

# CHAPTER 1

## INTRODUCTION

### 1.1. Background

Recognizing faces sounds like a simple thing to humans. We can easily recognize someone in person or maybe through pictures or videos. Nevertheless, it is not that easy for computer vision to do it.

Back in the day, we could only see the use of face recognition technology on television or maybe in movies, but now facial recognition technology is commonly used in various fields. Our smartphone is one such example, and almost every phone has the face unlock feature nowadays. Because of that, so many algorithms are developed in order to find the most optimum in terms of time, speed, and costs. One of them is Neural Networks. There has been a surge of interest in neural networks, particularly deep and large networks. These networks have exhibited impressive results [1].

However, besides the significant advantages, beginner researchers have one problem: The approach is computationally expensive and requires a high degree of correlation between the test and training images [2]. What should we do to overcome this weakness?

There is another approach to recognizing faces by using a statistical approach or trying to search for patterns. One of the most popular algorithm was Principal Component Algorithm (PCA). This algorithm is also called Eigenfaces when implemented on the images. With more increase in the size of the training set, the algorithm shows increased accuracy [3]. This might be a solution for beginners who want to implement face recognition and do not have adequate resources.

That is why this project specifically compares those two algorithms. Will there be significant results in time, speed, accuracy, or even if these two algorithms compete with the results. Which one is more effective with the environment given? Should we still use a neural network with relatively large resources but high accuracy? Or is it better to use the PCA algorithm as an alternative, which has fewer resources?

## **1.2. Problem Formulation**

1. How is the performance of the DNN and PCA algorithms in terms of accuracy and speed?
2. Which algorithm is more optimal with the given circumstances?
3. Could the PCA algorithm outperform the DNN algorithm in a particular condition?

## **1.3. Scope**

Two algorithms, namely DNN and PCA, are used to perform facial recognition. The DNN algorithm is implemented with the help of the Keras library, while the PCA algorithm is done from scratch. As the first step, load the dataset. For the DNN algorithm coding, the dataset is loaded from Scikit Learn or often called Sklearn. Later the Sklearn is used as a loader for the Olivetti faces dataset (ORL). This dataset contains pictures of 40 people with some variances in lighting, expressions, and accessories. Then, after the dataset preprocessing is complete, the face recognition process begins. This process includes the training and test phase. The DNN and PCA algorithm are implemented separately, and a k-fold cross-validation technique is used to split the datasets for the training and test phase. Finally, we compare the result in terms of time, speed, and ease of implementation.

## **1.4. Objective**

This project aims to test the DNN and PCA algorithm in face recognition and compare each algorithm's accuracy in given circumstances. Both of the algorithms are trained and tested with the same but randomized image datasets from ORL.