الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــة

Faculté : TECHNOLOGIE

Département : ELECTRONIQUE

Domaine : SCIENCES ET TECHNIQUES

Filière : Electronique

Spécialité : Electronique des systèmes embarqués.

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Réalisation d'un système de sécurité utilisant la reconnaissance faciale avec Raspberry pi

Présenté par : *Kriba Riad*

Encadrant : *Yahi Amira* *Grade* *MCB* *Université ANNABA*

Jury de Soutenance :

AMARA Fethi	MCB	U.ANNABA	Président
YAHI Amira	MCB	U.ANNABA	Encadrant
ZERMI Narima	MCA	U.ANNABA	Examinateur

Année Universitaire : 2021/2022

Dedications

I dedicate this end of studies report to my dear parents;two exceptional people who, through their dedication, patience and support allowed me to get where I am.

I also dedicate it to my closest friends, associates and strangers i met during this journey that shaped me to become what i am here and now

My sincère Gratitude

As a preamble to this dissertation, which is the culmination of long and tedious years of university studies, I thank ALLAH for everything.

Many people have contributed scientifically, intellectually or technically to the reduction of this memoir. Also, I would like to thank Dr. Yahi Amira, my internship tutor who trained and supported me throughout this professional experience with a lot of patience and pedagogy.

To all the people who participated in this project from near or far in sincerely thank you

Finally, my thanks go to all the members of the jury : Dr. Zermi and Dr. Amara.

Abstract:

This project consists of implementing a real time facial recognition security technologies set up with limited resources hardware like raspberry pi 4 based on artificial intelligence, where we compare the real time application of both HOG and MMOD (CNN) facial recognition based models algorithms while using mediapipe' revolutionary detection algorithm, a lightweight and well-performing face detector . In order to maximize the limited resources while showing the results in real time with appropriate action the whole project will be written in a robust python environment.

Keywords: artificial intelligence ; MMOD(CNN); python; raspberry pi; real time face detection and recognition, biometric security ; mediapipe

Résumé:

Ce projet consiste à mettre en œuvre des technologies de sécurité de reconnaissance faciale en temps réel configurées avec du matériel à ressources limitées comme le raspberry pi 4 basé sur l'intelligence artificielle, où nous comparons l'application en temps réel des algorithmes de modèles basés sur la reconnaissance faciale HOG et MMOD (CNN) tout en utilisant mediapipe : un algorithme de détection révolutionnaire et un détecteur de visage (léger) et performant. Afin de maximiser les ressources limitées tout en montrant les résultats d'exécution en temps réel, l'ensemble du projet sera écrit dans un environnement python robuste.

Mots clés : intelligence artificielle ; MMOD(CNN); python; Raspberry pi; détection et reconnaissance faciale en temps réel, sécurité biométrique ; mediapipe

الملخص

يتكون هذا المشروع من تنفيذ تقنيات أمان للتعرف على الوجه في الوقت الفعلي تم إعدادها باستخدام أجهزة ذات موارد محدودة مثل raspberry pi 4 استنادًا إلى الذكاء الاصطناعي ، حيث نقارن التطبيق في الوقت الفعلي لكل من خوارزميات نماذج التعرف على الوجه HOG و MMOD (CNN) أثناء استخدام mediapipe خوارزمية الكشف الثورية ، جهاز كشف الوجه خفيف الوزن وذو أداء جيد. من أجل تعظيم الموارد المحدودة مع إظهار النتائج في الوقت الفعلي مع الإجراء المناسب ، سيتم كتابة المشروع بأكمله في بيئة python قوية.

الكلمات المفتاحية: ذكاء اصطناعي؛ MMOD (CNN) ؛ Raspberry pi .python؛ كشف الوجه والتعرف عليه في الوقت الحقيقي ، والأمن البيومتري ؛ ميديابيب.

Abbreviations list

NIST :National Institute of Standards and Technology
ATM : Automated teller machine(a machine, usually outside a bank, which customers can use to get money)
IoT : Internet of Things
ID : Identification
2FA : Two factor authentication
MFA : Multi factor authentication
UK: United Kingdoms
FRT: Facial Recognition Technology
ANNs : Artificial neuron network
SVM : support vector machine
OpenCV:Open Source Computer Vision Library
HOG : Histogram of oriented gradients
CNN : Convolutional neural network
DNN : Deep neural network
KNN : k-nearest neighbors algorithm
MMOD : Max-Margin Object Detection
MTCNN : Multi-Task Cascaded Convolutional Neural Networks
CPU : Central processing unit
GPU : Graphics processing unit
PC : Personal Computer
USB :Universal Serial Bus
SD :Secure Digital
OS : Operating System
SSD: Single Shot MultiBox Detector
3D : three-dimensional
VS code : Visual Studio Code
LSVM: Linear Support Vector Machine
FPS: frame per second
Res: resolution

SUMMARY

CHAPTER III : Applying The Modern Facial Recognition Pipeline

Chapter IV:Applying and Comparing Different modern Technologies on the raspberry 4 Embedded system

Figures list

Tables list

General Introduction

Face recognition technology is a biometric technology based on the automatic identification of the unique facial features of a person, for the purposes of identification or authentication. Once considered a thing of science fiction, biometric Facial Recognition is quickly becoming an integrated part of people's everyday lives. Several major industries have benefited from the rapid advancements that have been made in Facial Recognition technology over the past 60 years and these include law enforcement, border control, retail, mobile technology and banking and finance.

Often leveraging a digital or connected camera, facial recognition software can detect faces in images, quantify their features, and then match them against stored templates in a database. Face-scanning biometric tech is incredibly versatile and this is reflected in its wide range of potential applications.

In this project, we implement a real time facial recognition using two algorithms model : HOG and CNN, we use mediapipe revolutionary lightweight face detection algorithm, the project is written in robust python environment and tested using Raspberry pi 4 technology.

The manuscript is organized as follow:
Chapter 1 provides an introduction to biometrics security and its properties to have a good secure system. Chapter 2 discusses facial recognition with some of the most used techniques. Chapter 3 details the steps followed for the implementation of a modern facial recognition system based on both HOG and CNN algorithm, chapter 4 shows the tests and the comparisons were made using the raspberry pi 4 board. We finish with a general conclusion and perspectives.

Chapter I: biometrics and its role in security

I.1 Introduction:

Ten years ago, biometrics seemed like something out of a science fiction movie. Fast forward to now, and people everywhere are unlocking their phones with their faces. And the appetite for continued use of biometrics is significant — 86% of consumers want to use biometrics to verify their identity. With biometric technology, people no longer have to worry about memorizing lengthy passwords, and the chances of them ever losing their face or fingerprints is next to none.

New advances have made some biometrics approaches safer and more secure, but these techniques are more costly to implement. So, how do we address these issues moving forward? This piece will explore the technologies behind facial recognition biometrics and adapting it on a raspberry pi embedded system.

I.2 A Brief History in Facial Biometrics:

As we look forward to the future uses of Facial Recognition software, it's good to take a step back and see how far we have come since the early beginnings.

First steps in the facial recognition biometrics field : An early cataloging of fingerprints dates back to 1885 when Juan Vucetich started a collection of fingerprints of criminals in Argentina.[1]Josh Ellenbogen and Nitzan Lebovic argued that Biometrics originated in the identification systems of criminal activity developed by Alphonse Bertillon (1853–1914) and by Francis Galton's theory of fingerprints and physiognomy.[2] According to Lebovic, Galton's work "led to the application of mathematical models to fingerprints, phrenology, and facial characteristics".[3] Accordingly, "the biometric system is the absolute political weapon of our era" and a form of "soft control".[4]

The dawn of Facial Recognition – 1960s : The earliest pioneers of facial recognition were Woody Bledsoe, Helen Chan Wolf and Charles Bisson. In 1964 and 1965 they began work using computers to recognise the human face. Their initial work involved the manual marking of various "landmarks" on the face such as eye centers, mouth etc. These were then mathematically rotated by a computer to compensate for pose variation. The distances between landmarks were also automatically computed and compared between images to determine identity. Altho severely hampered by the technology of the era, it remains an important first step in proving that Facial Recognition was a viable biometric.

FERET Programme – 1990s/2000s : The Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST) rolled out the Face Recognition Technology (FERET) programme in the early 1990s in order to encourage the commercial facial recognition market.

Kelly A. Gates identified 9/11 as the turning point for the cultural language of our present and a new discourse formation is established: automated facial recognition as a homeland security technology."[5]

Social Media – 2010-Current : Facebook began implementing facial recognition functionality that helped identify people whose faces may feature in the photos that Facebook users update daily. The feature was instantly controversial with the news media, sparking a slew of privacy-related articles. However, Facebook users by and large did not seem to mind.

iPhone X – 2017 : Facial Recognition technology advanced rapidly from 2010 onwards and September 12, 2017, was another significant breakthrough for the integration of facial recognition into our day-to-day lives. This was the date that Apple launched the iPhone X – the first iPhone users could unlock with FaceID – Apple's marketing term for facial recognition.

The future of Facial Recognition Technology : As we move into 2022, facial recognition technology continues to develop at pace and the uses of the technology are becoming more widespread. From Pay by Face to solving issues caused by the increased use of facemasks in light of the COVID-19 pandemic.

I.3 What are Biometrics?

Biometrics (ancient Greek: bios ="life", metron ="measure") refers to two very different fields of study and application. The first is the collection, synthesis, analysis and management of quantitative data on biological communities such as forests.[6]

A concise modern definition of biometrics is "the automatic recognition of a person using distinguishing measurable, robust and distinctive physical characteristic or personal trait that can be used to identify an individual or verify the claimed identity of an individual."

Measurable: means that the characteristic or trait can be easily presented to a sensor, located by it, and converted into a quantifiable, digital format. This measurability allows for matching to occur in a matter of seconds and makes it an automated process.

The robustness: It refers to the extent to which the characteristic or trait is subject to significant changes over time. These changes can occur as a result of age, injury, illness, occupational use, or chemical exposure. A highly robust biometric does not change significantly over time while a less robust biometrics will change. For example, the iris, which changes very little over a person's lifetime, is more robust than one's voice.

Distinctiveness : is a measure of the variations or differences in the biometric pattern among the general population. The higher the degree of distinctiveness, the more individual the identifier is. A low degree of distinctiveness indicates a biometric pattern found frequently in the general population. The iris and the retina have higher degrees of distinctiveness than hand or finger geometry.[7]

I.4 Biometrics's Role in Security

Biometric technique is now becoming the foundation of a wide array of highly secure identification and personal verification. As the level of security breach and transaction scam increases, the need for well secure identification and personal verification technologies is becoming apparent. Recent world events have led to an increased interest in security that will impel biometrics into majority use. Areas of future use contain Internet transactions, workstation and network access, telephone transactions and in travel and tourism.

There are different types of biometrics. The most recognized biometric technologies are fingerprinting, retinal scanning, hand geometry, signature verification, voice recognition, iris scanning and facial recognition A biometric system can be either an 'identification' system or a 'verification' (authentication) system, which are defined below. Identification (1: n) – One-to-Many: Biometrics can be used to determine a person's identity even without his awareness or approval. Such as scanning a crowd with the help of a camera and using face recognition technology, one can verify matches that are already stored in a database. Verification (1:1) One-to-One: Biometrics can also be used to verify a person's identity. Such as one can allow physical access to a secure area in a building by using finger scans or can grant access to a bank account at an ATM by using retina scan.

I.5 How do Biometrics Work?

Physical characteristics are relatively fixed and individualized — even in the case of twins. Each person's unique biometric identity can be used to replace or at least augment password systems for computers, phones, and restricted access rooms and buildings. Once biometric data is obtained and mapped, it is then saved to be matched with future attempts at access. Most of the time, this data is encrypted and stored within the device or in a remote server. Biometrics scanners are hardware used to capture the biometric for verification of identity. These scans match against the saved database to approve or deny access to the system.In other words, biometric security means your body becomes the "key" to unlock your access.

Biometrics are largely used because of two major benefits. In addition to security, the driving force behind biometric verification has been convenience, as there are no passwords to remember or security tokens to carry that cannot be easily lost or forgotten.Difficult to steal or impersonate . Some biometric methods, such as measuring a person's gait, can operate with no direct contact with the person being authenticated.While these systems are not perfect, they offer tons of promise for the future of cybersecurity.

I.6 Authentication

Authentication is the process of validating the user's identity. Users are identified using different authentication mechanisms. In a security system the authentication process checks the information provided by the user with the database[8][9]. If the information matches with the database information, the user is granted access to the security system.

There are three types of authentication mechanisms used. Validation is the initial phase in access control, and there are three regular variables utilized for verification – something you know, something you have, and something you are [8].Something you know mostly requires an individual to get access to the system by typing the username and password. Something you have is where the user uses a smart card for authentications [8] [9]. Where you are is where the user is using biometrics methods to get access control. All types of authentication mechanisms allow users to get access to the system however they all work differently.

I.7 The Uses of Biometrics

Biometrics make a good replacement for usernames as part of a two-factor authentication strategy.Two-factor authentication makes a powerful combination, especially as IoT devices proliferate. By layering the protection, secured internet devices become less vulnerable to data breaches.

Advanced biometrics are used to protect sensitive documents and valuables. Citibank already uses voice recognition, and the British bank Halifax is testing devices that monitor heartbeat to verify customers' identities. Ford is even considering putting biometric sensors in cars.

Biometrics are incorporated in e-Passports throughout the world. In the United States, e-passports have a chip that contains a digital photograph of one’s face, fingerprint, or iris, as well as technology that prevents the chip from being read — and the data skimmed — by unauthorized data readers.

I.8 Types of Biometrics

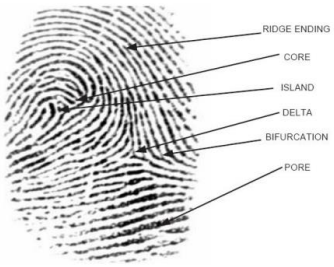
Based on the above guidelines, several biometrics are being developed and are in use This paper describes popularly used biometrics in terms of the character measured , the devices used to collect the biometric , features extracted , the algorithms used and the areas of applicability.

Figure I.1 : Biometrics authentication Infographic

FACE RECOGNITION : Face appearance is a biometric which is used everyday by everyone as a primary means of recognizing other humans[10]. Because of this naturalness it is more acceptable than other biometrics.

Figure I.2 : Face recognition sketch

FINGERPRINT BIOMETRIC METHOD : popular due to its ease in acquisition, number of sources (ten fingers) and their acceptance over a long period of time by law enforcement offices. Fingerprints form part of an individual's phenotype and are not determined by genetics and hence qualify as good biometrics.[11]



FigureI.3 : Fingerprints phenotype

HAND RECOGNITION : the geometric features of the hand such as the lengths of fingers and the width of the hand and other distinct features and the hand's overall bone structure are recorded. In the verification phase, the user is prompted to place his/her hand only once on the platen.This technology is mostly used in physical access entry applications. [12]

Figure I.4 : Hand model

Vein scan biometric : It identifies a person from the patterns of the blood vessels in the back of the hand. The technology uses near infrared light to detect vein vessel patterns. Vein patterns are distinctive between twins and even between a person's left and right hand. These are developed before birth, highly stable and change through one's life only in overall size. The technology is not intrusive, and works even if the hand is not clean. It is commercially available and implemented by Fijitsu of Japan.[13]

IRIS RECOGNITION : The colored part of the eye between the pupil and sclera is called the Iris. Since Iris is a protected internal organ with a complex texture , and because it is unique from person to person and stable throughout life, it forms a very good biometric. Iris image acquisition is done in two ways: Daugman System and Wilde's system.

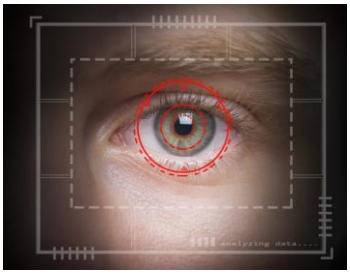


Figure I.5 : Eye scan

Retina scan : It analyzes the layer of blood vessels located at the back of the eye. quite accurate and very unique to each individual similar to the Iris scan; it requires the user to look into a receptacle and focus on a given point.very accurate for use in identification but susceptible to retina diseases such careacts , [14].[15]

Facial thermography : Facial thermography detects heat patterns created by the branching of blood vessels and emitted from the skin. Unlike visible light systems, infrared systems work accurately even in dim light or total darkness. Although identification systems using facial thermograms were undertaken in 1997, the effort was suspended because of the cost of manufacturing the system.

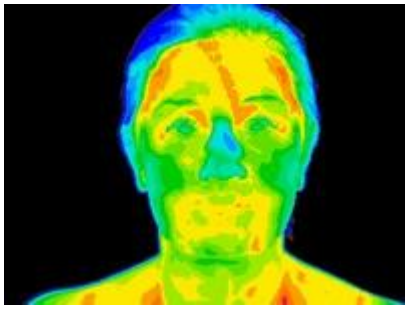


Figure I.6 : Facial thermography

Skin pattern : The exact composition of all the skin elements such as skin layer thickness, undulations between layers , pigmentation, collagen fibers etc is distinctive to each person. Skin pattern recognition technology measures the characteristic spectrum of an individual's skin. A light sensor illuminates a small patch of skin with a beam of visible and near-infrared light.

Voice : Speaker recognition or voice authentication uses a microphone to record and digitalize the voice of a person.. 'The speech can be acquired from the user enunciating a known speaking (text independent) or text (text dependent)'.it has low system cost ; compact and easy to use. The other side is that the voice varies with age and there can be drastic change from childhood to adolescence. Moreover emotions and illness may affect the voice as well as room acoustics and environmental noise.

I.9 Biometric Traits

Faundez-Zanuy (2006) identified that biometric traits can be split into two main categories:

-Physiological Biometrics : It is based on direct measurements of a part of the human body like fingerprint, face, iris, and hand-scan recognition.

-Behavioral Biometrics : It is based on measurements and data derived from an action performed by the user, and thus indirectly measures some characteristics of the human body like Signature and key stroking recognition.

Faundez-Zanuy (2006) explains that a good biometric trait must accomplish the following set of properties.:

-Universality : Every person should have this characteristic.
-Distinctiveness : Also referred to as uniqueness, Any two persons should be different enough to distinguish them from each other based on this characteristic.
-Permanence : Characteristics should be stable enough (with respect to the matching criterion) along time, different environment conditions, etc.

-Collectability : Characteristics should be acquirable and quantitatively measurable without waiting time and must be easy to gather passively.
-Comparability and Reducibility : reducible to a state that makes it digitally comparable from others. It has less probability
-Acceptability : People should be willing to accept the biometric system, and not feel that it is annoying, invasive, etc.
-Performance : Identification accuracy and required time for a successful recognition must be reasonably good.
-Privacy: This process should not break the privacy of the individual.
-Circumvention : Ability of fraudulent people and techniques to fool the biometric system should be negligible.
[16]

Characteristics	**Fingerprints**	**Hand Geometry**	**Retina**	**Iris**	**Face**	**Signature**	**Voice**
Ease of Use	High	High	Low	Medium	Medium	High	High
Error Incidence	Dryness,dirt, âge	Hand injury,age	Glasses	Lighting	Lighting,age, glasses;hair	Changing signature	Noises,colds
Accuracy	High	High	Very high	Very high	High	High	High
User Acceptance	Medium	Medium	Medium	Medium	Medium	High	High
Long Term Stability	High	Medium	High	High	Medium	Medium	Medium

Table I.1 : Table represents the classification of characteristic features of biometric technology.[16]

I.10 Issues and concerns

Biometric authentication is convenient, but privacy advocates fear that biometric security erodes personal privacy. The concern is that personal data could be collected easily and without consent.

I.10.1 Biometric Data Security Concerns

A more immediate problem is that databases of personal information are targets for hackers. For example, when the U.S. The Office of Personnel Management was hacked in 2015, cybercriminals made off with the fingerprints of 5.6 million government employees, leaving them vulnerable to identity theft.

India's Unique ID Authority of India Aadhaar program is a good example. Initiated in 2009, the multi-step authentication program incorporates iris scans, fingerprints from all 10 fingers, and facial recognition.This information is linked to a unique identification card that is issued to each of India's 1.2 billion residents. Soon, this card will be mandatory for anyone accessing social services in India.

Human dignity Biometrics have been considered also instrumental to the development of state authority (to put it in Foucauldian terms, of discipline and biopower[17]). By turning the human subject into a collection of biometric parameters, biometrics would dehumanize the person, infringe bodily integrity, and, ultimately, offend human dignity.

Other scholars[18] have emphasized, however, that the globalized world is confronted with a huge mass of people with weak or absent civil identities. Most developing countries have weak and unreliable documents and the poorer people in these countries do not have even those unreliable documents.[19] Without certified personal identities, there is no certainty of right, no civil liberty.[20] One can claim his rights, including the right to refuse to be identified, only if he is an identifiable subject, if he has a public identity. In such a sense, biometrics could play a pivotal role in supporting and promoting respect for human dignity and fundamental rights.[21]

I.10.2 Privacy and Discrimination

Further information: privacy, right to privacy, and medical privacy. It is possible that data obtained during biometric enrollment may be used in ways for which the enrolled individual has not consented. For example, most biometric features could disclose physiological and/or pathological medical conditions (e.g., some fingerprint patterns are related to chromosomal diseases, iris patterns could reveal sex, hand vein patterns could reveal vascular diseases, most behavioral biometrics could reveal neurological diseases, etc.).[22] Moreover, second generation
biometrics, notably behavioral and electro-physiologic biometrics (e.g., based on electrocardiography, electroencephalography, electromyography), could also be used for emotion detection.[23]

I.10.3 The Need to Protect Biometric Identity

With the risks to privacy and safety, additional protections must be used in biometric systems,so that Unauthorized access becomes more difficult when systems require multiple means of authentication, such as life detection (like blinking) and matching encoded samples to users within encrypted domains.Some security systems also include additional features, such as age, gender, and height, in biometric data to thwart hackers.

I.11 Conclusion

In this chapter we have given definitions of biometrics, how it works, its context of use, as well as its different types used for security, the goal is to extract the parameters: features and aspects to follow in order to have a good biometrics identification.

Chapter II : Facial Recognition

II.1 Introduction

Formally, facial or face recognition is defined as “the science which involves the understanding of how the faces are recognized by biological systems and how this can be emulated by computer systems” (Martinez 2009). It is a biometric technique to uniquely identify a person by comparing and analyzing patterns based on their "facial contours"; in other words, it is a method of identifying or confirming a person's identity from their face. Other types of biometric software include voice recognition, fingerprint recognition, and retinal or iris recognition. This technology is primarily used for security and law enforcement purposes, but it is becoming increasingly popular in other areas as well.

II.2 Common Uses of Face Recognition Technology

This technology is used by many companies and organizations, some that you’re probably aware of, and there are some that you’re possibly not. Here are some examples of Face Recognition Technology:

-Access Control : Access control of personal computers, homes, cars, offices, and other premises is one of the most apparent methods of using Face Recognition. And Apple’s iPhone X is a perfect example of using FRT to unlock a smartphone.

-Shopping Online: Alibaba, a prominent Chinese e-commerce company, plans to use the Alipay platform to let users make purchases over the Internet. And as a first step, Alipay has already launched a ‘Smile to Pay’ facial recognition system at a KFC in Hangzhou. The system recognizes a face within two seconds and then verifies the scan by sending a mobile alert. ‘Smile to Pay’ is also able to identify people wearing make-up or wigs as a disguise.

-Helping addictive gamblers: The face recognition system merely compares the faces of individuals who play slots with self-proclaimed problem gamblers in casinos. It alerts the security team when the device detects a match, which then discreetly approaches the gamblers and escorts them off the premises.

-Tracking down criminals : And this one won’t come as a surprise. Facial recognition is a crime-fighting technology that is used to recognize targets by law enforcement and intelligence agencies. For example, the officer only has to snap a picture and voila with the assistance of MORIIS (Mobile Offender Identification and Information System)-a portable biometric device attached to a smartphone.

-Organizing photos : The most widespread way to use this technology is done by Apple, Google, and even Facebook to differentiate a portrait from a landscape, find a user in a frame, and sort photos by categories using their own face recognition systems. And we all provide tremendous support for the facial recognition algorithm every time we upload a picture and tag our friends on it.

-Taking attendance in school : Schools in the UK use FRT in order to attend. This has been going on for a while in the UK, but will definitely spread to other nations as well. Both students and teachers in the UK love this new technology that scans faces with infra-red light and matches them with archived images.

II.3 Artificial Neural Network

Artificial neural networks (ANNs), usually simply called neural networks (NNs) or, more simply yet, neural nets, are computing systems inspired by the biological neural networks that constitute animal brains.

II.3.1 Definition

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron receives signals then processes them and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times

.

II.3.2 Training a Neural Network Algorithm

Neural networks learn (or are trained) by processing examples, each of which contains a known "input" and "result," forming probability-weighted associations between the two, which are stored within the data structure of the net itself. The training of a neural network from a given example is usually conducted by determining the difference between the processed output of the network (often a prediction) and a target output. This difference is the error. The network then adjusts its weighted associations according to a learning rule and using this error value. Successive adjustments will cause the neural network to produce output which is increasingly similar to the target output. After a sufficient number of these adjustments the training can be terminated based upon certain criteria. This is known as supervised learning.

Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers, and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process.

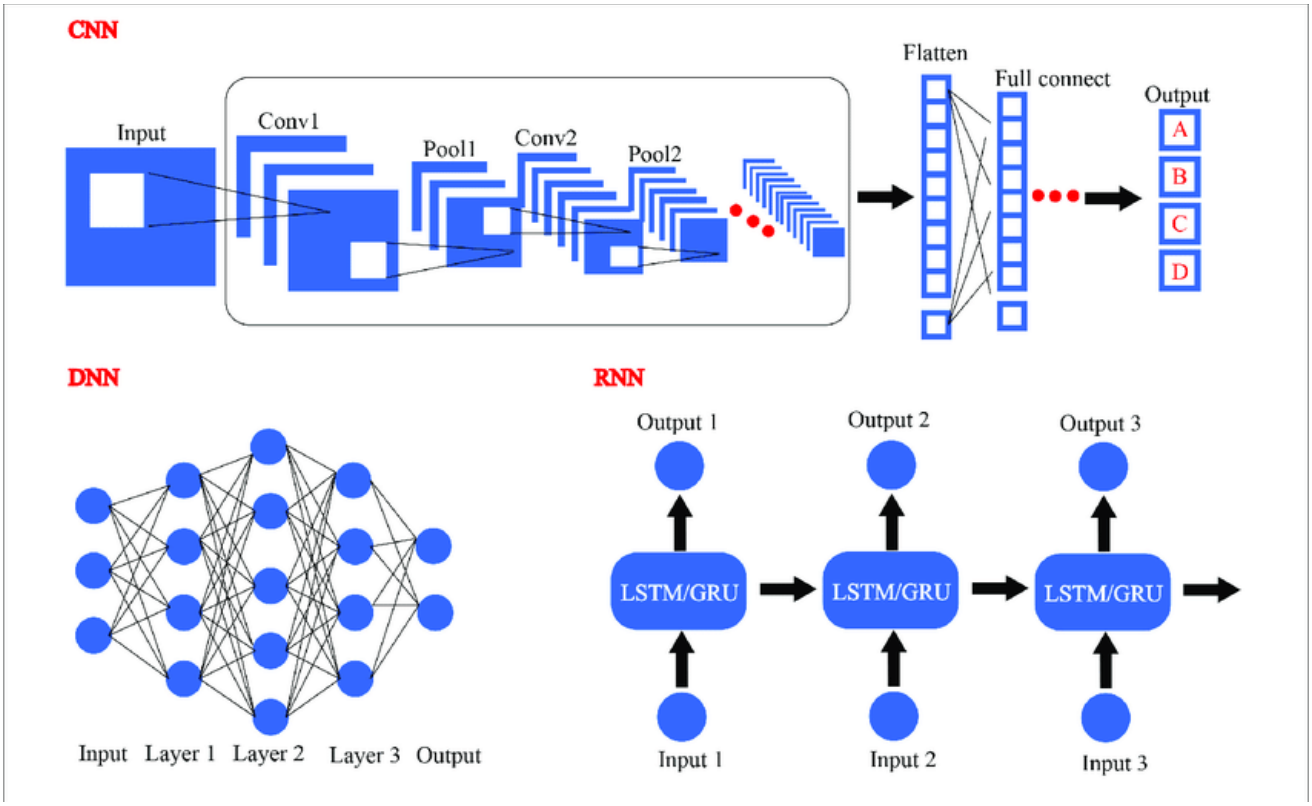


Figure II.1 : several different neural network frameworks

II.4 The Modern Facial Recognition Pipeline

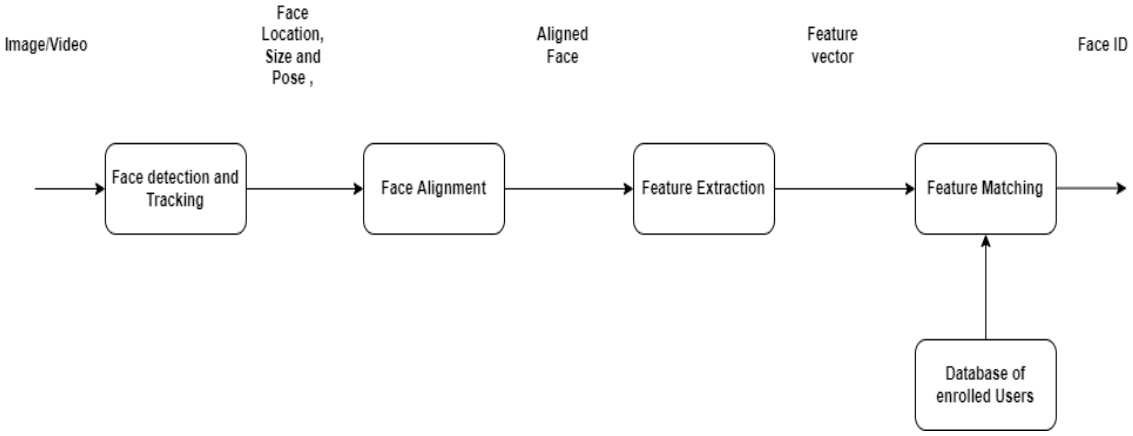


Figure II.2: Figure representing the modern facial recognition pipeline steps

II.4.1 Face Detection

A Face detection method is used to find the faces present in the given image, extract faces if they exist, and crop the face only to create a compressed file for further feature extraction. There are multiple algorithm options to perform this task in a face detection/recognition system.

II.4.1.1 Methods used in Face Detection

A) Haar cascade Face Detection : The Haar Cascade based Face Detector has been the state-of-the-art in Face Detection for many years since 2001 when it was introduced by Viola and Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features". There have been many improvements in recent years. This method has a simple architecture that works nearly real-time on the CPU. Also, it can detect images at different scales. But the major drawback is that it gives false results as well as it doesn't work on non-frontal images.Haar Cascade based Face Detector has been the state-of-the-art in Face Detection for many years since 2001, when it was introduced by Viola and Jones. There have been many improvements in recent years.

Pros	Cons
-Simple Architecture -Works almost real-time on CPU -Detects faces at different scales	-a lot of False predictions. -Doesn't work under occlusion

Table II.1 : Table represents Pros and Cons of Haar cascade classifiers

B) Dlib (HOG) Face Detection : This is a widely used face detection model, based on HoG features and SVM published in 2005 in the paper "Histograms of oriented gradients for human detection". HOG, or Histogram of Oriented Gradients, is a feature descriptor that is often used to extract features from image data. It is the fastest method on the CPU which can work on frontal and slightly no-frontal images. But it is incapable of detecting small images and handling occlusions. Also, it often excludes some parts of the chin and forehead while detection.

This is a widely used face detection model, based on HoG features and SVM. You can read more about HoG in our post. The model is built out of 5 HOG filters – front looking, left looking, right looking, front looking but rotated left, and a front looking but rotated right. The model comes embedded in the header file itself.

Pros	Cons
-Fastest method on CPU -Works very well for frontal and slightly non-frontal faces -Light-weight model as compared to the other three. Works under small occlusion -Basically, this method works in most cases except a few as discussed below.	-The major drawback is that it does not detect small faces as it is trained for a minimum face size of 80×80. Thus, you need to make sure that the face size should be more than that in your application. You can, however, train your own face detector for smaller sized faces. -The bounding box often excludes part of the forehead and even part of the chin sometimes. Does not work very well under substantial occlusion Does not work for side faces and extreme non-frontal faces, like looking down or up.

Table II.2 : Table represents Pros and Cons of Dlib (HOG) Face Detection

C) Dlib (CNN) Face Detection : This method first introduced in the 2016 paper "CNN based efficient face recognition technique using Dlib" uses a Maximum-Margin Object Detector (MMOD) with CNN based features. The training process for this method is very simple and you don't need a large amount of data to train a custom object detector. It works very fast on the GPU and is capable of working for various face orientations in images. It can also handle occlusions. But the major disadvantage is that it is trained on a minimum face size of 80*80 so it can't detect small faces in images. It is also very slow on the CPU.

This method uses a Maximum-Margin Object Detector (MMOD) with CNN based features. The training process for this method is very simple and you don't need a large amount of data to train a custom object detector.

Pros	Cons
Works for different face orientations Robust to occlusion Works very fast on GPU Very easy training process	Very slow on CPU Does not detect small faces as it is trained for a minimum face size of 80×80. Thus, you need to make sure that the face size should be more than that in your application. You can, however, train your own face detector for smaller sized faces. The bounding box is even smaller than the HoG detector.

Table II.3 : Table represents Pros and Cons of Dlib (CNN) Face Detection

D) DNN Face Detector in OpenCV : This model was included in OpenCV from version 3.3. It is based on the Single-Shot-Multibox detector and uses ResNet-10 Architecture as its backbone. The model was trained using images available from the web, but the source is not disclosed. OpenCV provides 2 models for this face detector.

-Floating point 16 version of the original caffe implementation (5.4 MB)
-8 bit quantized version using Tensorflow (2.7 MB)

Pros	Cons
One of Most accurate methods Runs at real-time on CPU Works for different face orientations – up, down, left, right, side-face etc. Works even under substantial occlusion Detects faces across various scales (detects big as well as tiny faces)	The DNN-based detector overcomes all the drawbacks of the Haar cascade-based detector, without compromising on any benefit provided by Haar. almost major drawback for this method except that it is slower than the Dlib HoG based Face Detector

Table II.4 : Table represents Pros and Cons of DNN Face Detector in OpenCV

II.4.1.2 Limitations to Consider in Cases of Use

General Case : In most applications, we won't know the size of the face in the image before-hand. Thus, it is better to use OpenCV – DNN method as it is pretty fast and very accurate, even for small sized faces. It also detects faces at various angles, OpenCV-DNN is most recommended.

For medium to large image sizes : Dlib HoG is the fastest method on the CPU. But it does not detect small sized faces (< 70×70). So, if you know that your application will not be dealing with very small sized faces (for example a selfie app), then a HoG based Face detector is a better option. Also, If you can use a GPU, then MMOD face detector is the best option as it is very fast on GPU and also provides detection at various angles.

High resolution images : Since feeding high resolution images is not possible to these algorithms (for computation speed), HoG / MMOD detectors might fail when you scale down the image. On the other hand, the OpenCV-DNN method can be used for these since it detects small faces.

II.4.2 Face Alignment

Face alignment is an early phase of the modern pipeline of face recognition. Google has reported that face alignment improves the accuracy of its FaceNet face recognition model from 98.87% to 99.63%. This is an increase in accuracy of almost 1 percent. We can easily apply 2D face alignment inside OpenCV in Python. With haar cascade configurations, OpenCV has modules for both frontal face and eye detection. Extracting the eye locations is very important to align faces. OpenCV finds eye locations with a conventional haar cascade method. After getting the eye location of the detected face you can rotate the image 1 degree until both eyes are horizontal. This will increase the complexity of the solution so you can align the face based on angles between two eyes using the cosine rule. MTCNN also finds some facial landmarks such as eye, nose, and mouth locations. If we are using MTCNN in the face recognition pipeline it will automatically do an alignment to the face detected.

II.4.3 Feature Extraction

Feature extraction is the basic and most important initializing step for face recognition. It extracts the biological components of your face. These biological components are the features of your face that differ from person to person. There are various methods which extract various combinations of features, commonly known as nodal points. No two people can have all the nodal points similar to each other except for identical twins.

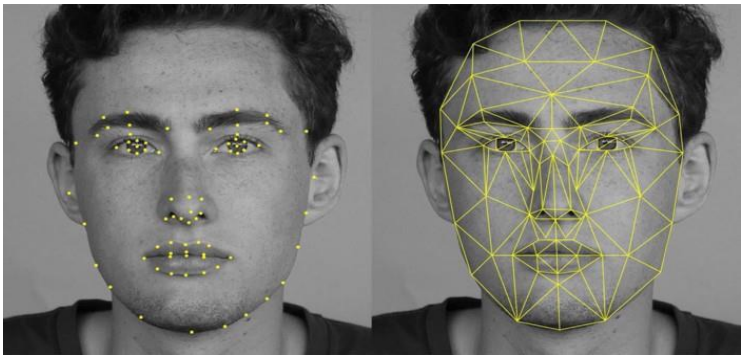


Figure II.3 : 128 Face Embeddings

II.3.4.1 Methods used in Feature Extraction (Deep Approach)

There was a flurry of research and publications in 2014 and 2015 on deep learning methods for face recognition. Capabilities reached near-human-level performance rapidly, then surpassed human-level performance over a three-year span on a standard face recognition dataset. So these advances have been powered by four milestone systems for deep learning for face recognition: DeepFace, the DeepID series of systems, VGGFace, and FaceNet.

A) VGGFace : The VGG-Face CNN descriptors are computed using our CNN implementation based on the VGG-Very-Deep-16 CNN architecture and are evaluated on the Labeled Faces in the Wild and the YouTube Faces dataset. VGG uses various architectures such as VGGFace1, VGGFace2 by Keras. The basic difference among these models is the number of layers included in its architecture that varies from model to model. These models have quite good accuracy.

B) FaceNet : FaceNet is a face recognition system developed in 2015 by researchers at Google in their 2015 paper titled "FaceNet: A Unified Embedding for Face Recognition and Clustering", that achieved then state-of-the-art results on a range of face recognition benchmark datasets and presented an innovation called 'triplet loss' that allowed images to be encoded efficiently as feature vectors that allowed rapid similarity calculation and matching via distance calculations. The FaceNet system can be used broadly thanks to multiple third-party open-source implementations of the model and the availability of pre-trained models. The FaceNet system can be used to extract high-quality features from faces, called face embeddings, which can then be used to train a face identification system.

C) DeepFace : DeepFace is a system based on deep convolutional neural networks. It was described in the 2014 paper titled "DeepFace: Closing the Gap to Human-Level Performance in Face Verification." It was perhaps the first major leap forward using deep learning for face recognition, achieving near human-level performance on a standard benchmark dataset.

D) DeepID (Deep hidden IDentity features) : The DeepID is a series of systems (e.g. DeepID, DeepID2, etc.), first described by Yi Sun, et al. in their 2014 paper titled "Deep Learning Face Representation from Predicting 10,000 Classes." Their system was first described much like DeepFace, although it was expanded in subsequent publications to support both identification and verification tasks by training via contrastive loss.

The DeepID systems were among the first deep learning models to achieve better-than-human performance on the task.

II.4.4 Feature Classification

The final stage of face detection technology is to make a decision whether the face's features of a new sample are matching with the one from a facial database or not. These template-based classifications are possible using various statistical approaches. It usually takes just seconds.

A) Euclidean Distance : It is a distance-based feature classification method that calculates the distance between the facial nodes and the face which has the minimum difference between these distance values and is considered to be the match. But it is suitable for datasets having a smaller number of classes and lower-dimensional features.
Cosine Similarity: In cosine similarity, the solution that we obtain after calculating the cosine of an angle is brought into concern. Here, we would compare the differences between these results. The more the value is to 1, the greater is the probability of the match. But it may give a false result if the test data features are incomplete.

B) SVM (Support vector machine) : SVM creates an optimal hyperplane to classify the classes of the training dataset based on the different features of the face. The dimensionality of the hyperplane is one less than the number of features. Different kernels can be applied to see what features are used by the classifier to remove the features if required. This can help to improve speed

C) KNN (K-Nearest Neighbor) : KNN is all about the number of neighbors i.e. the k value. In KNN, if k=3 then we check that the data is close to the 3 data points. Thereafter, it is decided that the majority of the closest data points belong to which class. Now, the test data is predicted to be in this class. KNN has a curse of dimensionality problem which can be solved by applying PCA before using the KNN classifier. You can get a better understanding of KNN

D) ANN (Artificial Neural Network) : ANN uses a very detailed algorithm for face recognition. It classifies the local texture using a multi-layer perceptron for face alignment. It uses a geometric feature-based and independent component analysis for feature extraction and a multi-artificial neural network for feature matching.

II.5 Conclusion

In this chapter, we have focused on the facial recognition part, for this we have defined the different algorithms used, namely the artificial neural network. then, we defined the most modern facial recognition model that works in pipeline and we detailed its different stages of operation. In the next chapter we will put aside all the elements we need to build a reliable facial recognition system.

CHAPTER III : Applying The Modern Facial Recognition Pipeline

III.1 List of Hardware and Software Needed

Before we start the testing and experimenting first we need to prepare what we will need from hardware to software.

Hardware used :
PC - Raspberry pi 4 - A USB or Micro SD card - USB camera - Network Router

Software used :
Visual studio code - Python compiler and interpreter - C++ build tool - Raspberry pi imager - Raspberry pi OS
Coding language:
Python
Libraries used:
Imutils - CV2 - Mediapipe - Pickle - Face_recognition - MTCNN - DLIB

III.1.1 Overview of The Raspberry Pi

Raspberry pi : The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

What's more, the Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting bird houses with infra-red cameras.

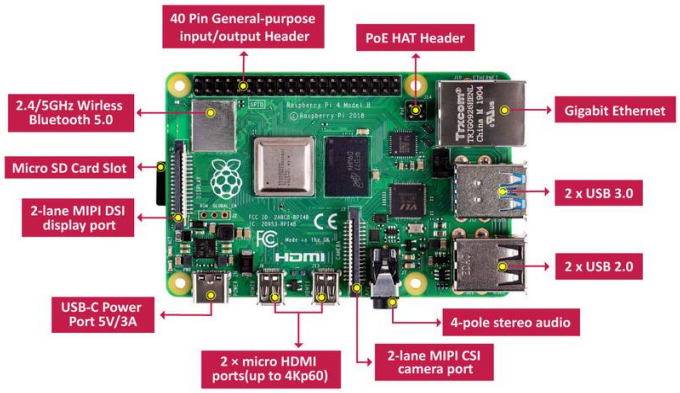
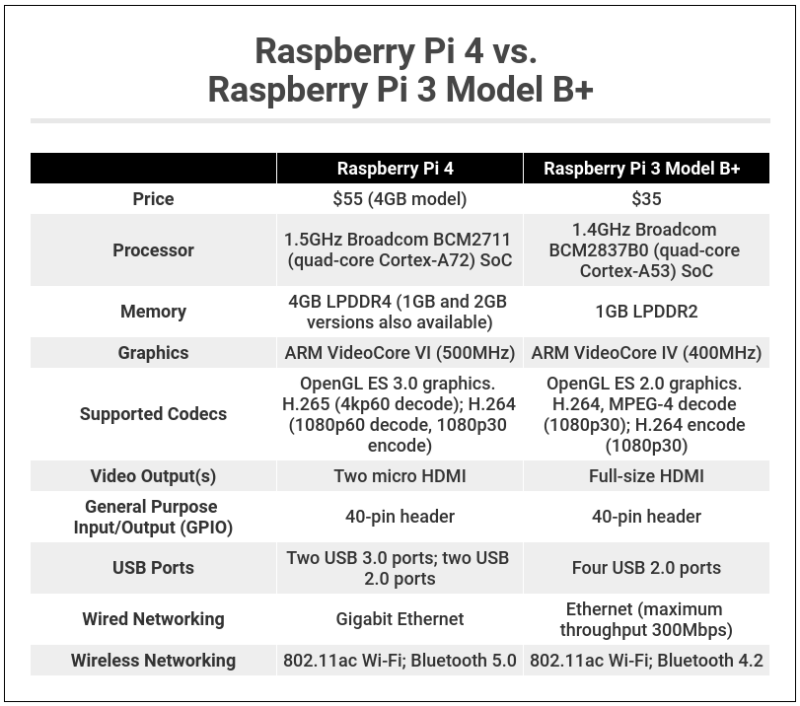


Figure III.1 : Overview over raspberry pi 4



Raspberry Pi 4 vs. Raspberry Pi 3 Model B+

	Raspberry Pi 4	Raspberry Pi 3 Model B+
Price	$55 (4GB model)	$35
Processor	1.5GHz Broadcom BCM2711 (quad-core Cortex-A72) SoC	1.4GHz Broadcom BCM2837B0 (quad-core Cortex-A53) SoC
Memory	4GB LPDDR4 (1GB and 2GB versions also available)	1GB LPDDR2
Graphics	ARM VideoCore VI (500MHz)	ARM VideoCore IV (400MHz)
Supported Codecs	OpenGL ES 3.0 graphics. H.265 (4kp60 decode); H.264 (1080p60 decode, 1080p30 encode)	OpenGL ES 2.0 graphics. H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30)
Video Output(s)	Two micro HDMI	Full-size HDMI
General Purpose Input/Output (GPIO)	40-pin header	40-pin header
USB Ports	Two USB 3.0 ports; two USB 2.0 ports	Four USB 2.0 ports
Wired Networking	Gigabit Ethernet	Ethernet (maximum throughput 300Mbps)
Wireless Networking	802.11ac Wi-Fi; Bluetooth 5.0	802.11ac Wi-Fi; Bluetooth 4.2

Figure III.2 : Raspy 4 and 3 models specs comparison

III.1.2 Overview of Key Libraries Used

OpenCV(Open Source Computer Vision Library) : is a computer vision and machine learning software library built to provide a common infrastructure for computer vision applications The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

Imutils library : A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Python 3.
This library will allow us to process and display the video stream using a separate CPU core (thread) and multitask which will increase the speed of the whole process

Mediapipe : an ultrafast face detection solution that comes with 6 landmarks and multi-face support. It is based on BlazeFace, a super-realtime performance lightweight and well-performing face detector enables it to be applied to any live viewfinder experience that requires an accurate facial region of interest as an input for other task-specific models.BlazeFace uses a lightweight feature extraction network inspired by, but distinct

from MobileNetV1/V2, an anchor scheme modified from Single Shot MultiBox Detector (SSD), and an improved tie resolution strategy alternative to non-maximum suppression.

Face_recognition : Recognize and manipulate faces from Python or from the command line with the world's simplest face recognition library.Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.comes with HOG and CNN pre built in algorithms.

Cmake : an open-source, cross-platform family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of your choice.

Dlib : a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments.

Pickle : The pickle module implements binary protocols for serializing and de-serializing a Python object structure.

III.2 Preparing the working environment

1. Download and install python from the official site https://www.python.org/ and integrate into system PATH

Figure III.3 : Python installer

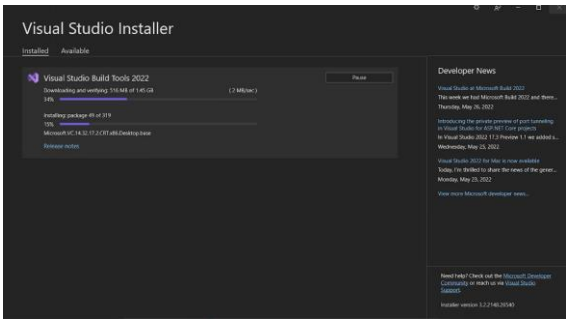


Figure III.4 :C++ build tools

2 download and install C++ build tools from the official site https://visualstudio.microsoft.com

3 Download and install Visual studio code from official site and we install the python extension

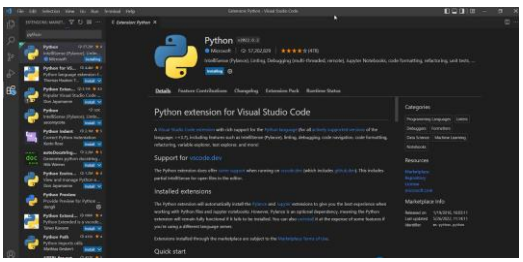


Figure III.5 :VS code python extension integration

4 Download required libraries using the pip3 command

5 Start writing the code needed

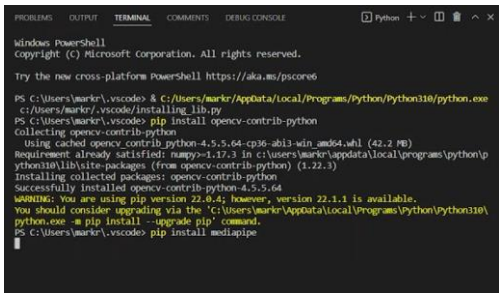


Figure III.6 :VS code terminal

6 download Raspberry pi imager and flash latest raspberry OS image on a usb or micro SD card

7 Insert the micro SD card or the usb into inthe raspberry pi and follow booting instruction proceeding to setting up for first time use while connected to internet to update

8 redownload required libraries again on the raspberry pi

9 copy the finished code and run it

Figure III.7 : Raspberry pi OS imager

III.3 Applying the modern facial recognition pipeline

III3.1 Face Detection

For the first step we'll be using mediapipe's revolutionary lightweight face detection algorithm ,an ultrafast face detection solution that comes with 6 landmarks and multi-face support. It is based on BlazeFace, a lightweight and well-performing face detector, an ultrafast face detection solution that comes with 6 landmarks and multi-face support. It is based on BlazeFace, a lightweight and well-performing face detector

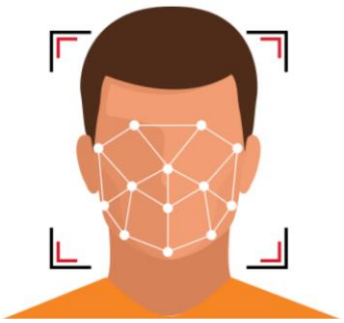


Figure III.8 : Facial Landmarks

III.3.2 Face Detection Concepts

Face detection locates human faces in visual media such as digital images or video. When a face is detected it has an associated position, size, and orientation; and it can be searched for landmarks such as the eyes and nose.Face tracking extends face detection to video sequences. Any face that appears in a video for any length of time can be tracked from frame to frame. This means a face detected in consecutive video frames can be identified as being the same person. Note that this isn't a form of face recognition; face tracking only makes inferences based on the position and motion of the faces in a video sequence.

A landmark is a point of interest within a face. The left eye, right eye, and base of the nose are all examples of landmarks, While a contour is a set of points that follow the shape of a facial feature.and Classification determines whether a certain facial characteristic is present. For example, a face can be classified by whether its eyes are open or closed, or if the face is smiling or not.

Minimum Face Size is the desired face size, expressed as the ratio of the width of the head to the width of the image. For example, the value of 0.1 means that the smallest face to search for is roughly 10% of the width of the image being searched.The minimum face size is a performance vs. accuracy trade-off: setting the minimum size smaller lets the detector find smaller faces but detection will take longer; setting it larger might exclude smaller faces but will run faster.

III.3.3 Face Alignment and Feature Extraction

The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 128 (x,y)-coordinates that map to facial structures on the face.

Regardless of which dataset is used, the same dlib framework can be leveraged to train a shape predictor on the input training data — this is useful if you would like to train facial landmark detectors or custom shape predictors of your own.

The indexes of the 128 coordinates can be visualized on the image below:

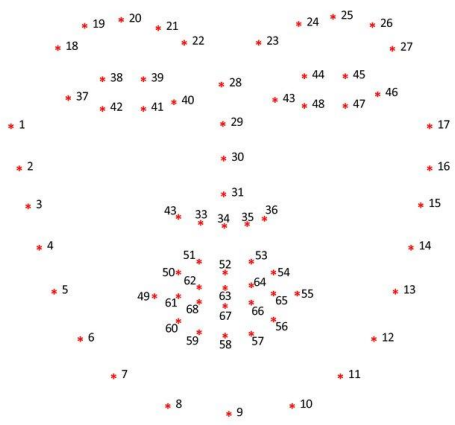


Figure III.9 : The indexes of the 68 coordinates

III.3.4 Feature Matching and Classification:

Feature matching refers to finding corresponding features from two similar images based on a search distance algorithm. One of the images is considered the source and the other as target, and the feature matching technique is used to either find or derive and transfer attributes from source to target image.

III.3.4.1 Overview of the Used Methods

A) MMOD(CNN) : A Max-Margin (MMOD) CNN face detector that is both highly accurate and very robust, capable of detecting faces from varying viewing angles, lighting conditions, and occlusion. Using the CNN and distance metric to compare faces
Finally, we can use the CNN to extract 128-dimensional vectors of faces from the aligned images. In the 128-d space, Euclidean distance corresponds directly to the measure of face similarity.

After transforming all dataset images using the pre-trained CNN, we input the image we would like to recognize through the same process to get a similar 128-d vector (embedding). Faces similarity can be measured using the Euclidean distance.

The following figure shows graphically how this process works and illustrates an example of distances measured by the algorithm between similar and different faces.

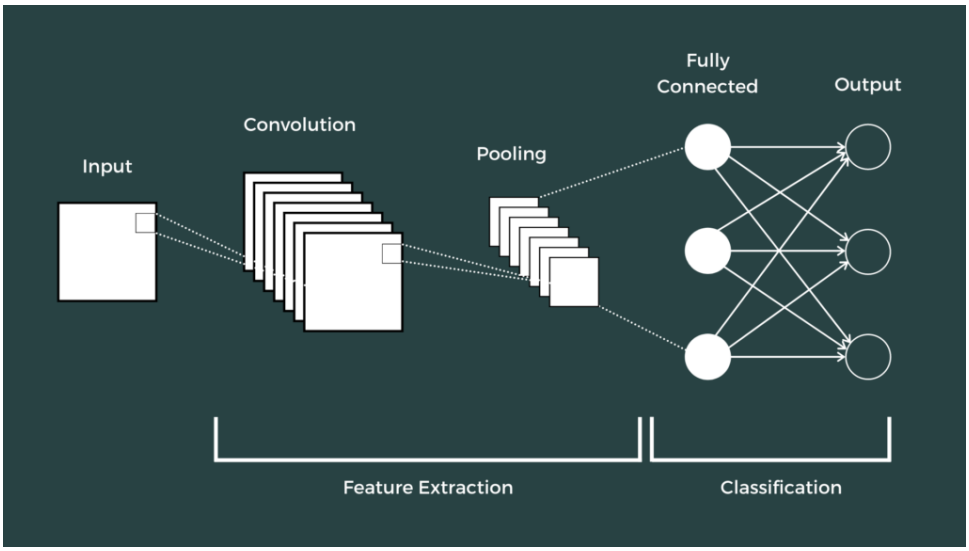


Figure III.10 : CNN algorithm architecture

B) HOG(Histogram Oriented Gradients) : HOG is a simple and powerful feature descriptor. It is not only used for face detection but also it is widely used for object detection like cars, pets, and fruits. HOG is robust for object detection because object shape is characterized using the local intensity gradient distribution and edge direction.The basic idea of HOG is dividing the image into small connected cells Computes histogram for each cell.

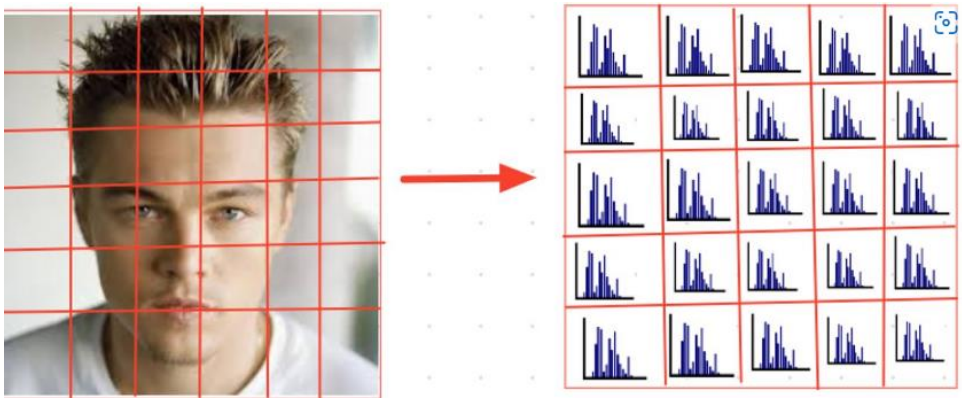


Figure III.11 : Separate cells Histogram Computing

Bring all histograms together to form feature vector i.e., it forms one histogram from all small histograms which is unique for each face

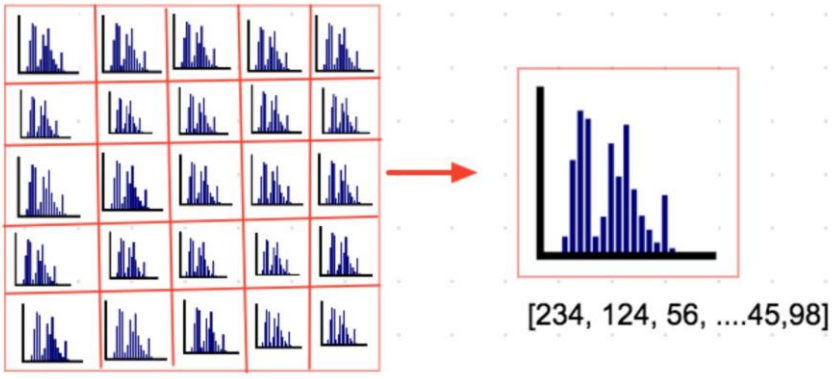


Figure III.12 : Merging histograms into vectors

III.3.4.2 Optimal distance threshold

To be able to tell whether two faces belong to the same person or not, a distance threshold must be determined. To find the optimal value of the threshold, different values will be tested using our database images. We will plot accuracy versus different values of the threshold distance and pick the best value.

III.3.4.3 HoG Vs. MMOD

The appeal of HoG + Linear SVM under Dlib is its low use of resources; its efficacy when operating on CPU; the fact that it has at least some latitude for non-frontal faces; its low-impact model requirements; and a relatively capable occlusion detection routine.

Negatively, a default deployment requires a minimum face-size of 80x80 pixels. If you need to detect faces below this threshold, you'll need to train your own implementation. Additionally, this approach gives poor results on acute face angles; generates bounding boxes that may over-crop facial features; and struggles with challenging occlusion cases.

The advantage of MMOD (CNN) under Dlib is (perhaps above all) its ability to recognize difficult face orientations (which may be the deciding factor, depending on your target environment); its impressive speed when allowed access to even a moderately-specced GPU; its lightweight training architecture; and its superior occlusion handling.

Negatively, it can produce bounding boxes even more restricted than HoG + Linear SVM in a default deployment; performs notably more slowly on a CPU than HoG/LSVM; and shares HoG/LSVM's native inability to detect faces smaller than 80 pixels square — again, necessitating a custom build for certain scenarios, such as acute street surveillance viewpoints that extend into the distance.

The pre-trained Inception CNN model for facial recognition that was tuned with the dataset gave a very good performance (97% accuracy on our custom dataset) especially when we added the MMOD approach.The ResNet network is a much deeper pretrained network than the Inception that achieved 99% accuracy on a publicly available dataset containing more than 13000 images called “Labeled Faces in the Wild” provides accurate face detection model. However CNN based models are required heavy resources on the hardware considering HOG technique requires far less resources and barely falls off in accuracy compared the CNN models its the valid option.

III.3.4.4 Conclusion

In this chapter we have defined our needs in terms of hardware and software for the realization of a modern facial recognition system using different detection and recognition algorithms. The libraries used are also defined as well as the preparatory steps to achieve such a system, we end the chapter with a theoretical comparison between the CNN and the HOG which will allow us to choose the model to use in our project.

Chapter IV:Applying and Comparing Different modern Technologies on the raspberry 4 Embedded system

IV.1 Pretreatment and Initializing

IV.1.1 Generating dataset

1) Here we start by writing a script to generate our dataset that includes the name of the persons and the images associated with them, first we have to ask for the identifiers of the person to classify them with such name or ID and initialize the path to the location for saving the outputs.

```
print("Enter the id and name of the person: ")
userName = input()
userId = input()

cwd = os.getcwd()
Path("dataset").mkdir(parents=True, exist_ok=True)
Path("dataset\{}".format(userName)).mkdir(parents=True, exist_ok=True)
```

2) In this step we load our mediapipe face detector and initialize out minimum confidence .

```
mp_face_detection = mp.solutions.face_detection
detector = mp_face_detection.FaceDetection( min_detection_confidence = 0.8)
```

3)Next step we initialize the video stream, allow the camera sensor to warm up, and initialize the counter for the total number of example faces written to disk.

```
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
time.sleep(2.0)
total = 0
```

4) And we start looping over the captured frames from the video stream while using mediapipe detector to detect faces and bound with a box to show face location in the image and crop it to reduce background noise .

```
while True:
  frame = vs.read()
  frame =cv2.flip(frame,1)
  results = detector.process(frame)
  if results.detections:
    for face in results.detections:
```

```
        confidence = face.score[0]
        cfd="{:.0%}".format(confidence)
        bounding_box = face.location_data.relative_bounding_box

        x = int(bounding_box.xmin * frame.shape[1])
        w = int(bounding_box.width * frame.shape[1])
        y = int(bounding_box.ymin * frame.shape[0])
        h = int(bounding_box.height * frame.shape[0])

        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 255, 255), thickness = 2)
        crop=frame[y:y+h,x:x+w]
```

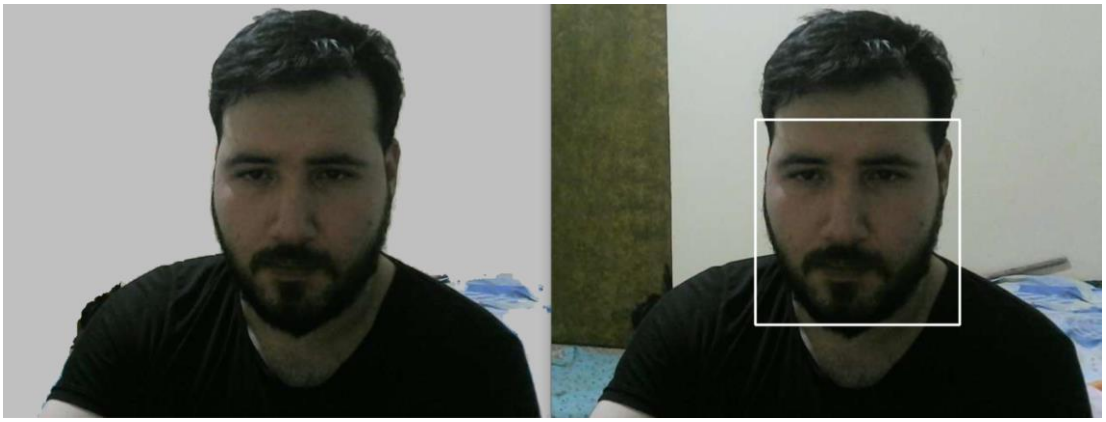


Figure IV.1 : Face detection using Mediapipe.

5) In this step we preview the results .

```
    cv2.imshow('crop',crop)
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
```

Figure IV.2 : Cropping the region of interest (the face).

6) And as the last step if the `k` key was pressed we write the cropped images we got under a folder classified using the name or ID we imputed earlier so we can later process it and use it for face recognition .

```
  if key == ord("k") and confidence >0.9:
    p            =            os.path.sep.join(["dataset\{}".format(str(userName)),
"{}_{}_{}.png".format(str(userName),str(userId),str(total).zfill(3))])
    cv2.imwrite(p, crop)
    total += 1
  elif key == ord("q") :
    break
```

IV.1.2 Extracting and encoding the facial features

1) Here we start by grabbing the paths to the input images in our dataset and initializing the list of known encodings and known names.

```
imagePaths = list(paths.list_images(datasetP))
knownEncodings = []
knownNames = []
```

2) Next we loop over the image paths, extract the person name from the image path, load the input image and convert it from BGR (OpenCV ordering) to dlib ordering (RGB) while initializing the feature extraction methods in this example CNN.

```
for (i, imagePath) in enumerate(imagePaths):
  print("[INFO] processing image {}/{}".format(i + 1,len(imagePaths)))
  name = imagePath.split(os.path.sep)[-2]
  image = cv2.imread(imagePath)
  rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
  boxes = face_recognition.face_locations(rgb,model='cnn')
  encodings    =    face_recognition.face_encodings(rgb,    boxes,    num_jitters=
10,model='large')
  for encoding in encodings:
    knownEncodings.append(encoding)
    knownNames.append(name)
```

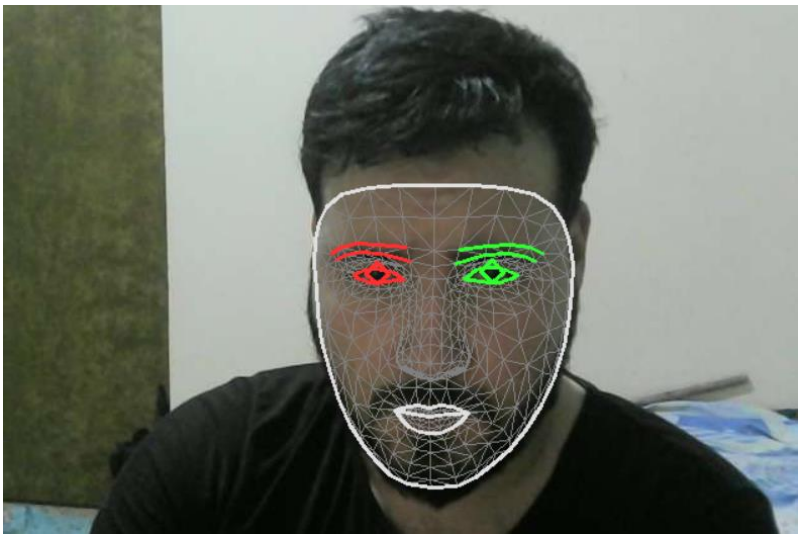


Figure IV.3 : Face landmarks mesh

3) And in the last step we dump the extracted features in a pickle file, a file type that is developed to be fastest at reading, writing or compressing using the python language.

```
print("[INFO] serializing encodings...")
data = {"encodings": knownEncodings, "names": knownNames}
f = open("Encodings", "wb")
f.write(pickle.dumps(data))
f.close()
```

IV.2 : Real time Application of Facial Recognition

1) As always we start by importing our libraries.

```
from imutils.video import VideoStream
from pathlib import Path
import time
import cv2
import mediapipe as mp
import pickle
import face_recognition
import os
```

2) Next we Import the pickle file that is our database for the facial features encodings and the names or IDs associated with them

```
print("[INFO] loading encodings...")
data = pickle.loads(open('Encodings', "rb").read())
```

3) Now we input the initial facial detection parameters such as the face detection algorithm and the confidence threshold

```
mp_face_detection = mp.solutions.face_detection
detector = mp_face_detection.FaceDetection( min_detection_confidence = 0.7)
```

4) As always we initialize the video stream, allow the camera sensor to warm up

```
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
time.sleep(2.0)
```

5) Again we apply medipipe facial detection module and get the face cropped region of interest

```
with                              mp_face_detection.FaceDetection(model_selection=0,
min_detection_confidence=0.7) as face_detection:
  while True:
    frame =cv2.flip(vs.read(),1)
    # detect faces in the grayscale frame
    results = detector.process(frame)
    if results.detections:
      for face in results.detections:
        confidence = face.score[0]
        cfd="{:.0%}".format(confidence)
        bounding_box = face.location_data.relative_bounding_box

        x = int(bounding_box.xmin * frame.shape[1])
        w = int(bounding_box.width * frame.shape[1])
        y = int(bounding_box.ymin * frame.shape[0])
        h = int(bounding_box.height * frame.shape[0])

        crop=frame[y:y+h,x:x+w]
```

6)In this step we initialize the facial recognition method(In this example we are using HOG method) and start the process while setting our confidence parameter high enough to get accurate results

```
        boxes = face_recognition.face_locations(crop,model='hog')
        encodings = face_recognition.face_encodings(crop, boxes)
        names = []
        while confidence >0.9:
```

7) We initialize any detected face as “Unknown” first before we cycle throughout encoded facial features in the dataset and attempt to match each face in the input image to our known encodings determining the recognized face with the largest number of votes

```
        for encoding in encodings:
          matches = face_recognition.compare_faces(data["encodings"],
            encoding)
          name = "Unknown"
          if True in matches:
            matchedIdxs = [i for (i, b) in enumerate(matches) if b]
            counts = {}
            for i in matchedIdxs:
              name = data["names"][i]
              counts[name] = counts.get(name, 0) + 1
            name = max(counts, key=counts.get)
```

8) We update the list of names and draw the predicted face name over the bounding box in the output stream and show throughput result

```
          color=(0,0,255)
          names.append(name)
          cv2.rectangle(frame, (x, y), (x + w, y + h), color, thickness = 1)
          cv2.putText(frame, name+' '+cfd, (x, y), cv2.FONT_HERSHEY_SIMPLEX,
          0.75, color, 2)
          count=0
        else:
          break
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
```

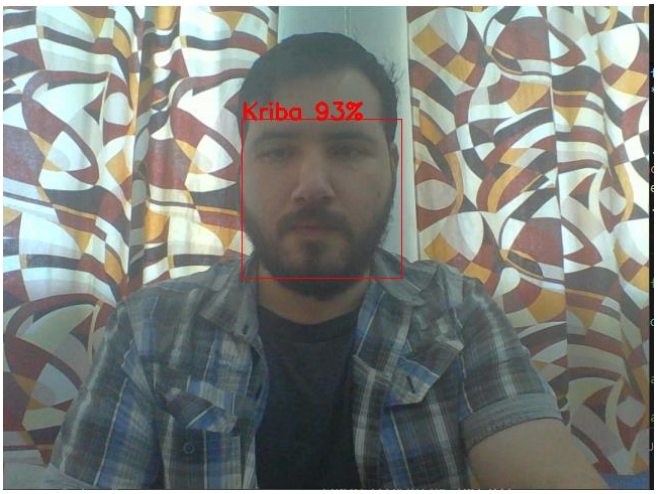


Figure IV.4 :HOG facial recognition result

IV.3 Comparison between some of the common face detection algorithms

We will be using Haar, dlib, Multi-task Cascaded Convolutional Neural Network (MTCNN), and OpenCV's DNN module and doing some tests under different circumstances Comparing Results on Images

IV.3.1 Comparing face detection techniques under different circumstances

Figure IV.5 : Frontal face detection comparison (HOG vs Mediapipe vs HAAR)

Figure IV.6 : Frontal face detection comparison (MTCNN vs MMOD vs DNN)

Figure IV.7 : Upward face detection comparison (HOG vs Mediapipe vs HAAR)

Figure IV.8 : Upward face detection comparison (MTCNN vs MMOD vs DNN)

Figure IV.9 : Side Face detection comparison (HOG vs Mediapipe vs HAAR)

Figure IV.10: Side Face detection comparison(MTCNN vs MMOD vs DNN)

Figure IV.11: Tilted head face detection comparison (HOG vs Mediapipe vs HAAR)

Figure IV12 : Tilted head face detection comparison (MTCNN vs MMOD vs DNN)

Figure IV.13 : Low lighting face detection comparison (HOG vs Mediapipe vs HAAR)

Figure IV.14 : Low lighting face detection comparison (MTCNN vs MMOD vs DNN)

Figure IV.15 : Face detection through occlusion comparison (HOG vs Mediapipe vs HAAR)

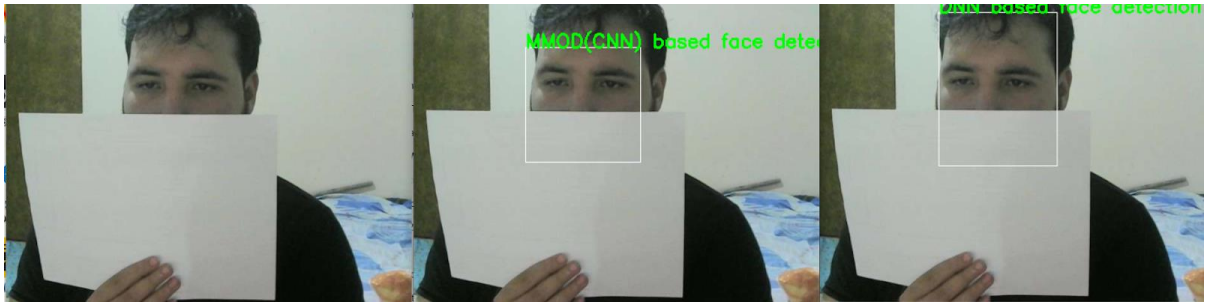


Figure IV.16 : Face detection through occlusion comparison(MTCNN vs MMOD vs DNN)

IV.3.2 Comparing face detection techniques results

The results achieved comparison is noted on the table below and the image size passed is 640x480 except for the DNN module which is passed a 300x300

FD methods	Frontal face	Upper face	Side face	Titles angle	Low lighting	Face abstraction
Haar	Detected	Failed	Failed	Failed	Failed	Failed
HOG	Detected	Detected	Detected	Failed	Detected	Failed
Mediapipe	Detected	Detected	Detected	Detected	Detected	Detected
MTCNN	Detected	Detected	Detected	Failed	Detected	Failed
MMOD(CNN)	Detected	Detected	Detected	Detected	Detected	Detected
DNN	Detected	Detected	Detected	Detected	Detected	Detected

Table IV.1 : Tables representing the results achieved from the tests

Haar cascades, as expected, performed the worst out of all of them;as it can only detect the face in certain immutable conditions ,followed next by MTCNN and HOG that both struggled with face abstraction and tilted angels while Dlib's MMOD(CNN),Mediapipe and DNN had pretty even performances being able to detect a the face at most conditions.

Frame Rate

The values reported are obtained using an AMD Ryzen 5 3400 and the image size passed is 640x360 except for the DNN module which is passed a 300x300 image as it has been done until now.

FD Methods	Fps (s)	latency (s)	50% res fps (s)	50% res latency (s)
HAAR	15.04	0.043	29.69	0.013
HOG	18.53	0.047	52.80	0.056
Mediapipe	75.20	0.0066	62.42	0.006
MTCNN	1.58	0.628	1.97	0.499
MMOD (CNN)	1.24	0.847	2.05	0.584
DNN	20.04	0.0404	20.30	0.037

Table IV.2 : PC testing results for FPS and execution time per frame on pc at 640x480 resolution

	Fps (s)	latency (s)	50% fps (s)	50% res latency (s)
HAAR	4.65	0.208	11.34	0.081
HOG	7.59	0.123	22.83	0.036
Mediapipe	23.61	0.028	34.51	0.214
MTCNN	0.33	2.969	0.49	2.023
MMOD (CNN)	0.09	11.035	0.35	2.822
DNN	3.84	0.253	4.07	0.240

Table IV.3 : Raspberry pi 4 testing results for FPS and execution time per frame on pc at 640x480 resolution

IV.3.3 Comparison between HOG and MMOD facial recognition algorithms

Algorithms	Time to encode 10 images dataset x 10 Jitters on PC	FPS on PC	Latency on PC	FPs on raspy4	Latency on raspy 4	Accuracy %
HOG	17.7881	2.62	0.322	2.60	0.342	92 $\pm$(*4%*)
MMOD (CNN)	57.80	0.13	7.265	0.46	1.849	94 $\pm$(*4%*)

Figure IV.4 : Face recognition results comparison

we notice that by reducing the resolution we achieve better frame rate and latency on CNN model with just CPU but that limits us to having perfect environment with reduced noise to be able to achieve high accuracy needed for security standards. However HOG method is designed specifically with CPU in mind reducing the need for higher output made it more compatible with embed systems that lack a dedicated GPU made it possible to achieve higher processing speed and real time application without latency delay

IV.4 Conclusion

Mediape came out as the clear winner as it was able to perform the process at high FPS with lowest latency by massive margin on both the PC and the raspberry pi 4 , followed by Haar,HOG and DNN algorithms as they are designed with CPU on mind while the CNN based algorithms MMOD and MTCNN fall behind and failing to achieve real time process which isn't surprising since they are designed with GPU in mind and run too slow on just a cpu.

For general computer vision problems on embedded systems with limited performance, Mediapipe module is the best. It works well with occlusion, quick head movements, and can identify side faces as well. Moreover, it also gave the highest fps with lowest latency among all.

General conclusion

The study presented in this master thesis show through a proof of concept that it is possible to build a system capable of identify a group of subjects through facial recognition, running on embedded hardware with good performances in terms of precision, simulated execution time and speed.

CNN, HAAR, mediapipe, MTCNN, DNN and HOG algorithms were chosen for the facial detection and only two algorithms CNN and HOG are used for facial recognition. Deep learning approach is using in the two phases since it become the central component of most face recognition algorithms being developed, massive gains in recognition accuracy have been made in the last years resulting in numerous technologies developed perfected simultaneously

In terms of hardware, the embedded development boards raspberry pi 4 and camera are both fast and accurate enough to establish a stable running environment for the design and development of the face recognition application based on deep learning algorithms to and extent without the need to have a dedicated gpu extension enabling it to be deployed at daily cheap cost compared to the alternative without losing much performance

The system has been trained and tested on a dataset of 10x10 jitter images. The final resulting pipeline is able to discriminate successfully more than 96% of the test using CNN algorithm and 94% using HOG algorithm, for facial detection, the maximum speed is 75.20 FPS using mediapipe when executing on PC. The maximum of latency 0.847 s using MMOD (CNN) when executing on PC. Results and comparative table are well defined in chapter 4.

Bibliography

[1]The History of Fingerprints Archived 12 March 2013 at the Wayback Machine.
[2]Josh Ellenbogen, Reasoned and Unreasoned Images: The Photography of Bertillon, Galton, and Marey (University Park, PA, 2012)
[3] Nitzan Lebovic, "Biometrics or the Power of the Radical Center", in Critical Inquiry 41:4 (Summer, 2015), 841–868.
[4]Nitzan Lebovic, "Biometrics or the Power of the Radical Center", in Critical Inquiry 41:4 (Summer, 2015), p. 853
[5]Kelly A. Gates, Our Biometric Future: Facial Recognition Technology and the Culture of Surveillance (New York, 2011), p. 100.
[6]http://www.smartcardalliance.org
[7]Woodward Jr, J. D., Horn, C., Gatune, J., & Thomas, A. (2003). Biometrics: A look at facial recognition.
[8]Webopedia, "Authentication," 2016. [Online]. Available:
[9]H. Abie, "semanticscholar," 12 12 2006. [Online]. Available:
https://pdfs.semanticscholar.org/3733/2607f7a7ac828
4c514845957fd00583e5614.pdf
[10]Rudd M.Bolle,Jonathan H.Connell,Sharath Pankanti , Nalini K . Ratha , Andrew W . Senior, Guide to biometrics , Springer Publication (2003)
[11]Davide Maltoni, Anil K. Jain ,Handbook of fingerprint recognition,Springer publication (2002)
[12]http://www.findbiometrics.com
[13]Biometrics: Retinal Scanning Amy Zalman
[14]http://www.tiresias.org/
[15]www.howstuffworks.com
[16]Iqbal, I., & Qadir, B. (2012). Biometrics Technology: Attitudes & influencing factors when trying to adopt this technology in Blekinge healthcare
[17]Iqbal, I., & Qadir, B. (2012). Biometrics Technology: Attitudes & influencing factors when trying to adopt this technology in Blekinge healthcare.
[18]Epstein C. (2007), "Guilty Bodies, Productive Bodies, Destructive Bodies: Crossing the Biometric Borders". International Political Sociology, 1:2, 149–64

[19]Mordini, E; Massari, S. (2008), "Body, Biometrics and Identity" Bioethics, 22, 9:488
[]UNICEF, Birth Registration Archived 6 September 2015 at the Wayback Machine
[20]Dahan M., Gelb A. (2015) "The Role of Identification in the Post-2015 Development Agenda" Archived 20 September 2015 at the Wayback Machine – World Bank Working Paper No. 98294 08/2015;
[21]Mordini E, Rebera A (2011) "No Identification Without Representation: Constraints on the Use of Biometric Identification Systems". Review of Policy Research, 29, 1: 5–20
[22]Mordini E, Ashton H,(2012), "The Transparent Body – Medical Information, Physical Privacy and Respect for Body Integrity", in Mordini E, Tzovaras D (eds), Second Generation Biometrics: the Ethical and Social Context. Springer-Verlag: Berlin
[23]Mordini E, Tzovaras D,(2012), Second Generation Biometrics: the Ethical and Social Context. Springer-Verlag: Berlin