

**Application of the Facial Reignitions**

by

Talha Tallat

This report is submitted in partial fulfilment of the requirements of the Degree in Electronics and Communications Engineering (DT008) of the Technological University Dublin

April 30th, 2020

Supervisor: Dr Kevin Tiernan

School of Electrical and Electronic Engineering

**DECLARATION**

I, the undersigned, declare that this report is entirely my own written work, except where otherwise accredited, and that it has not been submitted for a degree or other award to any other university or institution.

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgements**

The project would not be possible without the wonderful advice and guidance of a group of remarkable staff members in the school of Electrical and Electronic Engineering at Technological University Dublin.

I would like to thank the project supervisor Dr Kevin Tiernan for his great advice and supervision of the project from the very start till the final stage.

**Abstract**

The project aimed to build and test a facial recognition system which is cable of detecting faces and importantly can identify the face.

…

….

….

….

Contents

[1. Introduction to Facial Recognition 7](#_Toc37326332)

[1.1. Project Overview 7](#_Toc37326333)

[1.1.1. Introduction to Facial Recognition 7](#_Toc37326334)

[1.1.2. Introduction to Anaconda 7](#_Toc37326335)

[1.1.3. Introduction to OpenCV 8](#_Toc37326336)

[1.1.4. Aim of the project report 8](#_Toc37326337)

[2. Project Research 9](#_Toc37326338)

[3. Setup for facial Recognition 10](#_Toc37326339)

[3.1. Introduction to Python 10](#_Toc37326340)

[3.1.1. Anaconda Setup 10](#_Toc37326341)

[3.2. OpenCV setup 11](#_Toc37326342)

[3.2.1. Manual Installation of OpenCV for Python 11](#_Toc37326343)

[3.2.2. OpenCV Installation with Anaconda Navigator 12](#_Toc37326344)

[3.3. Haars Cascade algorithm 13](#_Toc37326345)

[3.4. DetectionMultiscale 14](#_Toc37326346)

[3.4.1. Os.walk 14](#_Toc37326347)

[4. Test procedures and results 15](#_Toc37326348)

[5. Conclusion 16](#_Toc37326349)

[6. References 17](#_Toc37326350)

Figures

[Figure 1 - List of Anaconda's available Package 8](#_Toc37326351)

[Figure 2 - Available versions for Anaconda's installation 10](#_Toc37326352)

[Figure 3 - Python Version is 3.7.4 11](#_Toc37326353)

[Figure 4 - Anaconda version is 1.7.2 11](#_Toc37326354)

[Figure 5 - Importing libraries to test, if these are installed 11](#_Toc37326355)

[Figure 6 - New environment created in Anaconda Navigator 12](#_Toc37326356)

[Figure 7 - Command for OpenCV installation in Anaconda prompt 12](#_Toc37326357)

[Figure 8 - Available OpenCV library to install in Anaconda Navigator 13](#_Toc37326358)

[Figure 9 - Importing 'OpenCV' library to check, if it's installed 13](#_Toc37326359)

# Introduction to Facial Recognition

## Project Overview

### Introduction to Facial Recognition

A **Facial recognition** system is a technology capable of identifying or verifying a human face in a digital image, video or a live video source from a camera. This technology detects people’s face and recognises who the person is. It They works by reading the facial features of a given image and compares them with facial features stored the faces with a face within a database. There are two major parts to this project which are listed below.

* Face Detection
* Facial Reignition

Face Detection algorithms are mainly focused on the frontal face only. Their purpose is to locate those regions within an image containing human faces. <<Don’t go off topic here, keep discussing face detection. You could incorporate the following highlighted section in a more detailed description of face recognition>> Once these regions are detected then a separate algorithm will operate It is an analogue to an image detection which an image of the person is matched bit by bit. The image match with the images stored in the database. Any facial feature changes in the database will evirate the match in the process. There are many applications based on face detection such as facial recognition, photography, marketing etc. In this project face detection operates as a pre-processing stage to face recognition by supplying the recognition algorithm with rectangular regions of the image containing just the human face. In contrast to facial recognition, it just focusses on the detection of faces.

So the primary difference is that face detection algorithms focus on locating any face within an image whereas facial recognition focuses on identifying the person corresponding to the image. recognising person too.

<Consider removing the following section as it is just repetition of earlier points>

* The **face detection** process is an important step as it detects and discovers human faces in images and videos.
* The **face capture** process transforms analogue information into digital information data based on the person’s facial features.
* The **face match** process compares two faces bit by bit and verifies that both faces belong to each other.

These facial recognition technologies are used every day for things like phone security and for serious uses from important organizations, such as like, banks and governments.

Some banks are looking at implementing facial recognition technologies into their ATMs as a security measure for verifying transactions.

Government agencies, such as ICE and the FBI are using facial recognition to create the database from existing documents like drivers’ licenses.

### Introduction to Anaconda

**Anaconda** is a free and open-source distribution of the Python and R programming language for scientific computing, such as data science, large scale data processing, machine learning applications etc. Its The goal is to provide simplified Python package management and deployment.

Anaconda is very popular because it provides short and simple setup and most importantly, it has many of the tools used in data science and machine learning with just only one installation. Anaconda Distribution contains Anaconda Navigator, Python and hundreds of scientific packages. For this project the components Spyder and NumPy were employed. Spyder is a Matlab like IDE for developing Python programs and NumPy extends the Python language for numerically intensive scientific applications. Anaconda was also used to setup and install OpenCV which is an optimised library for computer vision applications.

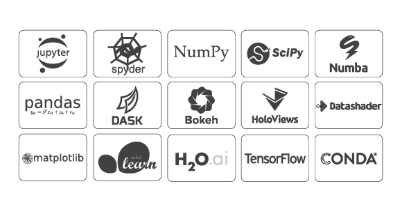


Figure 1 - List of Anaconda's available Package

### Introduction to OpenCV

**OpenCV Open CV** is an open-source computer vision library originally developed by the Intel. It mostly used for all the various face recognitions, object recognitions, and all other machine learning and a deep learning technique. Open cv is cross-platform itself but originally written in C/C++ and now it provides bindings for various programming languages, such as Python, Java, MATLAB, Interfaces, Windows and Linux.

For this project OpenCV uses machine learning algorithms to search for and identify faces within a picture. Don’t go in to any great detail of the algorithms here. You could just state that the Haars cascade was used for face detection and state the name of the face recognition algorithm used. You could then perhaps briefly describe what their adavantages or why they were chosen. Leave any detailed explanation later in another chapter that describing the algorithms. The introduction chapter should just give an overview of the project but should not go into too much detail. Because faces are difficult task since it requires thousands of small patterns and features to be matched. These algorithms break the task of identifying the faces into small bite-sized tasks, making it easy to solve. These tasks are also known as classifiers.

…

Classifier??

…

Plots??

….

The algorithms start at the top left of a picture and move down across small blocks of data, looking at each block, constantly asking, “Is this a face? ... Is this a face? ….” Since there are approximately 6,000 or more tests per block, that might have millions of calculations which will grind the computer to a halt.

To avoid this, OpenCV uses **cascades**, which breaks the problem of detecting faces into multiple stages. For each block, it performs a rough test, if the test passes then it performs a more detailed test and continues to do for the rest of them. These algorithms may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.

**Cascades** are just a bunch of XML files that contain OpenCV data used to detect objects. Cascade does all the work when initializing code with the selected cascade.

..

### Aim of the project report

This report contains the research, setup of the face detection and facial recognition of the overall facial reignition project.

The initial aim of the project was to design and program a facial recognition system that can identify human’s faces and making sure that is suitable for security purposes, whether somebody can rely on this system or not.

The project is divided into several different objectives parts that are required to complete this project.

Each chapter should start on a new page

# Project Research

* 1. Initial Research

**Facial Recognition** has made great impacts on the lives of people, thanks to the learning machines that can read the faces. The face is a password that can unlock smartphones devices and bank accounts or anything that requires security can be secured. The facial recognition can track any movements, it can guess human’s sexuality through facial features, it can identify people and capable of storing information of those people. The human face has an amazing variety of features, which does not only help us to recognize others but also allows to read and understand them through a constant flow of intentional and unintentional signals.

It’s one of the unique functions that separates humans from the machine, until late 2018, where companies like ‘Apple’ started to lunch these technologies that are built into their smartphones, which reads a face to unlock the phone. A user looks at the phone and it recognises that user in seconds Since there has been a huge amount of progress made in these technologies known as Machine learning, which allows pulling a very accurate faceprint out of a photograph that uniquely identifies a person.

The machine read faces with high precision and performs better when teaching machines more to read faces. This precision makes the technology more effective for any other use. These technologies are used in the retail stores to generate data on customers, tracking customers shopping habits and targeting in-store adverts. These facial reignition technologies are even used in churches and to monitor attendance and used in school’s to manage the student’s behaviour and put them into the watch list and these technologies are used in such place as surveillance, medical centre, retail stores, churches, mosques, etc.

The face recognition model can measure the distance between eyes, the width of your lips, the distance from lips to nose, etc. The software has the power to identify one face from millions in the database under one second and these Deep learning methods have accuracy under a controlled environment is around 99%. This technology can identify people without them knowing that they are being identified.

…

…

…

China is already using these technologies widely in security, looking for terrorists, people who have warrants out for their arrest, people can pay in fast-food restaurants, people can access theme parks without having to buy a ticket and also try and catch people who steal toilet papers from the public toilets. Companies in China have access to a government image database of more than 700 million people which is approximately half of its population. These abilities to record, store and analyse images of faces on a vast scale will fundamentally change notions of privacy, fairness and trust.

This chapter could have sections describing the Haars cacade and the recognition algorithm

…

.

…

…..Perhaps this chapter could describe the software

# Setup for facial Recognition

(This section could be bigger as you can explain how you got openCv and python working togather. Remember it took a few weeks to get it setup, so explain the difficulties. i.e.,  poor documentation and multiple versions of Python and openCv. that must have been compiled with the same compiler version to be compatible. Also, OpenCV is a huge open source library with many compiled variants available which may not contain all the components needed for this project, eg, Haars Cascades for face detection and FFMPEG for video capture.  Even though the eventual setup solution was simple, the poor documentation meant that much trial-and-error was needed to get started.)

## Introduction to Python

An appropriate way to start facial Recognition was to use Python programming language and the open-source library OpenCV to be able to process image. In order to run Python on Windows Operating system, a well-known “**Anaconda**” free and open-source distribution of them Python had to be installed.

…

…

…

Why did I choose to go with this???

….

…

### Anaconda Setup

Anaconda was installed on the windows operating system and the download link for the installation was available on Anaconda’s original website. The link for the installation as follows: <https://www.anaconda.com/distribution/>.

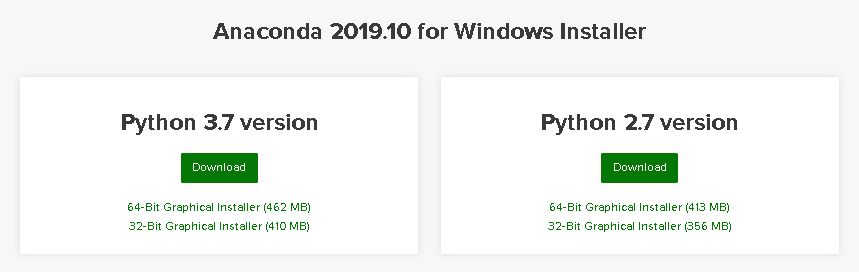


Figure 2 - Available versions for Anaconda's installation

There are “Python 3.7” and Python 2.7” versions available on Anaconda ‘s official website for windows installer as shown in figure 3. The latest version Python 3.7 (64-Bit) was installed on the computer with a size of 462 MB.

To make sure the right version is installed, the Anaconda Prompt was opened to check the right versions are installed. The commands for checking the Python version and Anaconda version are “python -V” & “Anaconda -V”.

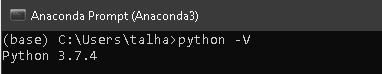


Figure 3 - Python Version is 3.7.4

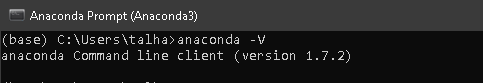


Figure 4 - Anaconda version is 1.7.2

Most of the libraries are already installed in Anaconda because Anaconda provides most of the useful libraries. The “sys” library was imported in the Python console to test the library, whether it is installed or not. If it is not, an error will show up, meaning that the library is not installed otherwise there is no error.

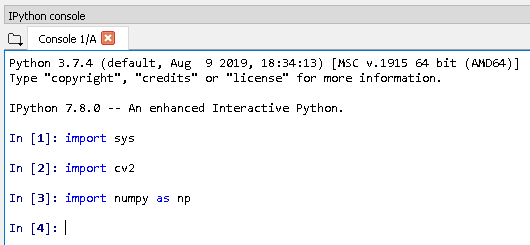


Figure 5 - Importing libraries to test, if these are installed

The most important library that needs to be installed is “OpenCV2” and Anaconda provides that library, which was installed throw Anaconda. The libraries were installed properly since it was checked in the console window by importing “import cv2” in the Console window. The window does not popes up with an error which meant that the libraries were installed properly as shown in figure 5.

..

..

## OpenCV setup

Open CV is the simplest way to recognize a face using Python language on Anaconda. To install open CV on the Anaconda, the Anaconda is installed first followed by OpenCV.

### Manual Installation of OpenCV for Python

OpenCV is a library for image processing and Computer Vision. Downloading the Anaconda graphical installer for windows platform from their original website to run the OpenCV library “<https://www.anaconda.com/distribution/>”, since I am using windows architecture on my computer.

A new environment had to be created in Anaconda Navigator to install OpenCV packages with an older version of Python 3.6. Since the latest 3.7 version was not supporting. Then OpenCV had to be activated in the Anaconda Prompt, which was creating a lot of issues.

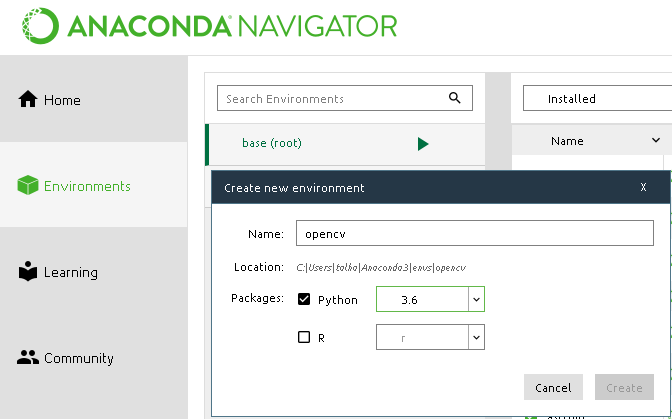


Figure 6 - New environment created in Anaconda Navigator

The following command is typed in the Anaconda prompt to install the OpenCV.

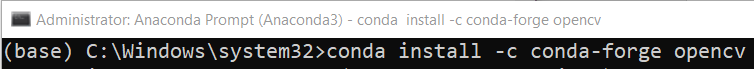


Figure 7 - Command for OpenCV installation in Anaconda prompt

This process took too long to install those packages due to the internet speed and it was not resolving the installation problems, because installing OpenCV manually was a difficult task since it was taken one and half days. A simple way was to install OpenCV was through Anaconda Navigator, it provides with OpenCV option to install it.

### OpenCV Installation with Anaconda Navigator

A simple way to install OpenCV was with the Anaconda Navigation. OpenCV was typed in the search bar and all library option was selected and there were three main libraries available for OpenCV to install “libopencv”, “opencv” and “py-opencv” as shown in figure 10. It does not take much time as compared to manually installed.

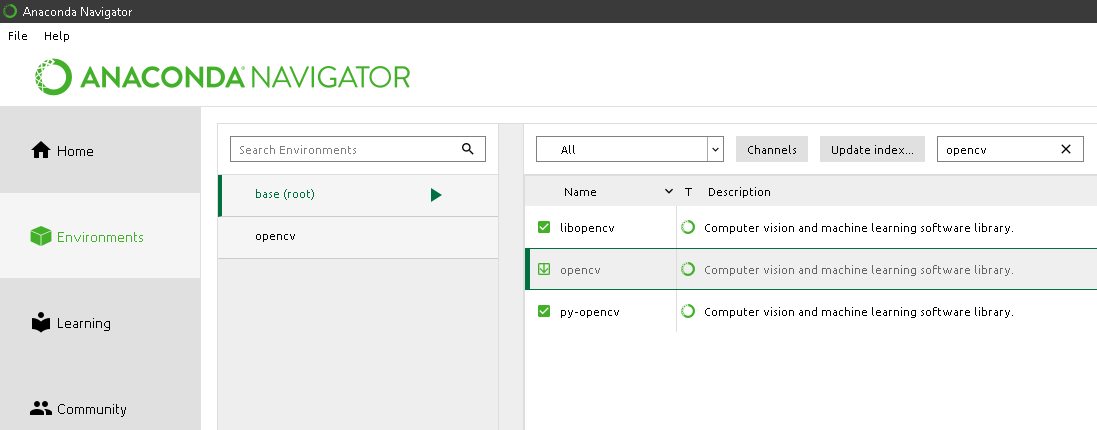


Figure 8 - Available OpenCV library to install in Anaconda Navigator

The OpenCV was imported in the Python console in ‘Spyder’ to check whether the “OpenCV” library was installed properly or not. Figure 11 shows the OpenCV was instated properly without an error.

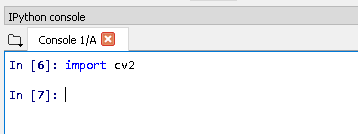


Figure 9 - Importing 'OpenCV' library to check, if it's installed

The window doesn’t show an error which means that the libraries were installed properly as shown in figure 9.

## Haars Cascade algorithm

## DetectionMultiscale

### Os.walk

# Test procedures and results

# Conclusion

# References