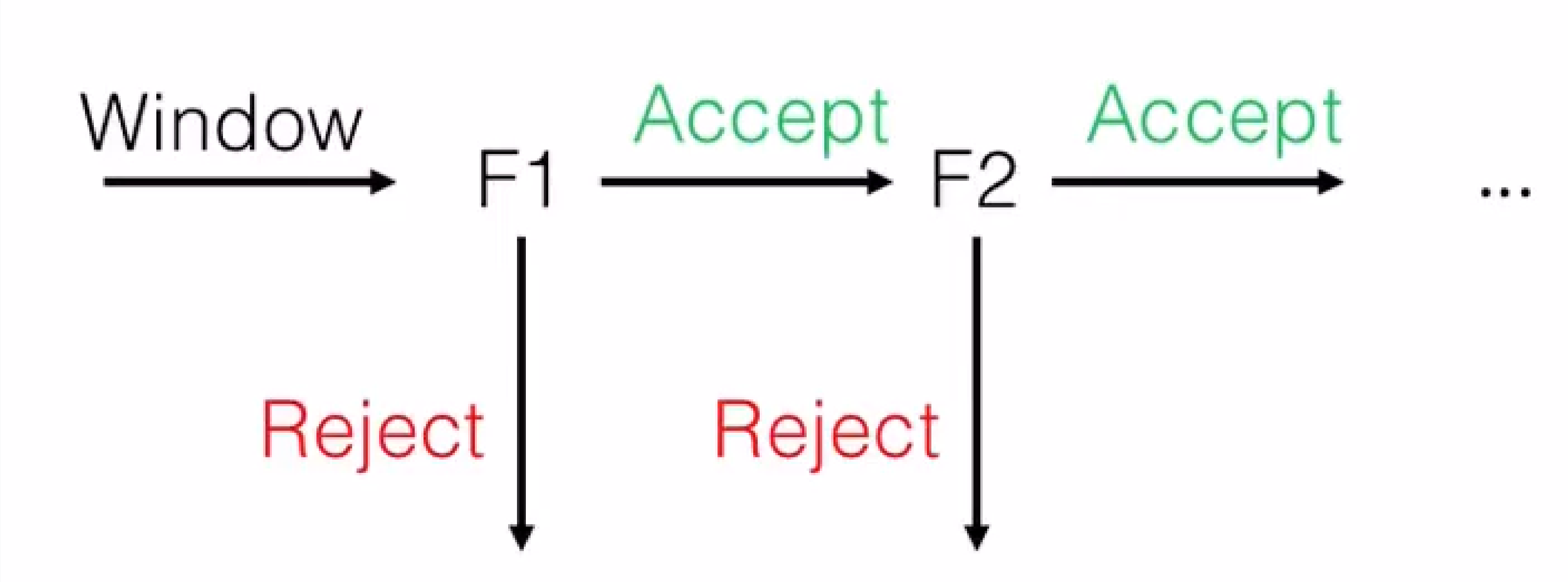


The higher the value, the more likely whatever the Haar-like feature represents exist in this region




# AdaBoost Training

Adaboost essentially chooses the best features and trains the classifiers to use them. It uses a combination of **"weak classifiers"** to create a **"strong classifier"** that the algorithm can use to detect objects.




## CASCADING CLASSIFIERS

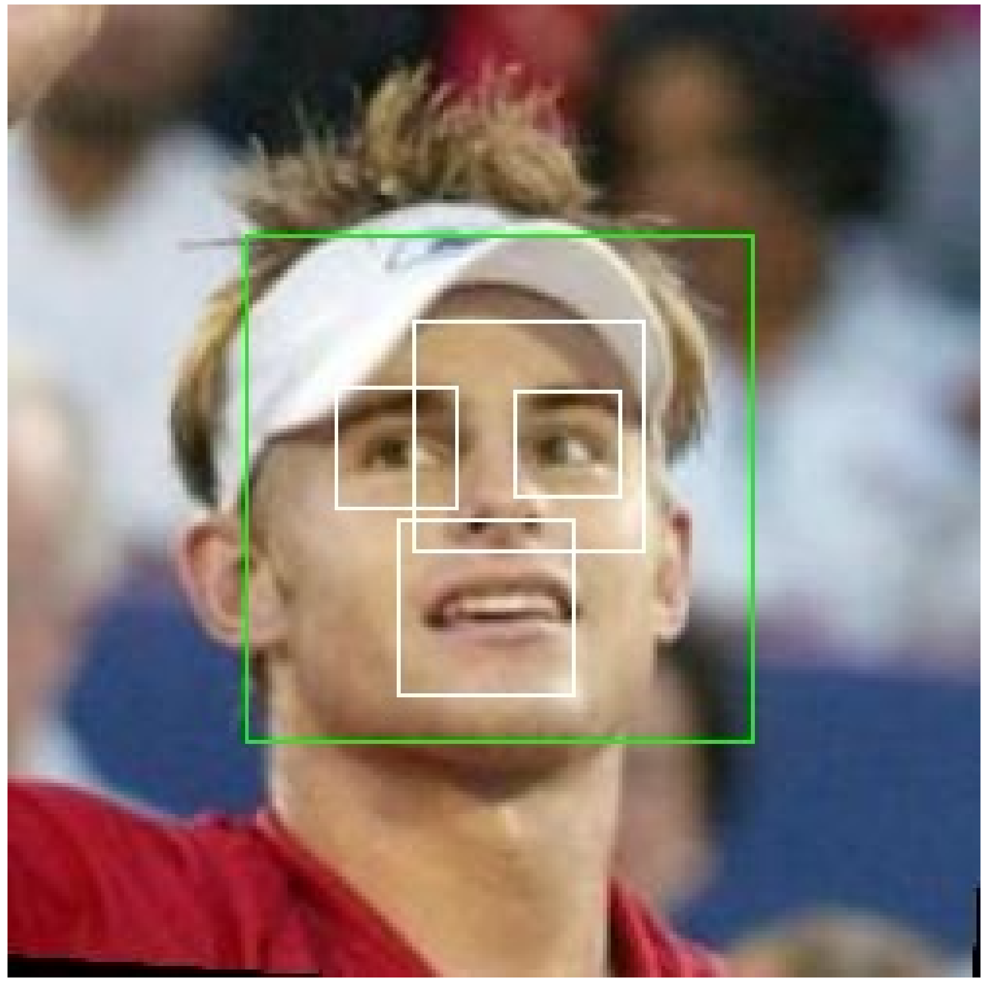
After performing the Adaboost training with a labeled dataset (face and non-face), Haar features are weighted and used for classification. The first classifier is with the most important feature (F1), the 2nd classifier is with the 2nd most important feature (F2), and so on. Trying all the features is time-consuming but with cascading, the process gets faster.

# Face Alignment

- Alignment minimizes variations due to head tilt, rotation, or scale differences, and establishes a standardized reference frame.
- Steps for alignment:
  - Finding the location of the eyes and their center.
  - Calculating the distance and angle between eyes.
  - Applying rotation transformation.

Images in the dataset is already aligned.
The issues with Haar Feature-based Cascade Classifier when detecting eyes:

- Only performs well on the front-face images
- Requires manual tuning of hyparameters to be executed.



# Image Cropping

Face cropping separates the face region from the backdrop components, focusing the emphasis on key facial characteristics. The reduction of noise also improves the model's capacity to extract discriminative features.

# Grayscaling Image

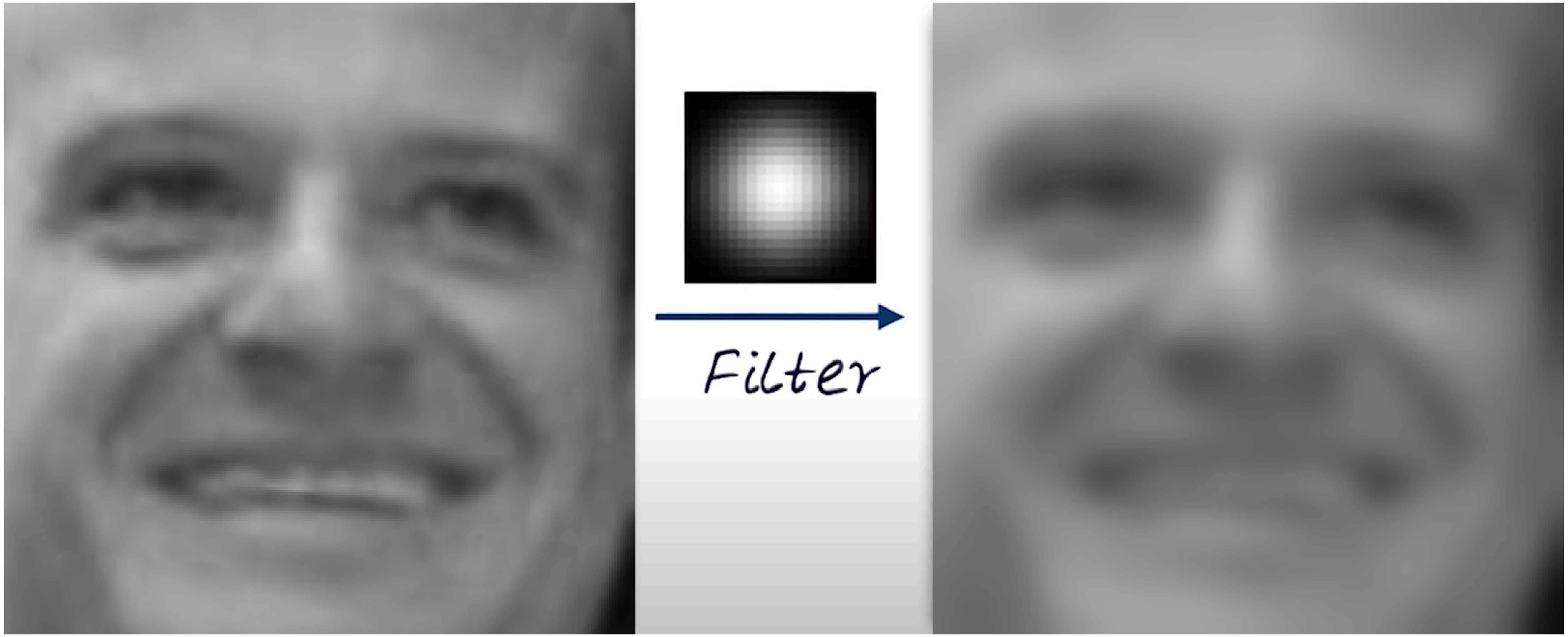
Dimensional reduced and grayscaled images are less sensitive to changes in illumination compared to color images

We convert RGB image to grayscale image by using formula:

$$Value = 0.299 \times Red + 0.587 \times Green + 0.114 \times Blue$$

# Gaussian Blur

A technique to reduce image noise, provide a smooth transition between neighboring pixel values. It involves convolving the image with a Gaussian filter. The convolution operation replaces each pixel value with a weighted average of its neighboring pixels, giving more weight to the central pixels.




Original Image

Blurred Image

The standard deviation σ would determine the spread of the blur, the lower it get, the more central-based it becomes

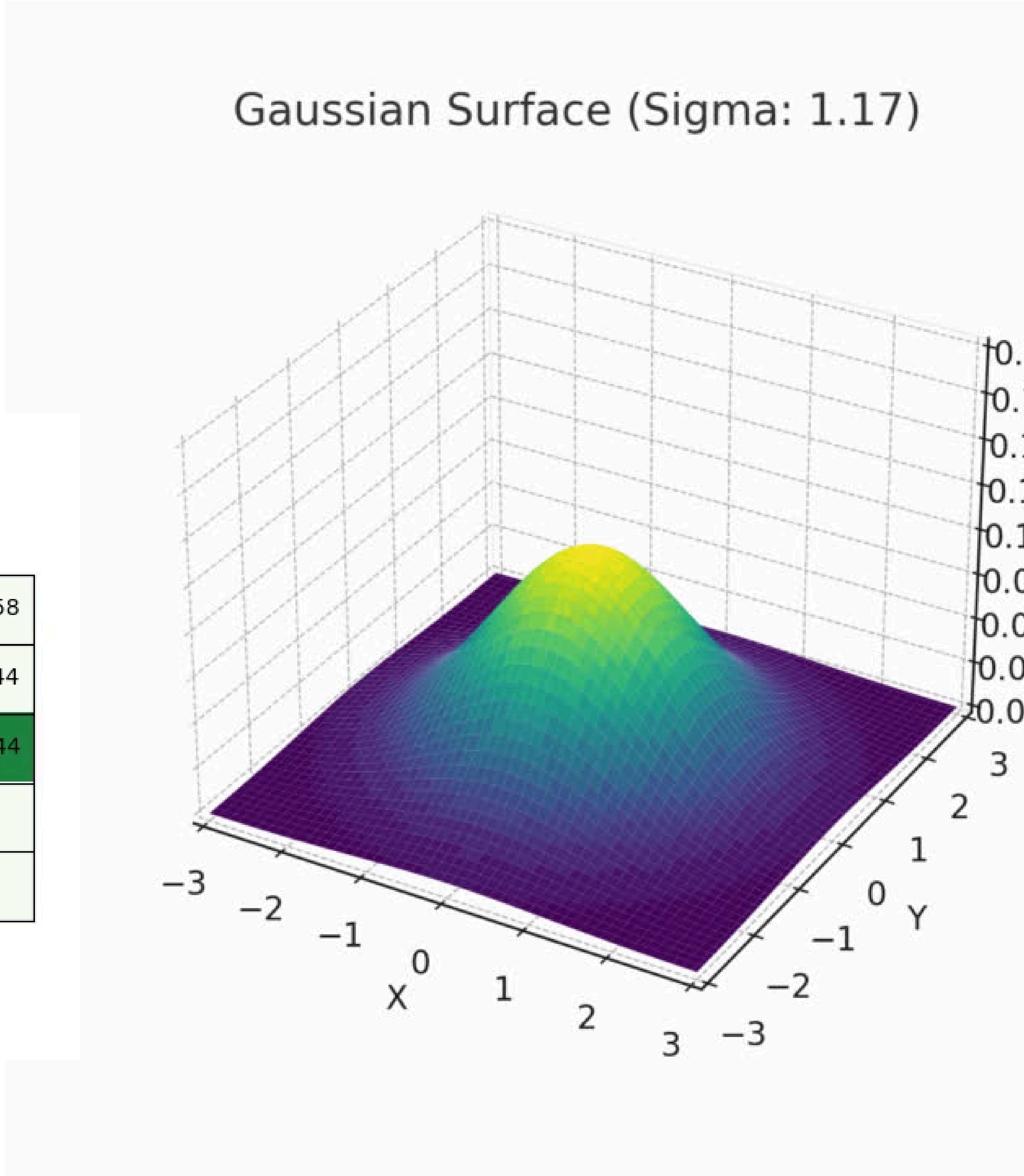
| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 60 | 113 | 56 | 139 | 85 | 0 |
| 0 | 73 | 121 | 54 | 84 | 128 | 0 |
| 0 | 131 | 99 | 70 | 129 | 127 | 0 |
| 0 | 80 | 57 | 115 | 69 | 134 | 0 |
| 0 | 104 | 126 | 123 | 95 | 130 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Kernel

| | | |
|---|---|---|
| 0.075 | 0.124 | 0.075 |
| 0.124 | 0.204 | 0.124 |
| 0.075 | 0.124 | 0.075 |

| | | | | |
|---|---|---|---|---|
| 114 | 328 | -26 | 470 | 158 |
| 53 | 266 | -61 | -30 | 344 |
| 403 | 116 | -47 | 295 | 244 |
| | | | | |
| | | | | |

*How the kernel applies over the image pixels*



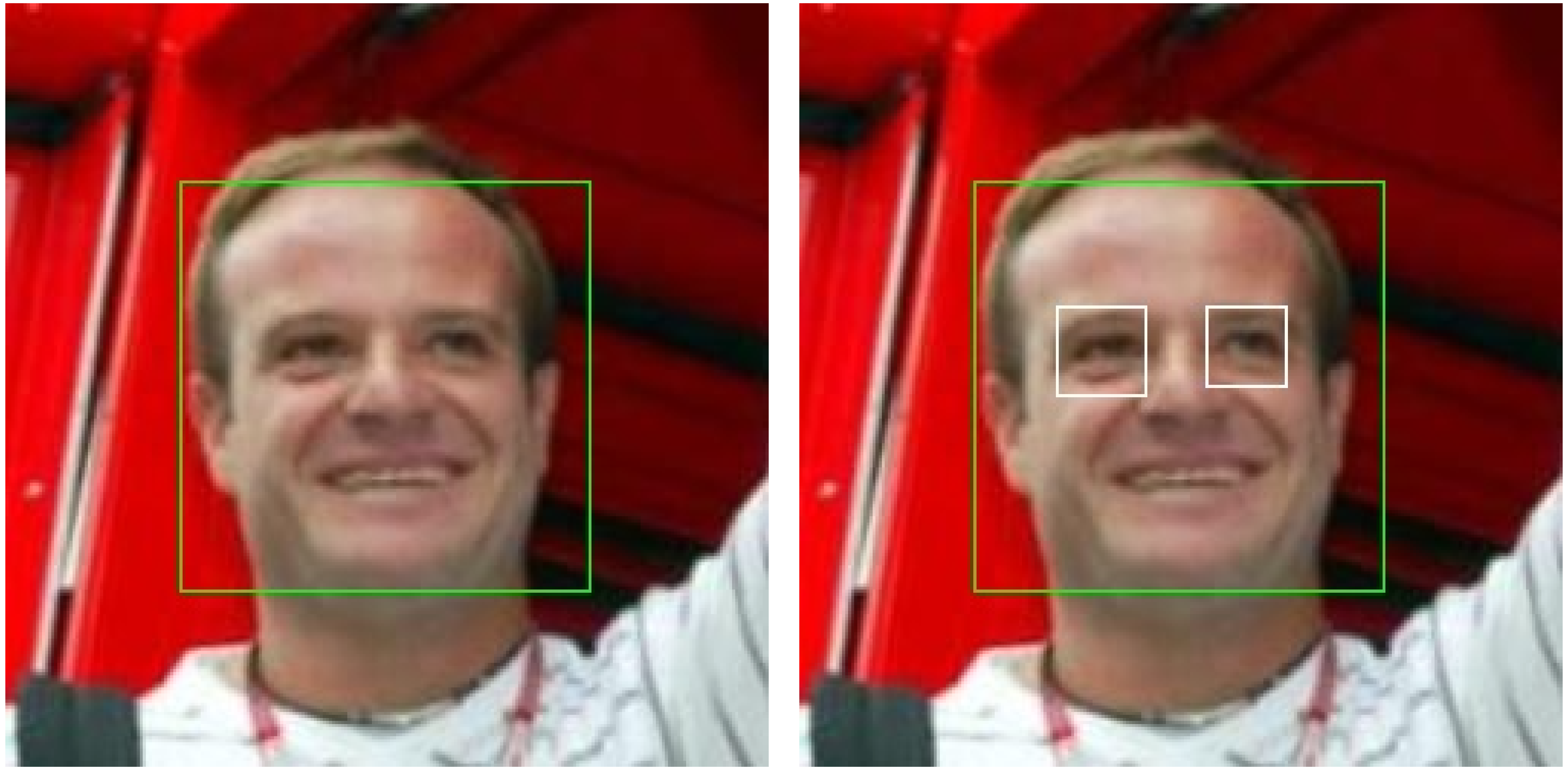
# Normalizing Images

Applying visual normalization in order to fix very dark/light pictures (can even fix low contrast) and improve model covergence.

original

normalized

# Illustrations



*Face and Eyes Detection*

*Aligning and Cropping Image*

***Cropped Image***

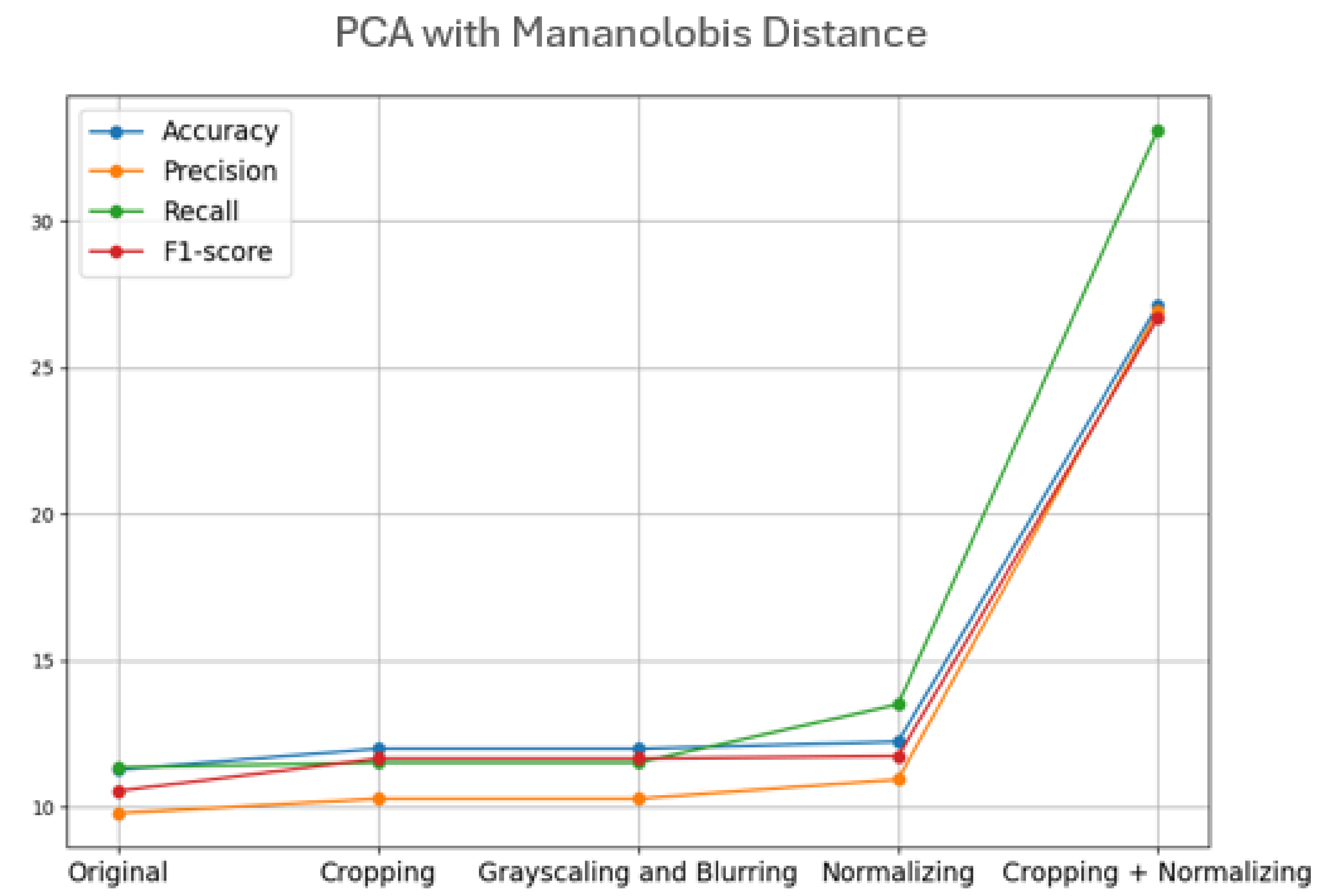
***Grayscaled Image***

***Blurred Image***

# Experiments

## PCA with Similarity Detection Approach

- While applying Chi Square shows worst results, Mahalanobis and Manhattan are on the opposite.
- Grayscaling and blurring before PCA have no impact on the performance.
- Cropping faces and normalizing images can improve the results up to 3 times



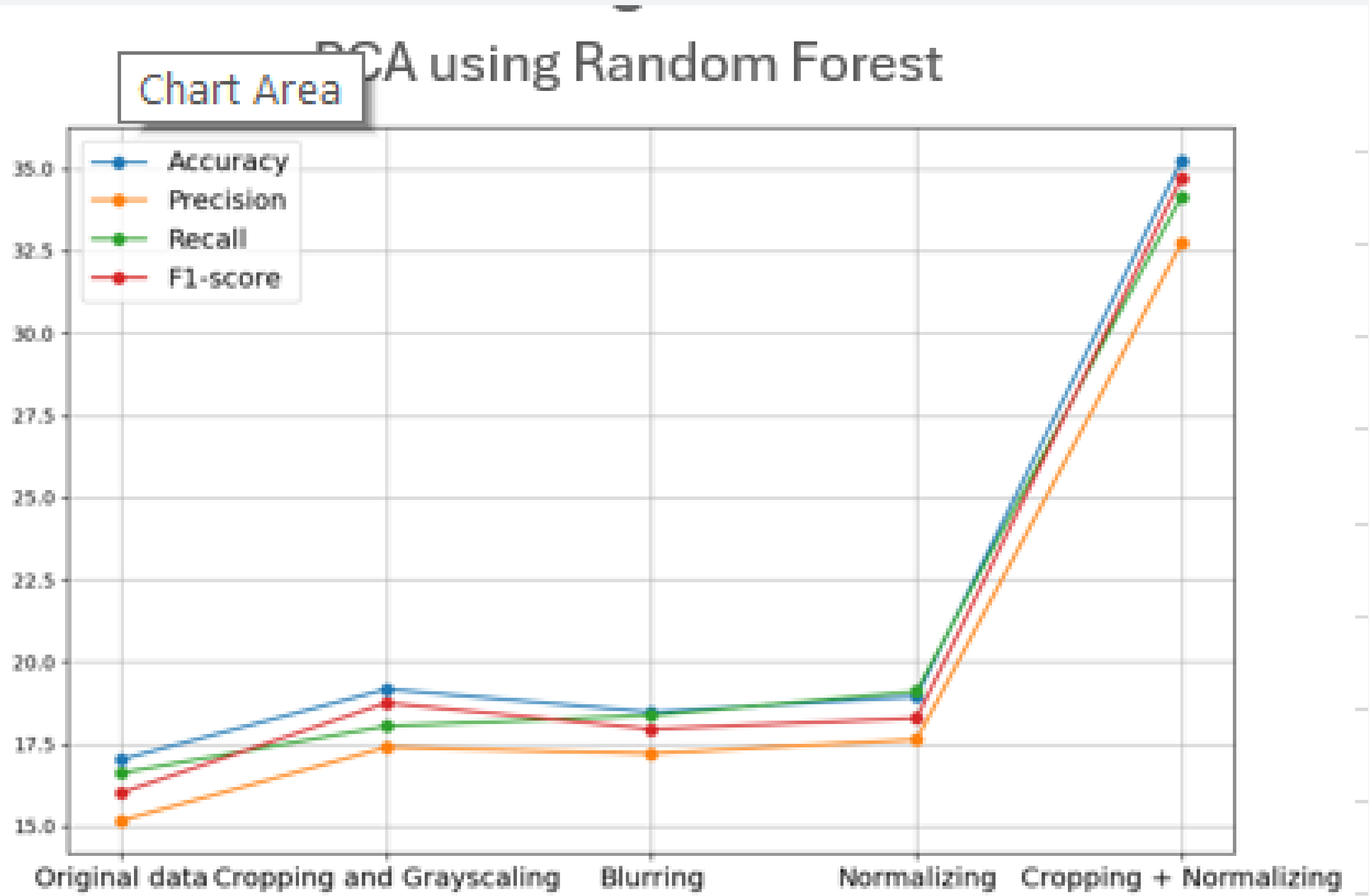

| | Original data | Cropping faces | Grayscaling and Blurring | Normalizing | Cropping + Normalizing |
|---|---|---|---|---|---|
| Accuracy | 11.27 | 11.99 | 11.99 | 12.23 | 27.1 |
| Precision | 9.79 | 10.28 | 10.28 | 10.94 | 26.91 |
| Recall | 11.35 | 11.51 | 11.51 | 13.5 | 33.1 |
| F1-score | 10.55 | 11.65 | 11.65 | 11.73 | 26.67 |

Accuracy Precision Recall F1-score

# Experiments

## PCA with Model Classification

- DecisionTreeClassifier always yield poorest outcomes while Logistic Regression and LDA tend to perform the best overall among all others. Notably, grayscaling images decrease the scores by nearly 50%



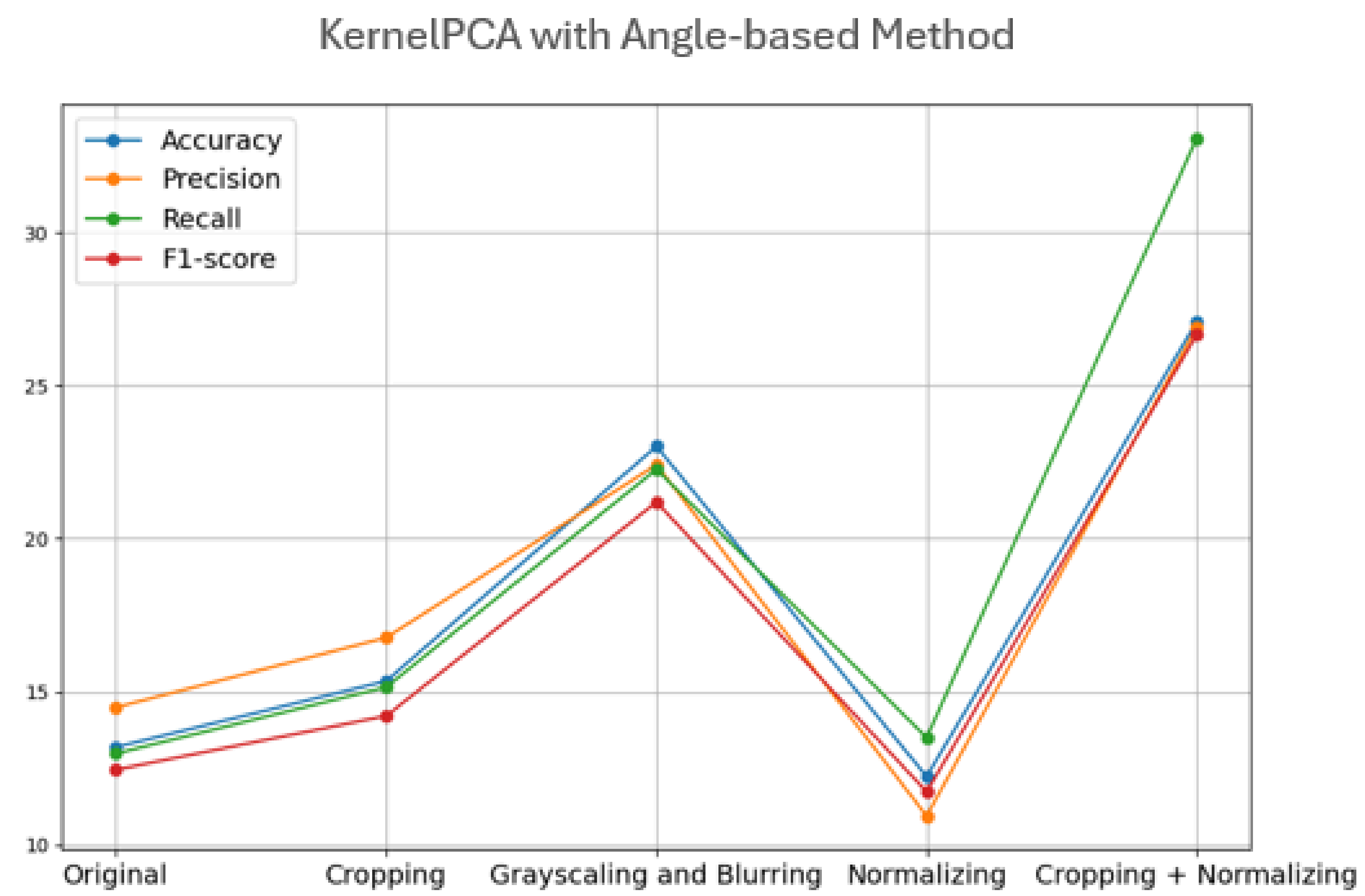

| | Original data | Cropping | Grayscaling | Blurring | Normalizing | Cropping + Normalizing |
|---|---|---|---|---|---|---|
| Accuracy | 18.47 | 19.94 | 11.03 | 17.75 | 19.18 | 35.01 |
| Precision | 14.82 | 15.27 | 7.34 | 13.2 | 14.72 | 30.22 |
| Recall | 17.63 | 15.83 | 7.18 | 15.27 | 15.59 | 32.06 |
| F1-score | 16.28 | 17.37 | 9.65 | 15.57 | 17.28 | 33.77 |

Accuracy Precision Recall F1-score

# Experiments

## KernelPCA with Similarity Detection Approach

- Angle-based approaches and Modified-SSE achieve the highest performance scores.
- During the face-cropping phase, Euclidean and Manhattan Distances express a downtrend.
- While blurring results in a significant increase, ranging between 6% to 9% across all metrics. However, this contributes to the dropdown of results after using normalization



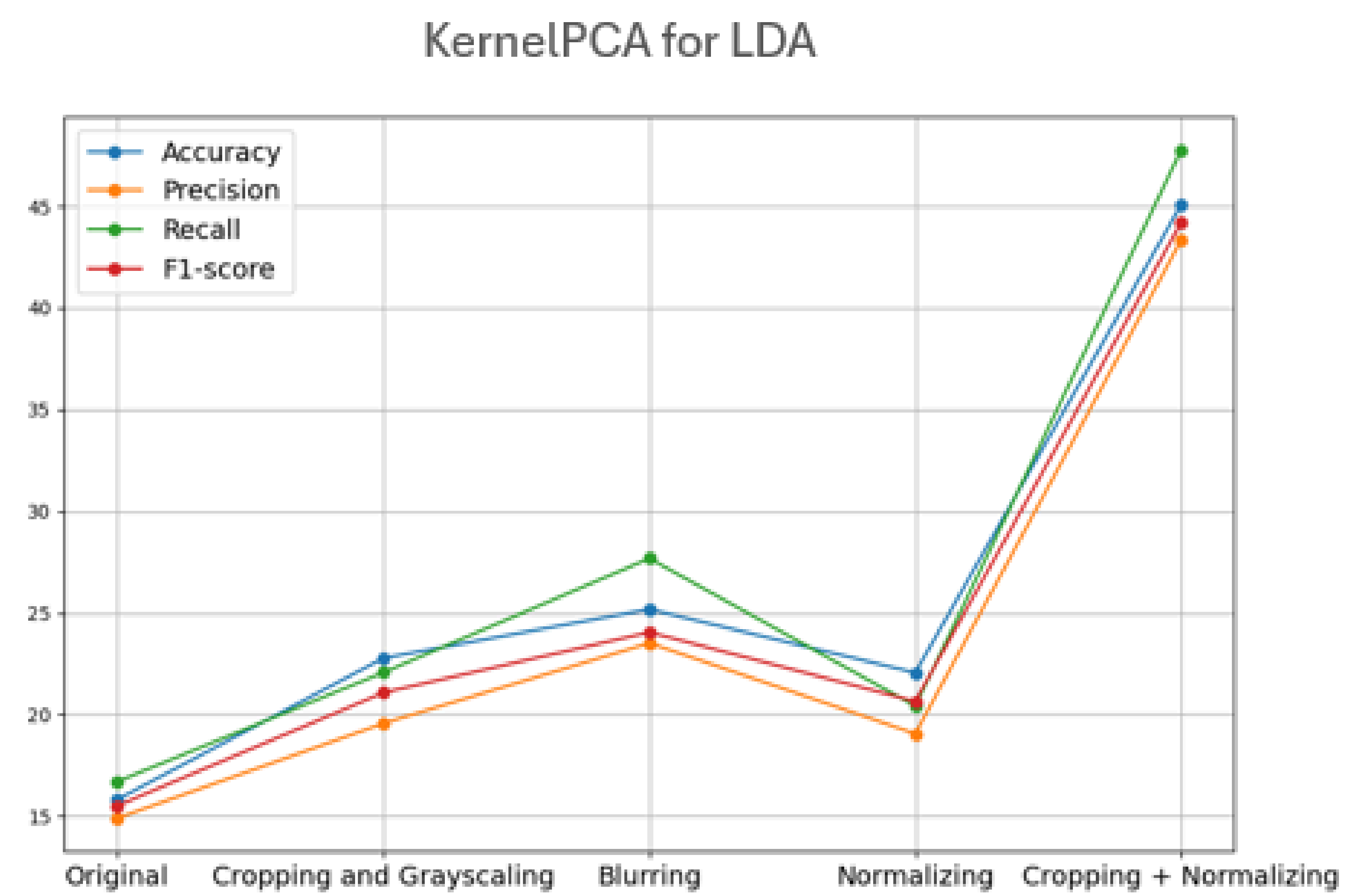

| | Original data | Cropping and Grayscaling | Blurring | Normalizing | Cropping + Normalizing |
|---|---|---|---|---|---|
| Accuracy | 13.19 | 15.35 | 23.02 | 12.23 | 27.1 |
| Precision | 14.49 | 16.77 | 22.42 | 10.94 | 26.91 |
| Recall | 12.98 | 15.15 | 22.27 | 13.5 | 33.1 |
| F1-score | 12.45 | 14.21 | 21.2 | 11.73 | 26.67 |

Accuracy Precision Recall F1-score

# Experiments

## KernelPCA with Model Classification

- Cropping faces increases significantly from 6-13%, except for Random Forest and LDA.




| | Original data | Cropping and Grayscaling | Blurring | Normalizing | Cropping + Normalizing |
|---|---|---|---|---|---|
| Accuracy | 15.83 | 22.78 | 25.18 | 22.06 | 45.08 |
| Precision | 14.91 | 19.57 | 23.52 | 19.06 | 43.34 |
| Recall | 16.71 | 22.07 | 27.68 | 20.42 | 47.77 |
| F1-score | 15.52 | 21.08 | 24.05 | 20.65 | 44.2 |

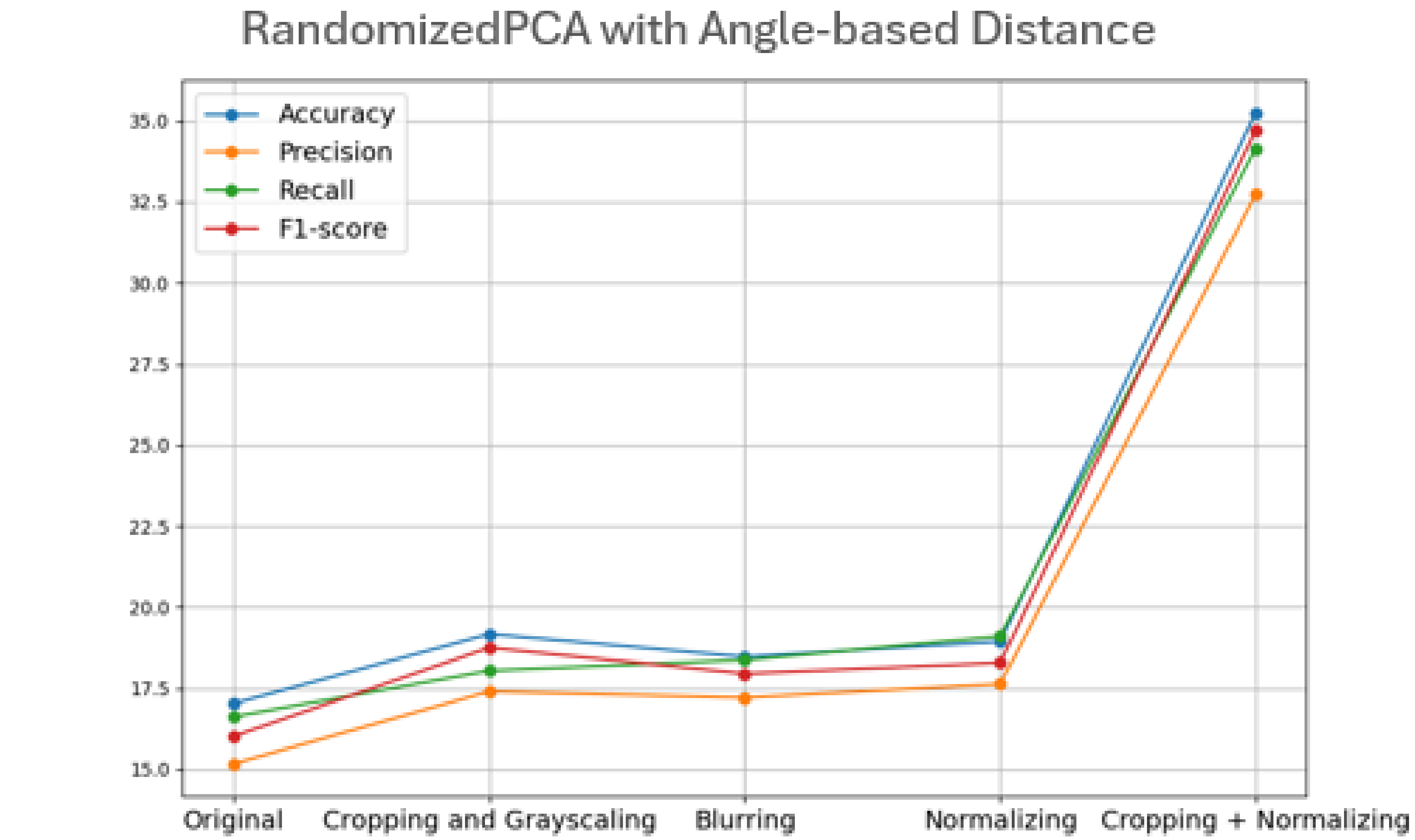
Accuracy Precision Recall F1-score

# Experiments

## RandomizedPCA with Similarity Detection Approach

Compared to PCA and SparsePCA, RandomizedPCA takes less training time with the highest scores achieved when using Angle-based and Correlation coefficient-based methods.

However, Grayscaling and Blurring would decrease the scores. Other methods like cropping faces, normalizing increases insignificantly



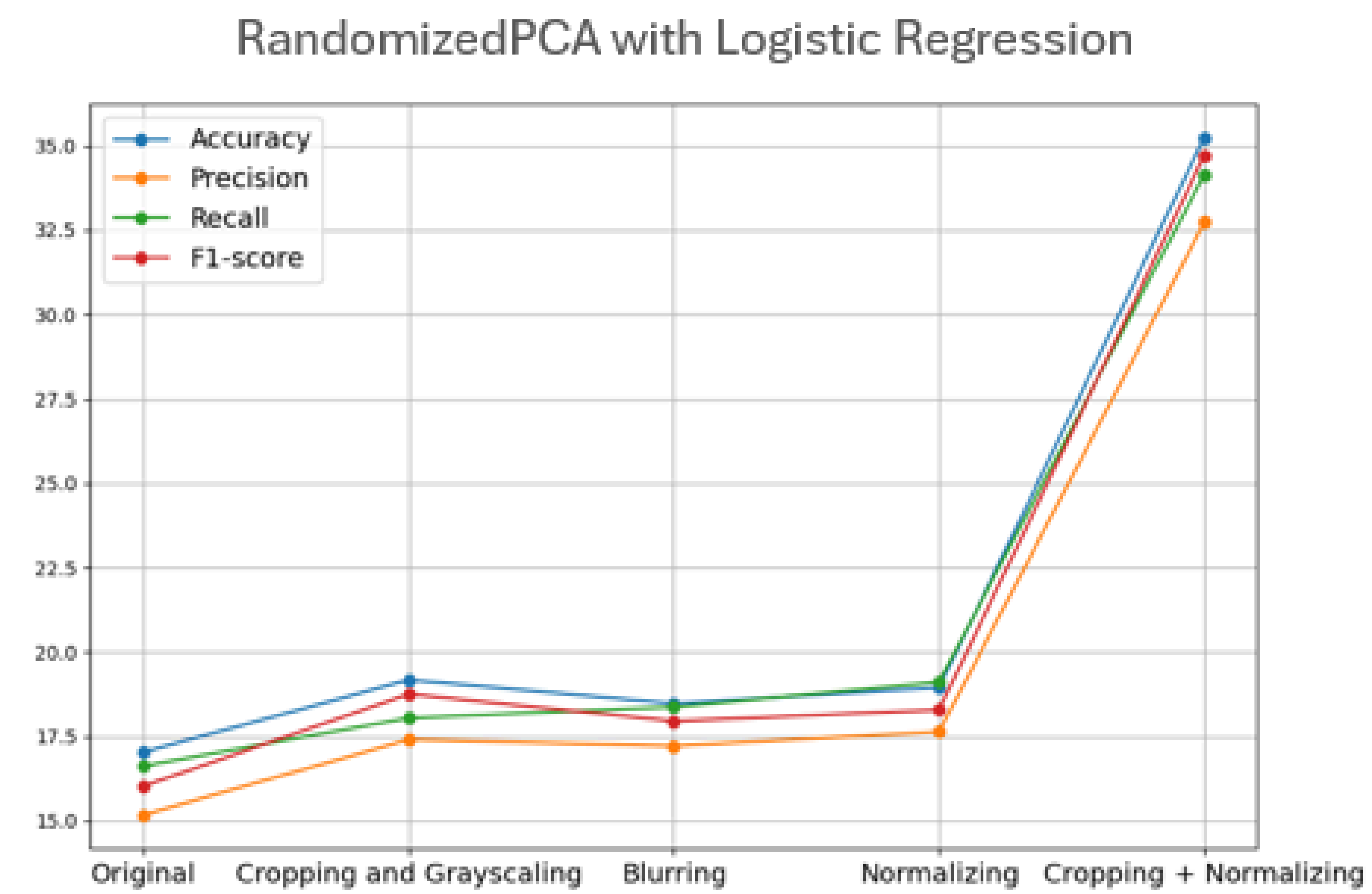

| | Original data | Cropping and Grayscaling | Blurring | Normalizing | Cropping + Normalizing |
|---|---|---|---|---|---|
| Accuracy | 17.03 | 19.17 | 18.47 | 18.94 | 35.25 |
| Precision | 15.17 | 17.4 | 17.21 | 17.63 | 32.73 |
| Recall | 16.62 | 18.04 | 18.37 | 19.1 | 34.15 |
| F1-score | 16.02 | 18.75 | 17.95 | 18.28 | 34.72 |

Accuracy Precision Recall F1-score

# Experiments

## RandomizedPCA with Model Classification

- Cropping faces contributes significantly in improving results, while bluring images has no effect.



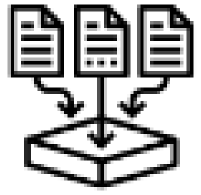

| | Original data | Cropping | Grayscaling and Bluring | Normalizing | Cropping + Normalizing |
|---|---|---|---|---|---|
| Accuracy | 13.19 | 15.35 | 23.02 | 23.26 | 41.49 |
| Precision | 14.49 | 16.77 | 22.42 | 18.54 | 38.98 |
| Recall | 12.98 | 15.15 | 22.27 | 18.69 | 43.17 |
| F1-score | 12.45 | 14.21 | 21.2 | 20.9 | 40.2 |

Accuracy Precision Recall F1-score

# Conclusion

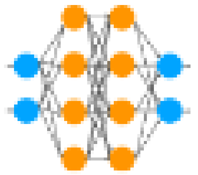
- Among distance measurement methods:
  - Chi-square distance demonstrates the least effective performance.
  - In task 2, Manhattan distance returns the best results, whereas angle-based distance proves to be the most effective in task 3.
- Regarding classification models, Logistic Regression and LDA exhibit superior performance, while Decision Tree performs the least effectively
- In preprocessing stage:
  - Significant improvement comes from cropping faces; combining it with image normalization triples results compared to using only original data.
  - Grayscaling image doesn’t contribute to improve results when using PCA as it decreases the variance of data.

# Future Work

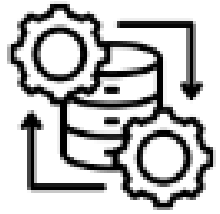


Adding more training data.

Applying more face detection algorithm such that RetinaFace, CNN models,etc.



Applying deep learning models for classification.



Trying new different techniques of preprocessing data

# Contributions

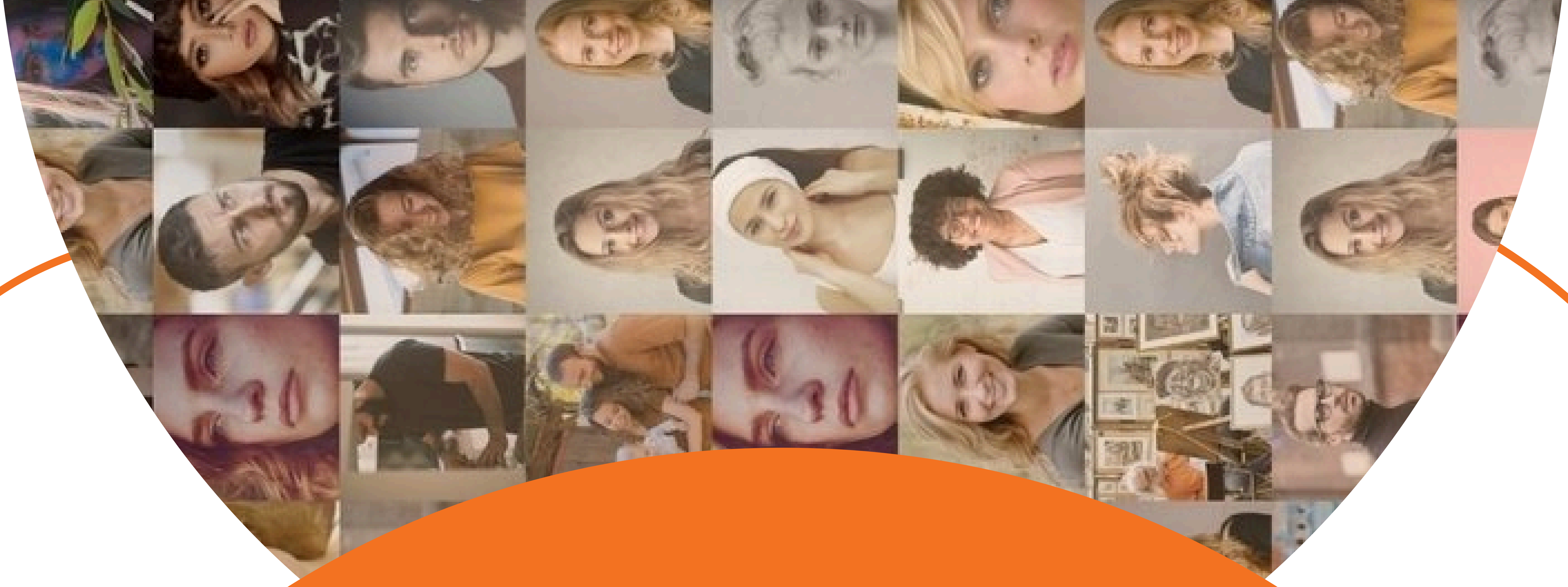
| Members | Tasks | Contributions (%) |
|---|---|---|
| **Trịnh Thị Minh Tâm 11215226** | • Planning, assigning and review tasks to team members<br>• Do background research about Face Recognition Survey, Preprocessing in task 3<br>• Baseline code for task 2 and task 3<br>• Inspect results and write report for task 3<br>• Making slides | **27** |
| **Phan Anh Khôi 11212902** | • Do background research about Eigenfaces method, Distance measures, Evaluation metrics, Sparse PCA, Haar-like features.<br>• Inspect results and write report for task 1 and 2.<br>• Implement PCA and fit the data to the model.<br>• Making slides | **23** |
| **Nguyễn Thành Long 11213549** | • Literature Review and Background Research Gaussian Blur<br>• Write 2 function for plotting: feature importance and misregconized cases<br>• Finding non-face and face data, error analysis<br>• Making slides | **12** |

# Contributions

| Members | Tasks | Contribution (%) |
|---|---|---|
| **Hoàng Thị Tú Anh 11219255** | • Literature Review and do background research about Evaluation Metric, Retina Face<br>• Code for model classification in task 2<br>• Inspect result for Hyperparameters Tuning: Using GridSearchCV<br>• Fit the data for the model KernelPCA and IncrementalPCA<br>• Making slides | **19** |
| **Kiều Thanh Tâm 11219286** | • Literature Review and do background reasearch about CNN, Kernel and Randomized PCA, Retina Face, SVM, and LDA<br>• Code fitting for model PCA and other model<br>• Fit the data for model Randomized PCA and Sparse PCA<br>• Making slides<br>• Inspect results for Hyperparameter Tuning: K-fold and LeaveoneOut | **19** |

# Reference


1. Mayank Kumar Rusia, Dushyant Kumar Singh (2023), A comprehensive survey on techniques to handle face identity threats: challenges and opportunities. Multimedia Tools and Applications: 1669-1748
2. Müge Çarıkçı and Figen Özen (2012), A Face Recognition System Based on Eigenfaces Method. Procedia Technology 1: 118 – 123
3. Rabia Jafri and Hamid R. Arabnia (2009), A Survey of Face Recognition Techniques. Journal of Information Processing Systems: Vol.5, No.2,
4. R. Gottumukkal, V.K. Asari, An improved face recognition technique based on modular PCA approach (2004). Pattern Recognition Letters 25: 429–436
5. RAMA CHELLAPPA, FELLOW, IEEE, CHARLES L. WILSON, SENIOR MEMBER, IEEE, AND SAAD SIROHEY, MEMBER and IEEE, Human and Machine Recognition of Faces: A survey(1995), Proceedings of the Ieee:Vol 86, No5.
6. Vytautas Perlibakas, Distance measures for PCA-based face recognition (2004). Pattern Recognition Letters 25: 711-724
7. Yassin Kortli, Maher Jridi , Ayman Al Falou and Mohamed Atri(2020). Sensors: 20,342
8. Md. Omar Faruqe and Md. Al Mehedi Hasan(2009). Face Recognition Using PCA and SVM. 978-1-4244-3884-6/ 09/ $25.00 02009 IEEE
9. Eman Zakaria, [2] Wael Abdel Rahman, [3] Abeer Twakol, [4] Ashraf Shawky, Face Recognition using Deep Neural Network (2016). SL International Conference Giza.



# Thank You