**DESIGN AND IMPLEMENTATION OF AN AI VIRUS CONTROL SYSTEM**

**ABSTRACT**

Artificial Intelligence is a powerful technology that has taken the world by storm in the last decade. It has a wide range of applications provides much utility to humans within this technological era where computers can run large numbers of simulations with the aid of Artificial Intelligence, Computer Vision and Big Data sets.

One of the biggest challenges to the human race is keeping up and overcoming with new and potential viruses and diseases that can be fatal.

However, we now have computer systems that can replicate the random evolutionary process of viruses to help us see what works and what does not work and also create logic for the purpose of assisting to constrain diseases and other by utilizing the power of computer vision.

The main purpose of this project and research was to device a more effective and efficient way to track, control and reduce the effect of the rapid spread of such influenza. Such issues have caused many causalities in the past and even in present time. The current pandemic has the brightest minds in the global health community busy devising countermoves to hold them off.

This system proposes a solution to such occurrences that get out of control rather quickly. An embedded system with the ability to detect people, objects, and anomalies in regards to safety and security protocols in a given set of constraints. The system is designed to be cost effective, reliable and safe to aid in the fight against viruses now and in the future.

**TABLE OF CONTENTS**

**[DEDICATION](#page6) [V](#page6)**

**[ABSTRACT](#page7) [VI](#page7)**

**[LIST OF FIGURES](#page10) [IX](#page10)**

**[LIST OF TABLES](#page11) [X](#page11)**

**[LIST OF ABBREVIATIONS](#page12) [XI](#page12)**

**[CHAPTER ONE](#page13) [1](#page13)**

**[INTRODUCTION](#page13) [1](#page13)**

*[1.1](#page13) [Overview](#page13) [1](#page13)*

*[1.2](#page14) [Background and Motivation](#page14) [2](#page14)*

*[1.3](#page15) [Statement of the Problem](#page15) [3](#page15)*

*[1.4](#page15) [Aim and Objectives](#page15) [3](#page15)*

*[1.5](#page16) [Significance of the Project](#page16) [4](#page16)*

*[1.6](#page17) [Project Risks Assessment](#page17) [5](#page17)*

*[1.7](#page17) [Scope/Project Organization](#page17) [5](#page17)*

**[CHAPTER TWO](#page19) [7](#page19)**

**[LITERATURE REVIEW](#page19) [7](#page19)**

*[2.1](#page19) [Introduction](#page19) [7](#page19)*

*[2.2](#page19) [Historical Overview](#page19) [7](#page19)*

*[2.3](#page28) [Related Hardware](#page28) [16](#page28)*

*[2.4](#page30) [Improvements](#page30) [18](#page30)*

*[2.5](#page30) [Summary](#page30) [18](#page30)*

**[CHAPTER THREE](#page31) [19](#page31)**

**[REQUIREMENTS ANALYSIS AND DESIGN](#page31) [19](#page31)**

*[3.1](#page31) [Overview](#page31) [19](#page31)*

*[3.2](#page31) [Proposed Methodology](#page31) [19](#page31)*

*[3.3](#page32) [Approach to Chosen Methodology/Methods](#page32) [20](#page32)*

*[3.4](#page33) [Tools and Techniques](#page33) [21](#page33)*

*[3.5](#page35) [Requirement Analysis](#page35) [23](#page35)*

*[3.6](#page35) [Requirements Specifications](#page35) [23](#page35)*

*[3.7](#page37) [System Design](#page37) [25](#page37)*

*[3.8](#page40) [Summary](#page40) [28](#page40)*

**[CHAPTER FOUR](#page41) [29](#page41)**

**[IMPLEMENTATION AND TESTING](#page41) [29](#page41)**

*[4.1](#page41) [Overview](#page41) [29](#page41)*

*[4.2](#page41) [Main Features](#page41) [29](#page41)*

*[4.3](#page46) [Problems Encountered](#page46) [34](#page46)*

*[4.4](#page47) [Solving Problems](#page47) [35](#page47)*

*[4.5](#page47) [Testing](#page47) [35](#page47)*

*[4.6](#page49) [Use Guide](#page49) [37](#page49)*

*[4.7](#page49) [Summary](#page49) [37](#page49)*

vii

**[CHAPTER FIVE](#page50) [38](#page50)**

**[CONCLUSION](#page50) [38](#page50)**

*[5.1](#page50) [Overview](#page50) [38](#page50)*

*[5.2](#page50) [Achievement](#page50) [38](#page50)*

*[5.3](#page50) [Limitations and Challenges](#page50) [38](#page50)*

*[5.4](#page51) [Continuous Development and Integration (CD/CI)](#page51) [39](#page51)*

*[5.5](#page51) [Summary](#page51) [39](#page51)*

**[REFERENCES](#page52) [40](#page52)**

**[APPENDICES](#page53) [41](#page53)**

**[APPENDIX A](#page53) [41](#page53)**

**[SOURCE CODES](#page53) [41](#page53)**

**[APPENDIX B](#page59) [47](#page59)**

viii

**LIST OF FIGURES**

[Figure 1.1](file://///Users/peter/Documents/Baze/400lvl/21A/Final%20Year%20Project/DESIGN%20AND%20IMPLEMENTATION%20OF%20AN%20AI%20VIRUS%20CONTROL%20SYSTEM%20page%205.docx%23_Toc75953924) [1](file://///Users/peter/Documents/Baze/400lvl/21A/Final%20Year%20Project/DESIGN%20AND%20IMPLEMENTATION%20OF%20AN%20AI%20VIRUS%20CONTROL%20SYSTEM%20page%205.docx%23_Toc75953924)

[Figure 2.1](file://///Users/peter/Documents/Baze/400lvl/21A/Final%20Year%20Project/DESIGN%20AND%20IMPLEMENTATION%20OF%20AN%20AI%20VIRUS%20CONTROL%20SYSTEM%20page%205.docx%23_Toc75953925) [8](file://///Users/peter/Documents/Baze/400lvl/21A/Final%20Year%20Project/DESIGN%20AND%20IMPLEMENTATION%20OF%20AN%20AI%20VIRUS%20CONTROL%20SYSTEM%20page%205.docx%23_Toc75953925)

[Figure 2.3](#page27) [15](#page27)

[Figure 3.4](#page40) [28](#page40)

[Figure 5](#page59) [47](#page59)

[Figure 6](#page60) [48](#page60)

[Figure 7](#page61) [49](#page61)

[Figure 8](#page61) [49](#page61)

[Figure 9](#page62) [50](#page62)

[Figure 10](#page63) [51](#page63)

ix

**LIST OF TABLES**

[Table 1.1](#page17) [5](#page17)

[Table 2.1](#page23) [11](#page23)

[Table 3.1](#page36) [24](#page36)

x

**LIST OF ABBREVIATIONS**

AI

ML

DL

CPU

REST

API

TDD

ERD

IT

UML

URL

HTTP

CSS

HTML

WWW

UI

ANN

GPU

Artificial Intelligence

Machine Learning

Deep Learning

Central Processing Unit

Representational State Transfer

Application Programming Interface

Test-Driven Development

Entity Relationship Diagram

Information Technology

Unified Modeling Language

Uniform Resource Identifier

Hypertext Transfer Protocol

Cascading Style Sheet

Hyper Text Markup Language

World Wide Web

User Interface

Artificial Neural Network

Graphic Processing Unit

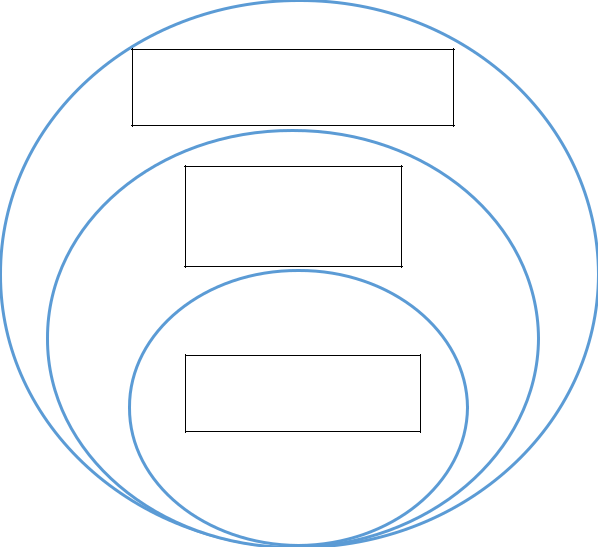
xi

**CHAPTER ONE**

**INTRODUCTION**

**1.1 Overview**

In contrast to natural intelligence shown by humans and animals which involve consciousness and emotions; artificial intelligence is a form of intelligence exhibited by machines and robots which enable computers to imitate the recognition, decision-making, learning and problem-solving abilities of the human mind. In majority of the use of Artificial Intelligence, a computer, robot or machine imitates the capabilities of the human brain by learning from a huge set of data which is more commonly called Big Data. Through the different types of learning which are supervised learning, unsupervised learning and reinforcement learning; AI models are able to go through Big Data and learn from these data sets to perform different abilities like facial recognition and predictive analysis. There are two subsets of Artificial Intelligence which are Machine Learning (ML) and Deep learning (DL).



Artificial Intelligence

Machine

Learning

Deep Learning

***Figure 1.1: Subsection of Artificial Intelligence***

1

One can think of AI as an onion with several layers. Machine Learning learns by itself by reprogramming its code as it digests more data in order to perform a specific task its designed to perform with higher accuracy. Deep learning is a subset of both machine learning and AI that teaches itself to perform a specific task with higher accuracy without human intervention.

AI is already widely adopted in most technology companies and is used by many individuals in our day to day lives. Examples of AI applications are Apple’s Siri, Amazon’s Alexa, self-driving cars, etc.

**1.2** **Background and Motivation**

The AI industry as of the 21st century has received so much publicity and promotion especially in the first decade of the century. Since then, it has grown to a point where its power and application have been better utilized in the enhancement of modern technology and civilization in specific industries like healthcare, automation, marketing/entertainment industry, risk management and so on.

In my opinion, the recent pandemic has reminded us of some loop holes we have in our current society knowing that there could have been protocols and technologies put in place earlier to prevent and contain the spread of this virus. Contact tracing systems, face mask protocols, social distance enforcement and regulatory systems among others would have been a big helping hand saved the lives of many resulting in a lower mortality rate and transfer rate.

The resources, methods and systems are available and all that is left is the integration process and the poise to translate these into a system that will greatly assist the healthcare workers in containing this and future spread of viruses and diseases.

2

**1.3** **Statement of the Problem**

The coronavirus disease (COVID-19) is a genetically mutated virus caused by a new strain of coronavirus (SARS-CoC-2) which has been identified in humans in the past. It was originally reported to the World Health Organization (WHO) on December 31, 2019. Just one month after this declaration, the WHO announced this initially minor outbreak as a global health emergency, and two months after, on 11 March, 2020, the WHO declared the COVID-19 A GLOBAL PANDEMIC, the first of its kind since declaring the H1N1 influenza pandemic in 2009. As of today, there is a staggering amount of 112 million cases worldwide, and unfortunately, 2.49 million deaths, most of which could have been avoided by taking the right measures and adopting the right technologies and protocols early enough in different states and countries. In Nigeria, there is a total number of 153 thousand cases and 1,874 confirmed deaths according to Google Trends and Statistics.

This pandemic is the number one pressing issue on the planet as of the moment. It is still spreading and there are now different variations to the virus which have proven to be either more fatal or more contagious. The main and most common symptoms of this disease are: fever; dry cough; and tiredness. Some lesser common symptoms are: aches and pains; sore throat; loss of taste or smell; etc. Furthermore, there are more serious symptoms in severe cases which include: difficulty breathing or shortness of breath; chest pain or pressure; or loss of speech or movement.

**1.4** **Aim and Objectives**

This brings me to my proposition for this project which is to utilize and integrate the powers of Artificial Intelligence (AI) and computer vision working hand-in-hand with external sensors and calibrators that will gather data from its environment and send it to the AI computer microprocessor and return signals and information that even a layman can interpret. This would be implemented using Nvidia Jetson Nano Developer Kit as the hardware.

3

Below is a list of all the aims objectives this project hopes to accomplish:

* To detect if a person is wearing a facemask;
* To detect when a person is running a high temperature;
* To aid with contact tracing in an event of an infected person;
* Help with social distancing rules and regulations;
* Alerts a red or green signal if all protocols are observed or if all tests are passed;
* Contacts all admin personnel via email with a screenshot when any test or protocol fails;
* Keeps record in a database when social distancing measures are not observed;
* Thermal sensor can also help detect heat signature to reveal if someone is carrying a weapon or something of the sort of possible weapons;
* Will make use of text to speech functionalities; and
* Will display information and recordings on website or mobile application for ease of use by selected users and admin.

Finally, I will be making use of an extra camera module that has the capability for thermal sensing for contactless reading and imaging of one’s body temperature to detect if the temperature is high and hence a fever.

All this information will be gathered into a database and will be output into a Website where the information can be managed and collected by restricted personnel.

**1.5** **Significance of the Project**

In one way or the other, most individuals have been deeply affected by this pandemic either directly or indirectly. This project brings numerous value propositions to the society. Adopting to a system like this will vastly improve the current situation in several areas and can potentially reduce the lockdown to allow people get back to their normal lives. The application will run through all the checks and conditions necessary to prevent the spread of the virus.

4

Knowing fully well that symptoms like high temperature do not always show in the early stages of the virus in a person, it will carry out the mundane checks performed by security personnel with a higher level of accuracy and effectiveness. This does not mean that the system would leave any working personnel jobless, but rather aid them in their job to achieve better results as the application does not get tired or display other emotions shown by humans ensuring that the work is done.

**1.6** **Project Risks Assessment**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RISKS |  |  | SOLUTIONS |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  |  |  | |  | |
|  | Loss of data while working. |  |  | **Daily backups of the data storage device on** |  | |  | |
|  |  |  |  | **a safe and secure hard drive and OneDrive.** |  | |  | |
|  |  | |  |  |  | |  | |
|  | Electrostatic Discharging leading to damage | |  | **Use of Electrostatic mat to prevent such an** | |  | |
|  | of hardware. | |  | **occurrence.** | |  | |
|  |  | |  |  | |  | |
|  | Material requirements needed for project does |  |  | **All deliveries have been set to priority and** |  | |  | |
|  | not arrive in time. |  |  | **insured. Also, there are alternative stores** |  | |  | |
|  |  |  |  | **to purchase from in Nigeria.** |  | |  | |
|  |  | |  |  |  | |  | |
|  | Bug in code near end of project completion. | |  | **All code for this project is stored in a** | |  | |
|  |  |  |  | **repository on Git hub and an instance of** | |  | |
|  |  |  |  | **said code is backed up at least once a day.** | |  | |
|  |  | |  |  | |  | |
|  | Someone hacking into the Website or |  |  | **The right measures have been put in place** |  | |  | |
|  | database |  |  | **to ensure the safety and security of all data** |  | |  | |
|  |  |  |  | **on the database due to the fact that there is** |  | |  | |
|  |  |  |  | **important information stored there.** |  | |  | |
|  | | |  |  |  | |  | |
| *Table 1.1* | | |  |  |  | |  | |

**1.7** **Scope/Project Organization**

This project has an underlining objective to design a suitable prototype that will aid in the prevention of contagious virus outbreaks in an environment through the use of external camera sensors and AI object detection abilities. This system will have it frontend on a

5

Website which can be accessed on all devices. This system can be utilized in a wide variety of areas.

Chapter one delivers a wide and basic synopsis on the project topic and relating matters like the background and motivation, risk assessment, statement of the problem, aims, objectives and significance of the project.

Chapter two is based upon the literature review for the project which consists of the hardware and software being used in the development of this project.

Chapter three covers the functional and non-functional requirements in regards to the analysis and design of the system for this project work such as the Unified Modelling Language (UML) diagrams showing the application flow and architecture and other sequences.

Chapter four entails the discussion of the development process of the project and looks at how the project was implemented and tested.

Chapter five summarizes the project indicating the problems encountered during development and iteration and how they were handled gracefully as well as plans for Continuous Development and Integration (CD/CI).

6

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1** **Introduction**

This chapter covers in detail the historical buildup of both software and hardware components which will be used in the development of this project.

For this project, the backend component will be programmed in mostly python using a framework called Django which was made for creating client-side Websites using python and has functionalities installed for creating REST APIs.

Additionally, the aforementioned programming language will be used in writing the logic behind the AI and CV which will be used in detecting objects through the sensors. For the frontend, I will be making use of JavaScript, CSS and HTML for the UI design and these two will communicate with each other with the help of a database system.

**2.2** **Historical Overview**

***a.*** ***History of Artificial Intelligence***

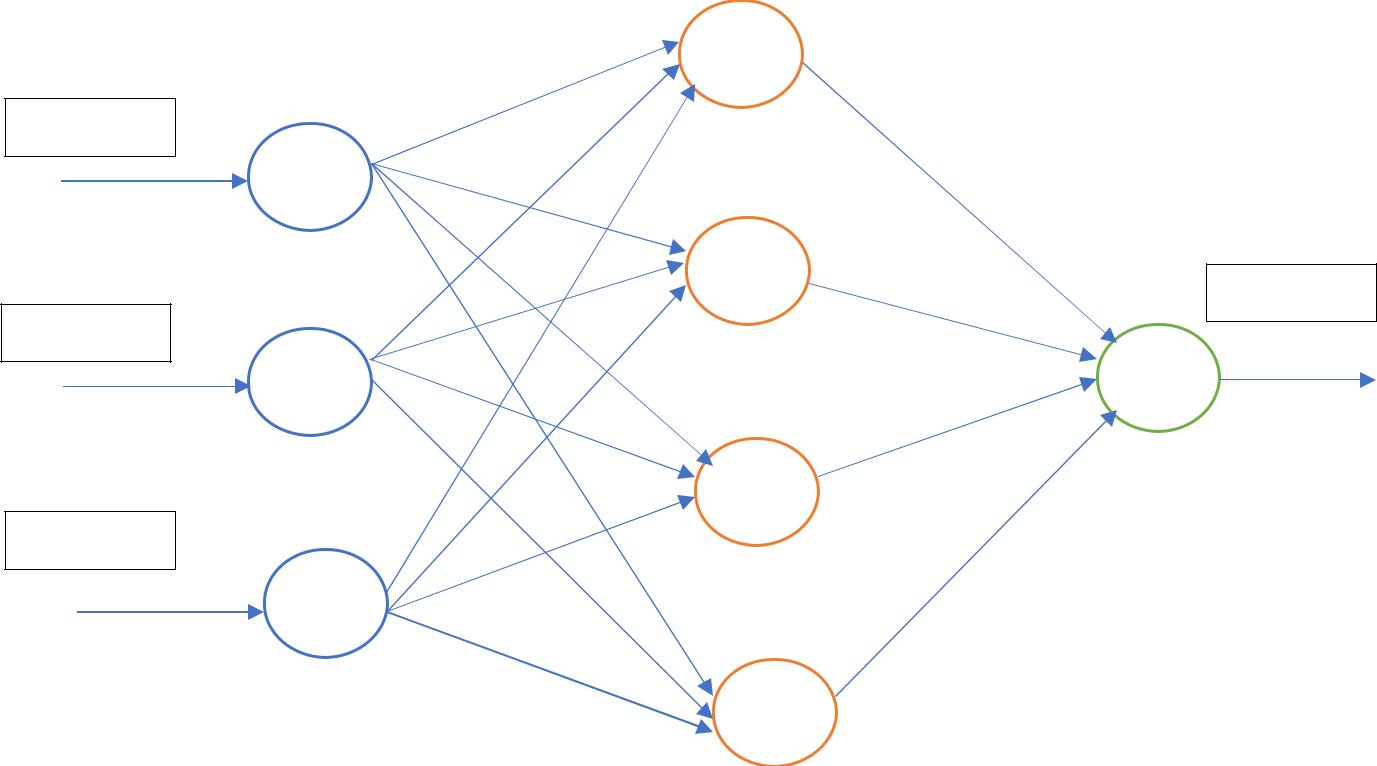
The origin of Artificial Intelligence tracks back to the 1950s, however, the thought of computers with the capability to think like man goes back to ancient Greece around 850 B.C. known as automata. There are many names that come to mind when taking about artificial intelligence but the most noticeable name is Alan Turing, who in his passion for this area of computer science went through so many years of hard work and at the same time, people did not believe in his dream, because back then, it was seeming more and more like a hoax. In 1950, Alan Turing published a paper called Computing Machinery and Intelligence in which he proposed to answer the question ‘can machines think?’ and

7

introduces the Turing Test to determine if a computer can demonstrate similar intelligence to that of human beings.

The name ‘Artificial Intelligence’ was coined by John McCarthy in the year 1956 at the first AI conference ever to hold at Dartmouth College. Later that same year, the first AI software was developed by Allen Newell, J.C. Shaw, and Herbert Simon which was called the Logic Theorist.

In 1967 the first computer based on a neural network was built by Frank Rosenblatt and this computer was called the Mark 1 Perceptron. A neural network is a software computing system which is inspired by the biological neural networks in the brain which connects the billions of neutrons to communicate with each other and make or break old or new connections respectively. An Artificial Neural Network (ANN) is based on a deep collection of interconnected nodes called artificial neurons and they are built to vaguely model the neurons of the biological brain. Below is an illustration of a typical ANN.



Input A

A

Input B

B

Input C

C

Output

*Figure 2.1 Artificial Neural Network*

The figure above gives a good description of the layout of an ANN. The circles are nodes with the ability to store data as input, process it, and display an answer as an output. A

8

good example of its working is, take inputs A, B, and C as pictures of 2 dogs, and a cat and we have assigned the ANN to tell us which picture is that of a cat. It will then run the data through the middle nodes and compare it to previous data that it has been feed. Finally, it will predict provide an answer based on its predications and its levels of certainty.

The year 1997 was a milestone in the advancement of AI. This was when IBM’s AI system called Deep Blue beat a then world class chess champion Garry Kasparov in a chess match and again in a rematch. This was the first-time machine was beating man in his own game. A similar occurrence happened in 2011 when IBM Watson—a famous AI machine—beats champions Ken Jennings and Brad Rutter at a game called *Jeopardy!* Afterwards, in 2015, a supercomputer was created which makes use of a deep neural network called convolutional neural network made for the purpose of identifying and categorizing images with a higher rate of accuracy than the average human.

Additionally, in 2016, a program developed by DeepMind called AlphaGo based upon the same architecture of deep neural network. AlphaGo attained victory over Lee Sodol, at the time the world champion Go player in a five-game match. This is another incredible milestone in the field of AI considering the fact that there is an enormous number of possible moves in the game play. A typical game depth of 150 moves, an estimation of about 250150 or 10360 possible moves.

***b.*** ***History of Microcomputers***

Microcomputers have experienced an explosion in its rate of usage and application since its invention. There are more than two billion microcomputers being produced every year and these controllers are found in various houses, homes and companies like hospitals, banks, schools, etc.

A microcontroller is an embedded computer Integrated Circuit (IC) which contains the Central Processing Unit (CPU) that controls the workings of the microcontroller which

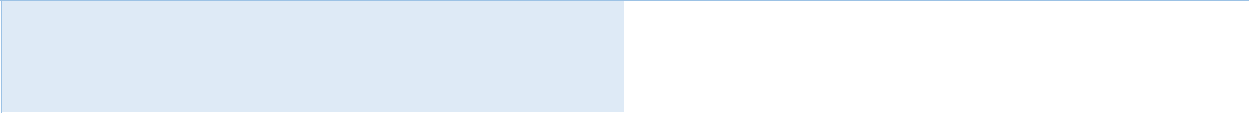
9

are mainly used whenever there is need for measurements, storing, calculation, controlling or display of information. The first microprocessor is the Intel 4004 followed by the Texas Instruments’ TMS1000. For a microprocessor to work it needs a collection of hardware and peripheral devices that can be found on a motherboard. The hardware consists of memory and ICs for peripheral devices.

The invention of this incredible chip tracks back to the years of 1970 and 1971 when Intel was working on a project to invent the first ever microcontroller, however, Gary Boone of Texas Instruments was working one something similar and beat intel to this invention and made the first microprocessor called TMS1802NC. The device had amazing capabilities with 5000 transistors providing 3000 bits of program memory and 128 bits of access memory.

As at now, microcomputers have reduced in size and increased in power, just as stated in Moore’s law. There are different companies producing microcomputers including Intel, Sony and Nvidia. Some of the most popular microcomputers are the Raspberry Pi by Sony, the Arduino, and the NVIDIA Jetson Nano amongst others. For this project I have decided to use The Jetson Nano for reasons I will explain in the related work subsection. Below is a table showing the difference between a microcontroller and a microprocessor.

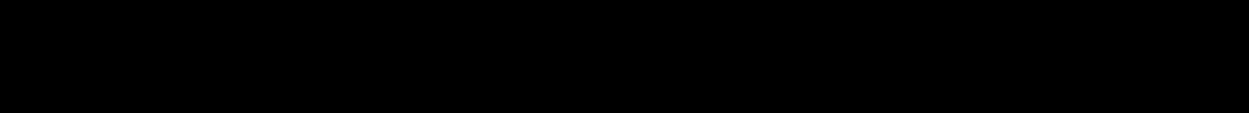
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Microcontroller |  |  | Microprocessors |  | |  | |
|  |  |  |  | |  | |
|  |  |  |  |  |  | |  | |
|  | A microcontroller is the center piece of an |  |  | **A microprocessor is the center piece of any** |  | |  | |
|  | embedded system |  |  | **Computer system.** |  | |  | |
|  |  |  |  |  |  | |  | |
|  | A microcontroller possesses a processor as |  |  | **A microprocessor is but just a processor** | |  | |
|  | well as an internal memory and input/output |  |  | **and to function fully it needs to be** | |  | |
|  | components |  |  | **integrated by memory and I/O components** | |  | |
|  |  |  |  |  | |  | |
|  | This can be used in compact systems. That is, |  |  | **You cannot use this in compact systems** |  | |  | |
|  | systems that a complete and closely packed |  |  |  |  | |  | |
|  | system. |  |  |  |  | |  | |
|  |  |  |  |  | |  | |
|  | Cost of making entire system is relatively |  |  | **Cost of making entire system is** | |  | |
|  | low. |  |  | **considerably high.** | |  | |
|  |  |  |  | |  | |  | |
|  |  | 10 | | |  | |  | |



Microcontrollers are used mainly in

embedded systems as stated above.

**Microprocessors are mainly used in personal computer systems.**



*Table 2.1*

***c.*** ***History of WWW Programming***

The World Wide Web was invented by Time Berners-Lee in 1989, roughly 20 years preceding the first connection established over what is known today as the internet. Tim was a software engineer at a company called CERN, a large physis laboratory near Geneva, Switzerland.

In October 1990, he had identified the three fundamental protocols and technologies that are the foundations of the WWW we use till date. These technologies are HTML, URL, and HTTP. Tim wrote the first Web page browser/editor and the first Web server, and by April 1993, CERN announced that the WWW technology would be available for anybody to use on royalty-free basis.

From that time till now, the Web has changed the world making it a global village and opening a plethora of opportunities to people all over the world. Moving forward there was the upgrade to Web 2.0 which was themed to be more personal and dynamic, and then Web 3.0 which is a semantic Web of linked data allowing everything in the Web to be interconnected and the possibilities for AI—in specific deep learning—systems to plough through the Web and other linked data and conduct research better than humans. As of 2008, there were over 1 trillion public pages, and 2009 had over 1.7 billion people on the Web.

Web programming and development continues to progress every year with new functionalities and methods added to the Web browsers, servers and programming languages. The internet continues to grow with different components and functionalities added to it and it will stay for years to come.

11

HTML and CSS are markup languages which are used mainly for the frontend of a Web application. You can think of the frontend of any application as the visible flesh of a human body and the backend as the internal organs and systems that allows for dynamic operations. In web development there are two main types of backend which are client-side and the server-side applications. In developing the backend of a server-side application different programming languages like Java, Python, Ruby and so forth are used in this development process. While for client-side application JavaScript and Node.js are used.

***d. OpenCV***

Similar to TensorFlow by Google and NumPy from Python; OpenCV (Open-Source Computer Vision Library) is an open-source program for computer vision and a machine learning software library originally developed by Intel with release date of June 2000 for the purpose of providing common grounds or infrastructure for computer and machine vision applications and products.

With over 2500 optimized algorithms in this library mainly for computer vision and machine learning algorithms. These algorithms can be used to recognize and detect objects, logos, faces, classify human action in videos and so on. This library is used to a large degree by different companies like Google, Yahoo, Microsoft, Intel, Sony, Honda and the list goes on.

It possesses Python, C++, JAVA and MATLAB interfaces and supports Windows, Linux, Android and Mac OS.

***e.*** ***Python***

Python is an omnipresent, general-purpose, high-level programming language which is very common amongst beginners and advanced programmers alike in the programming world. It is a fairly recent programming language which was designed initially by Guido van Rossum in the year 1991 and developed by the Python Software Foundation. The

12

main purpose of its development was code readability, and the easy syntax that allows programmers to express coding concepts in lesser lines of code.

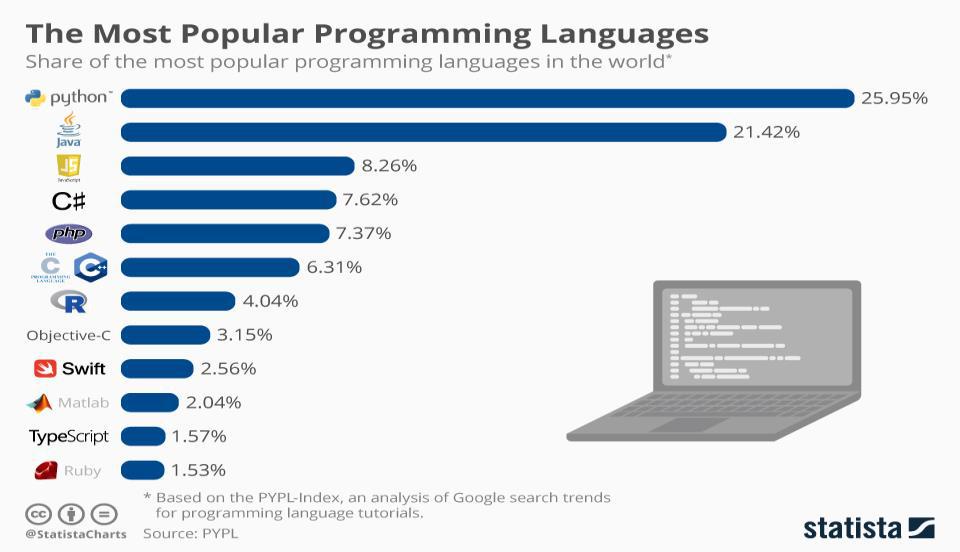
Back in the late 1980s, the landscape of the programming industry was about to be changed. This was when the work on Python had commenced.

Afterwards, Guido Van Rossum began working on the application-based project in December 1989 in a company called Centrum Wiskunde & Informatica (CWI) in Netherland. Just like the projects by Alan Turing and Tim Berners-Lee, this work started as a hobby project he used to keep him occupied during Christmas. Python is said to have succeeded the ABC Programming Language which had a feature of exception handling. Guido had helped in the creation of ABC before this, however, had also seen some issues with ABC but liked nearly all of the features on this programming language. Due to the fact that he had already worked on a similar project, he cleverly used the syntax of ABC and some of its good features and used that in the development of Python.

The inspiration for the name Python a show that he loved watching by BBC called, ‘Monty Python’s Flying Circus.’ He wanted a short, unique and mysterious name for the invention.

Python has upgraded its versions from Python 1.0 all the way to Python 3.7. Python is now the most popular coding language in the world and has also been an inspiration for several other programming languages like Ruby, Swift, OCaml, Groovy, Cobra, Julia, etc.

13



*Figure 2.2 (source Statista.com)*

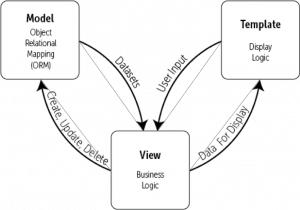
***f. Django***

Django is a web framework used to build Web applications based on the MVT (Models, Views, Templates) architecture. Django defines itself as a ‘batteries included’ Web-based framework, that has both robustness and simplicity assisting web developers to write clean, efficient and powerful code.

Django was designed and implemented by Lawrence Journal-World in the year 2003 and was finally released to the public in July 2005 through a software licensing company called BSD Licenses. As at now, the Django Software Foundation (DSF) regulates its development and release cycle and its current version is Django 3.0.

Django is a very popular framework and is used by top technology companies like Instagram, YouTube, Udemy, Google and even NASA for their website. Below is the structure of the MVT architecture.

14



*Figure 2.3 Django Architecture*

The MVT structure is a variation of the famous MVC (Model-View-Controller) structure which is used in building complex and lightweight applications. The workflow of this architectural structure starts when the Django server receives a request, which gets mapped through the URL router to the fitting view function or class. Afterwards, the view then fetches the data through the models, fill the template with HTML and CSS markup and returns it to the user. Django makes use of Object-Relational-Mapping (ORM) which is a library that implements a technique that allows one to interact with any database using a language of choice instead of SQL queries. The models in Django imports these libraries to allow developers to affect changes to a database table using Python code.

***g. PostgreSQL (Django Models)***

PostgreSQL is a state-of-the-art, efficient, and open-source object-relational database system that supports both relational (SQL) and non-relational (JSON) querying. It has been operational for over 20 years of active development and a proved architecture that has earned it a high believability rate for fata integrity, reliability and correctness. It runs

15

on all major operating systems including UNIX, Linux and Windows.It is used as the major database for several web applications as well as mobile and analytics applications.

PostgreSQL started in the year 1986 in the Berkeley Computer Science Department University of California. Initially, the project was named POSTGRES in reference to the older Ingres databases which were also developed in the Department of Computer Science in California.

At first, PostgreSQL was designed to operate on UNIX-like platforms and later evolved to running on numerous platforms such as macOS, Windows and Solaris due to the help of the dedicated community of contributors that continue to make the releases of open-source and free database projects.

***h.*** ***REST API***

An outstanding future the sets Django out of the crowd is the Django REST framework which is installed into the Python virtual environment and makes it easier to create RESTful APIs (Representational State Transfer Application Programming Interface) that can talk to both web and mobile frontend applications to add and manipulate data in the environment.

An API is an interface which calls a specific function within a program. For example, in developing a program with one functionality of delivering a message to someone, you can use an API call to do that and that function can be reused countless times in the future, this is called a function signature.

In other words, an API is an interface between the software backend and the hardware output device. Examples of APIs are Google Maps, Weather Channel, Twitter API, Facebook API, Instagram, etc. To explain further, Google Maps API can be used to embed a location on Google Maps into the Web pages as well as Mobile and Desktop applications.

**2.3** **Related Hardware**

In the microcontroller industry there are a few notable names to be mentioned due to their popularity and performance abilities. The main embedded systems in the microcontroller

16

community are the Raspberry Pi, Arduino, Google Coral, and NVIDIA Jetson Nano.

Below is a table comparing all the integrated chips.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Raspberry Pi** |  |  | **Arduino** |  |  | **Google Coral** |  |  | **Nvidia Jetson Nano** |  | |  | |
|  |  |  |  |  |  |  |  |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |
| **Type** |  | Microcomputer |  |  | Microcontroller |  |  | Microcontroller |  |  | Microcomputer |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |
| **OS** |  | Raspbian | |  | Does not need |  |  | Mendel OS, | |  | Ubuntu | |  | |
|  |  |  |  |  | OS |  |  | derivative of | |  |  |  | |  | |
|  |  |  |  |  |  |  |  | Debian | |  |  |  | |  | |
|  |  |  | |  |  |  |  |  | |  |  | |  | |
| **Difficulty** |  | Complicated |  |  | Best for |  |  | Complicated |  |  | Medium |  | |  | |
|  |  |  |  |  | Beginners |  |  |  |  |  |  |  | |  | |
|  |  |  | |  |  |  |  |  | |  |  | |  | |
| **CPU** |  | Quad core | |  | Atmel AVR |  |  | Quad core | |  | Quad core Cortex- | |  | |
|  |  | 1.2GHz | |  |  |  |  | Cortex-A53 + | |  | A57 | |  | |
|  |  | Broadcom | |  |  |  |  | low power | |  |  |  | |  | |
|  |  | BCM2837 | |  |  |  |  | CortexM4F | |  |  |  | |  | |
|  |  |  | |  |  |  |  |  | |  |  | |  | |
| **RAM** |  | 1-4 GB |  |  | 32 KB |  |  | 1GB LPDDR4 |  |  | 4 GB LPDDR4 |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |
| **GPU** |  | Broadcom | |  | None |  |  | Integrated | |  | Nvidia 128 Maxwell | |  | |
|  |  | BCM2711B0 | |  |  |  |  | Vivante GC7000 | |  | CUDA cores | |  | |
|  |  | quad-core A72 | |  |  |  |  | lite Graphics | |  |  |  | |  | |
|  |  | (ARMv8-A) | |  |  |  |  |  |  |  |  |  | |  | |
|  |  |  | |  |  |  |  |  | |  |  | |  | |
| **Programming** |  | C/C++, Python, |  |  | C/C++ |  |  | Arduino, C++ |  |  | Python, C/C++, |  | |  | |
| **Language** |  | Ruby, etc. |  |  |  |  |  |  |  |  |  |  | |  | |
|  |  |  |  |  |  |  |  |  | |  |  | |  | |
| **Cost** |  | 35 USD | |  | 16 USD |  |  | 149 USD | |  | 99 USD | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |

*Table 2.2*

In summary, I have chosen to use Nvidia Jetson Nano as it has better specs: better CPU, more RAM, more flash storage with the exception of no wireless connectivity. Finally, despite the lack of connectivity (which will be solved as explained in the chapter below) it is the best choice for individual educational purposes due to its power and versatility.

17

**2.4** **Improvements**

A problem with the Jetson Nano is its lack of wireless connectivity like Wi-Fi and Bluetooth functionalities. However, from my research this problem can be easily fixed by integrating a chip into the Embedded system. This chip is manufactured by a hardware company called Waveshare. The chip is built specifically for the purpose of WI-FI and Bluetooth functionalities in the Jetson Nano. It comes with an extension cable, nuts and antennas.

**2.5** **Summary**

In this chapter, I have explained in detail the literature review and related possible work regarding this project.

18

**CHAPTER THREE**

**REQUIREMENTS ANALYSIS AND DESIGN**

**3.1** **Overview**

This chapter narrates both the functional and none functional requirements for this project in addition to the methodologies, procedures and UMLs used during the making of this work.

**3.2** **Proposed Methodology**

There are different methodologies in software development including Waterfall model, AGILE model, Iterative development model and so on.

For this project, I will be using the Agile SDLC (Software Development Life Cycle) model to start and complete the development life cycle.

Agile SDLC model is an amalgamation of the incremental and iterative process models with its main focus on customer satisfaction and product adaptability through the rapid delivery of working software and (in this case) hardware products to the market. Agile methods are broken down into bit size incremental constructs. These structures are dispensed in the form of iterations or functions. On average, each iteration usually ranges for one to three weeks depending on the size and complexity of the project. Below, I have enumerated the list of iterations for this methodology:

1. Planning;
2. Requirements Analysis;
3. Design;
4. Coding;
5. Unit Testing; &
6. Acceptance Testing

19

Once this list of iterations is completed, a working product is displayed to the target market and important stakeholders. This is however not the end of the process. In future iterations, improvements will be made on the existing version and depending on the feedback received from the stakeholders/users. A complete iteration process is completed every two to three months.

Below are some Agile Manifesto principles created and followed by Agile Developers—

* **Individuals and interactions** – These are a principle that places self-organization and motivation (individuals) as proprieties alongside co-location and pair programming (interactions).
* **Working Software** – A Minimum Viable Product (MVP) is a product with enough features to attract early adopters and validate a product in the market instead of depending just on documentation.
* **Customer collaboration** – Continuous customer interaction in the form of surveys and one-on-one interactions from the beginning of the project is vital to collect correct product requirements.
* **Responding to change** – This type of development life cycle is centered on quick responses to change and continuous integration and development.

**Difference Between Agile and Traditional SDLC Models**

The foundation of Agile development is based upon adaptive software development methods, whilst the traditional SDLC models, for example, waterfall model are based upon a predictive approach.

**3.3** **Approach to Chosen Methodology/Methods**

The agile method has different techniques and procedures. Some of these techniques include but are not limited to: Agile Scrum Methodology, Extreme programming,

20

Dynamic Systems development method, crystal and future driven development. The method taken in the development of this system is the Lean and Kanban Software development.

Lean Kanban brings together two fields of process improvement called Lean and Kanban.

Let’s look at each of them in turn.

Lean manufacturing aims to:

* **Reduce waste.** This makes abundant sense for a repetitive, repeatable process.
* **Increase efficiency.** This is typically achieved by buying newer, more up-to-date machinery.
* **Eliminate bottlenecks.** This is achieved by restructuring the manufacturing workflow.

When it comes to applying this in software development, we need to think a little differently to get maximum value from these underlying Lean principles, and to make the terminology more palatable to digital teams.

**3.4** **Tools and Techniques**

Below I have enumerated some of the necessary tools, including software and hardware that were used in the development of this system.

* **Visual Studio**: This software is the most popular code editing software amongst developers and programmers. It supports a wide range of programming languages and is very intuitive and straight forward to use.
* **Python**: In the field of Artificial Intelligence, in specific computer vision and deep learning, the two dominating programming languages are Python and C++. I have chosen to used Python as it has a wider range of use and application for different tasks and projects.

21

* **OpenCV**: As defined in chapter 2, OpenCV is a free open-source software which is used by developers around the world to apply AI to images and videos to perform measurement and validations.
* **TensorFlow**: This is another opensource technology which aids to compute big data which in turn aids to develop and train open-source models.
* **FLIR Lepton 3.5**: This is a high-resolution micro thermal camera module. This component will be used for thermal imaging of both people and objects which has a high potential of applications form detecting fever, to scanning environments and things for heat signatures for security purposes.
* **PostgreSQL**: During data collection, the data will be stored and sent to the API through the aid of the PostgreSQL database.
* **Nvidia Jetson Nano**: This is a powerful microcomputer with high GPU processing capacities which allows for faster parallel processing. Parallel processing is a method in computing of running more than one processor to handle separate components of an overall assignment.
* **Raspberry Pi**: This is another microcomputer that is lower in computational power and hence is relatively slower when computing the large computer vision algorithms. Through SSH it can be controlled by the Nvidia Jetson nano and this will allow for distribution of computational power during execution of such algorithms.
* **Jumper Wires Lights and Sound System**: Male and female jumper wires are used to connect fans and push buttons to enable cooling mechanisms and a simple button to turn the systems on and off. The wires are connected to the GPIO

(General Purpose Input Output) pins and I2C serial connectors making the system more whole and coherent. In addition, the system will work with LED lights and a

22

sound system this will be set up as an alarm against potential threats and breaches of protocol.

**3.5** **Requirement Analysis**

In the next subsection, we will discuss the breakdown of the requirements for the CV and deep learning algorithms and this implementation to the web application built.

**3.6** **Requirements Specifications**

The requirement specification for this project has been divided into two main parts namely functional and non-functional. They are defined and enumerated in the tables that follow.

***a. Functional Requirement Specifications***

Functional requirements simply defined are the product features and functionalities that developers must implement to enable users to accomplish their tasks. Below is the table of functional requirements.

|  |  |  |
| --- | --- | --- |
| **Reg. No** | **Description** | **Type** |
|  |  |  |
| **R-101** | This application will be able to run from any device that has | Functional |
|  | a web browser. |  |
|  |  |  |
| **R-102** | The application shall include a user interface. | Functional |
|  |  |  |
| **R-103** | The application shall allow admin users to take and make | Functional |
|  | records and work with the visual data provided. |  |
|  |  |  |
| **R-104** | The application shall stream video to web browser/HTML | Functional |
|  | page in real time. |  |
|  |  |  |

23

|  |  |  |
| --- | --- | --- |
| **R-105** | The algorithm shall be able to perform object detection and | Functional |
|  | image classification, for example, face mask detection. |  |
|  |  |  |
| **R-106** | The algorithm shall be capable of performing human face | Functional |
|  | and body detection. |  |
|  |  |  |

*Table 3.1*

***b.*** ***Non-functional Requirement Specifications***

These are the requirements that describe the characteristics that a product must have to meet the needs of the stakeholders and the business itself.

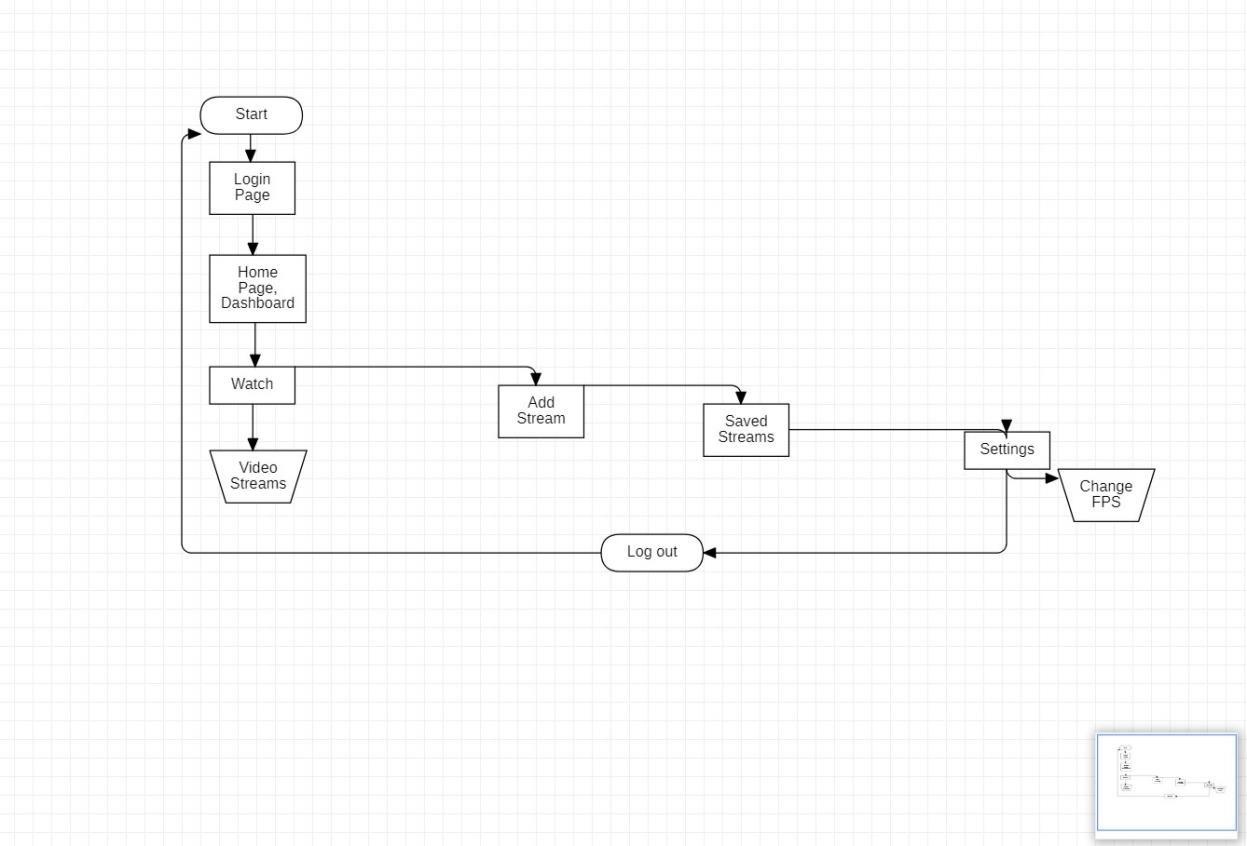
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Reg. No** |  |  | **Description** |  |  | **Type** |  | |  | |
|  |  |  |  |  |  | |  | |
|  |  |  |  |  |  |  |  |  | |  | |
|  | **R-101** |  |  | When launched the program will stay running given that |  |  | Performance |  | |  | |
|  |  |  |  | there is no manual shutdown of the application or its server. |  |  |  |  | |  | |
|  |  |  |  |  |  |  |  | |  | |
|  | **R-102** |  |  | The application will have an IP address. | |  | Performance | |  | |
|  |  |  |  |  | |  |  | |  | |
|  | **R-103** |  |  | The application will keep count of number of people that |  |  | Performance |  | |  | |
|  |  |  |  | violate social distancing rules. |  |  |  |  | |  | |
|  |  |  |  |  |  |  |  | |  | |
|  | **R-104** |  |  | The microcomputer will remain on till switched off | |  | Performance | |  | |
|  |  |  |  | intentionally. | |  |  |  | |  | |
|  |  |  |  |  | |  |  | |  | |
|  | **R-105** |  |  | Admin password will be protected and can only be seen by |  |  | Security |  | |  | |
|  |  |  |  | them. |  |  |  |  | |  | |
|  |  |  |  |  |  |  |  | |  | |
|  | **R-106** |  |  | The application will display a message if connection to | |  | Performance | |  | |
|  |  |  |  | microcomputer fails. | |  |  |  | |  | |
|  |  |  |  | | |  |  |  | |  | |
|  |  | *Table 3.2* | | | |  |  |  | |  | |

24

**3.7** **System Design**

This section of the chapter talks over the general purpose, developmental, modeling, language in the field of software engineering, called UML through the use of well thought out and designed illustrations depicting the internal workings of this application, including the software and hardware.

***a.*** ***Application Architecture***

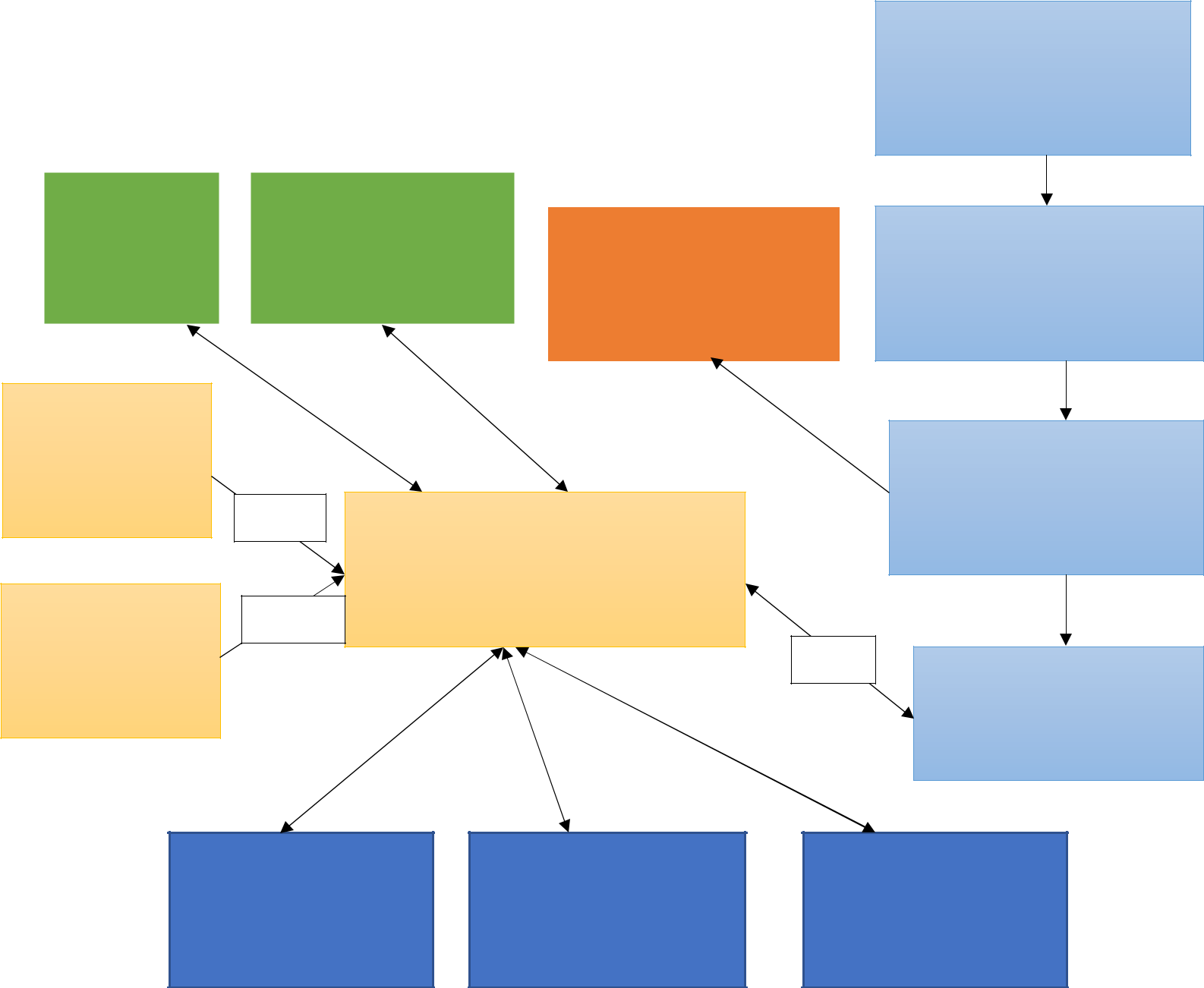


*Figure 3.1*

25

***b.*** ***Component Block Diagram***

This is the layout of all the hardware for the main part of the embedded system.



AC Power Supply

IP camera

Thermal Heat Camera

System Voltage Regulator

Battery Charger

Start Button

FR Mode/ Reset

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | I2C |  |
| SPI |  |  |
|  |  |  |
|  |  |  |  |

GPIO

Microcontroller

GPIO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | GPIO |  |
| UART |  | UAR |  |  |
|  |  |  |  |
|  |  |  |  |  |  |

I2C

LiPo Battery

Battery Monitor

WIFI / WIFI Direct

Bluetooth Low-Energy

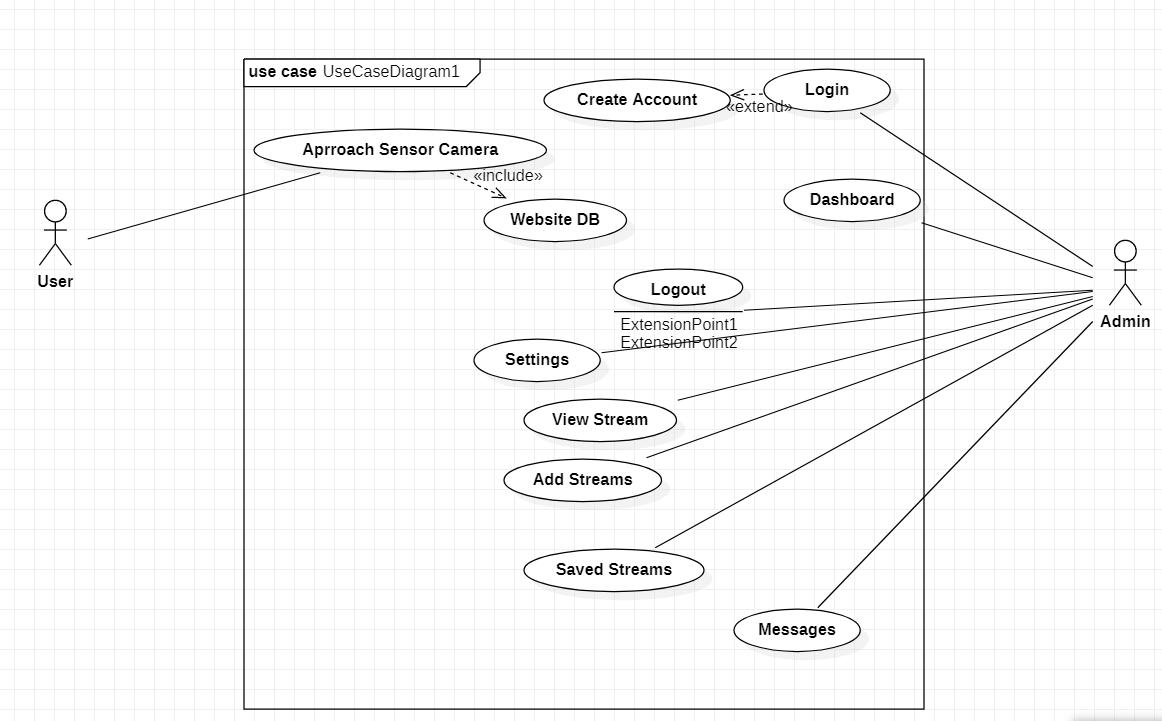
Heat Fans Connected To Heat Sink & Body

*Figure 3.2*

26

Where: Serial Peripheral Interface (SPI); Universal Asynchronous Receiver / Transmitter (UART); I2C is a serial communication protocol.

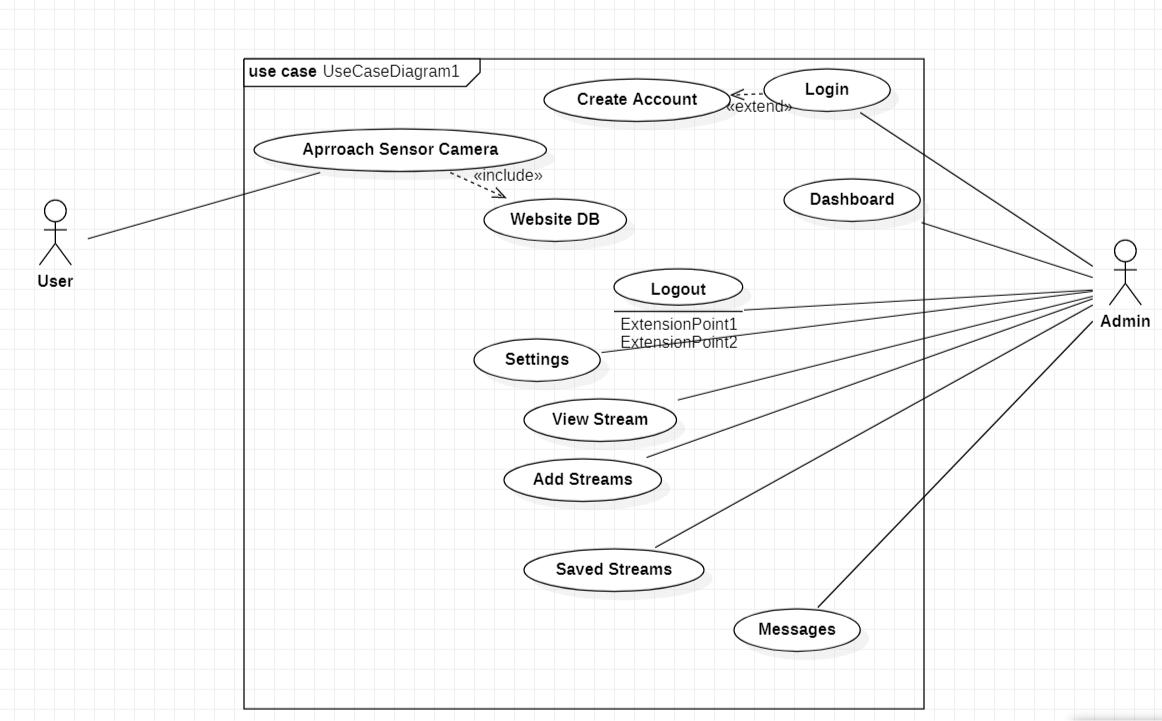
1. ***Use Case***



***Figure 3.3***

27

***d.*** ***Activity Diagram***



***Figure 3.4***

**3.8** **Summary**

This chapter covers in detail all the requirement analysis and design specifications in the making of this project.

The next chapter covers the implementation techniques and tools used in the development of the embedded system.

28

**CHAPTER FOUR**

**IMPLEMENTATION AND TESTING**

**4.1** **Overview**

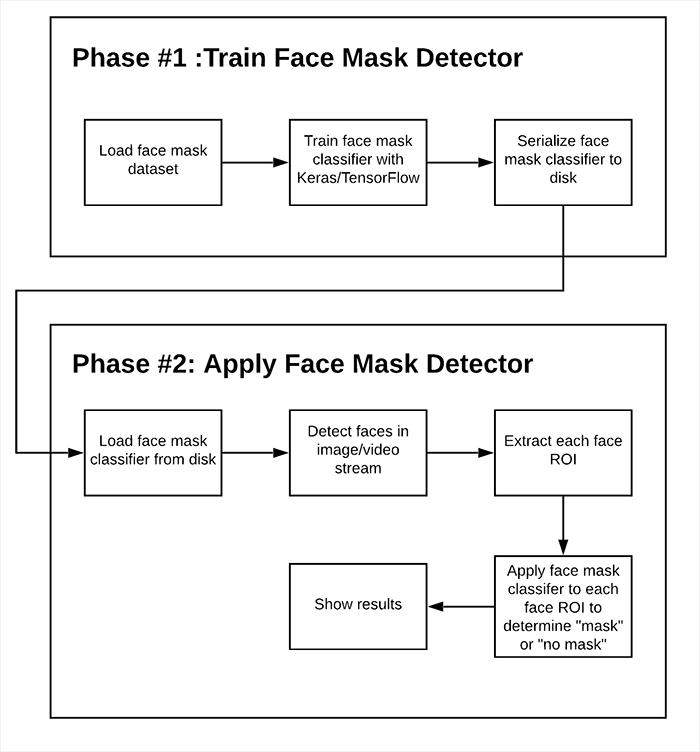
At the end of this chapter, I hope to have given a clear and concise layout of the structure of the AI algorithms used, the layout of the code and testing used in the development of this project.

**4.2** **Main Features**

***a.*** ***Face Mask Detector***

The face mask detector is an application that constitutes of OpenCV, Keras/TensorFlow and Deep Learning algorithms and code which is broken down into two phases. This application will be able to detect COVID-19 face masks in images and detect face masks in real-time video streams after computing through a data set of people wearing masks people not wearing masks.

29



*Figure 4.1 Face mask detector flow diagram*

As shown in the image above, the algorithm is divided into two parts. The first phase has to do with training the program on the computer to be able to tell if a person is wearing a face mask or not.

Then, the second phase has to do with computer vision and using the trained model to predict to a good accuracy if someone has a facemask on.

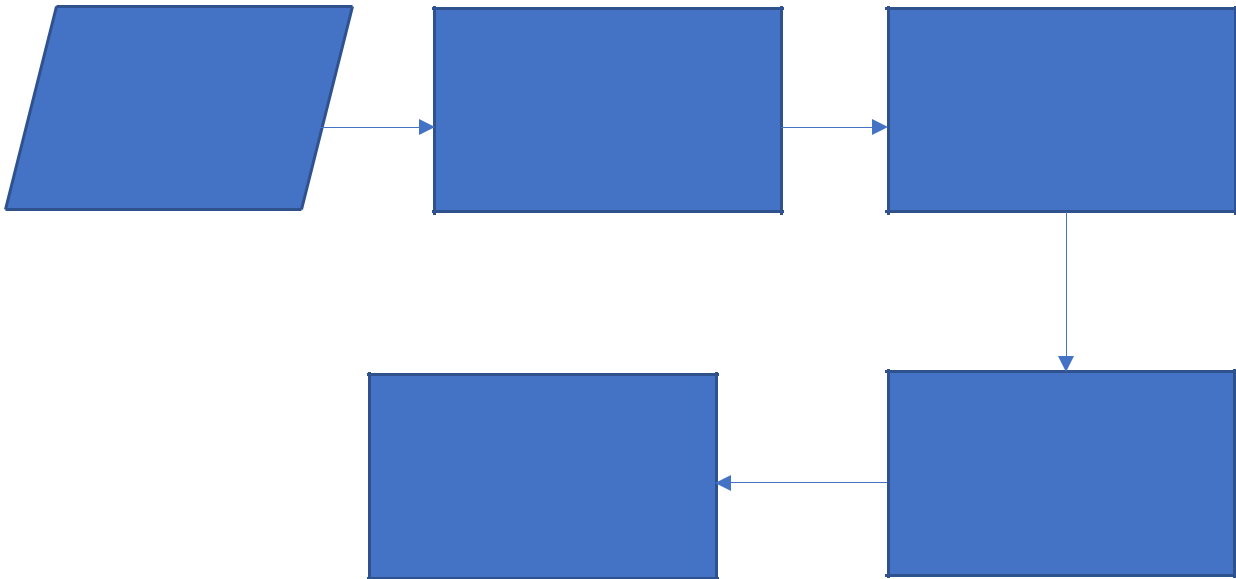
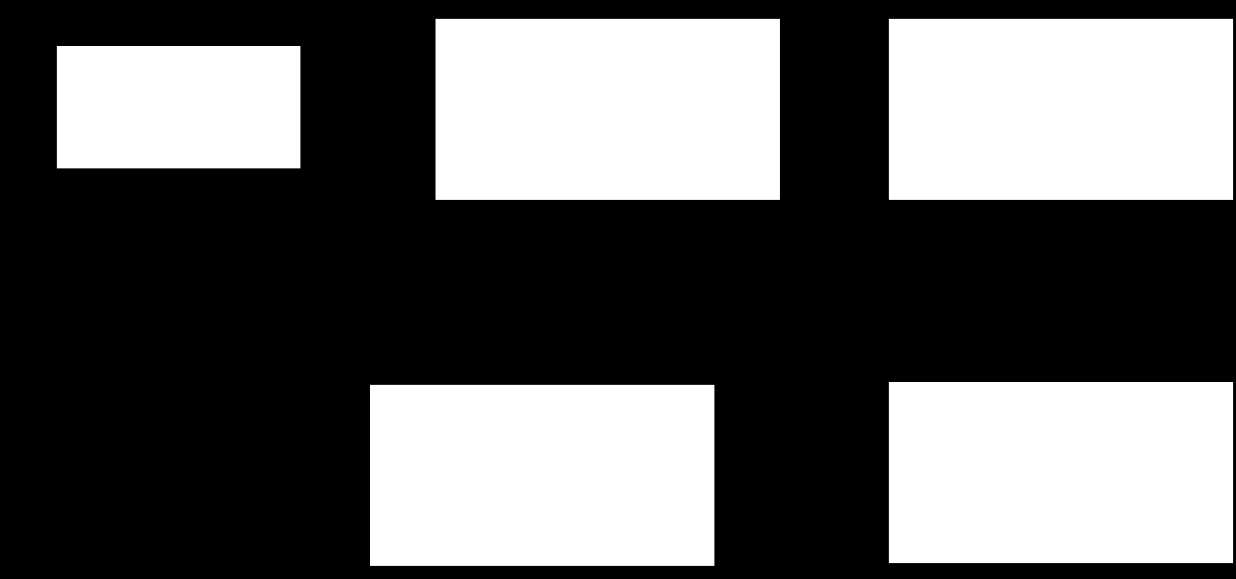
30

***b.*** ***Assisted Social Distancing***

This program is very much similar to the first algorithm. The difference here is that the program detects a human body instead of a human face. It then measures the proximity between a group of people (more than 30 at a time) by computing pairwise distances between centroids of the human body. Below is the flowchart diagram for the algorithm.



|  |  |  |  |
| --- | --- | --- | --- |
|  | **AI Assisted Social Distancing** |  |  |
| Input | Step 1: Object | Step 2: Compute |  |
| Pairwise Distances |  |
| Image/Frame | Detection |  |
| Between Centroids |  |
|  |  |  |
|  |  | Step 3: Check Distance |  |
|  | Display Result | Matrix for people < N |  |
|  |  | Pixels Apart |  |



*Figure 4.2 AI assisted social distancing flowchart*

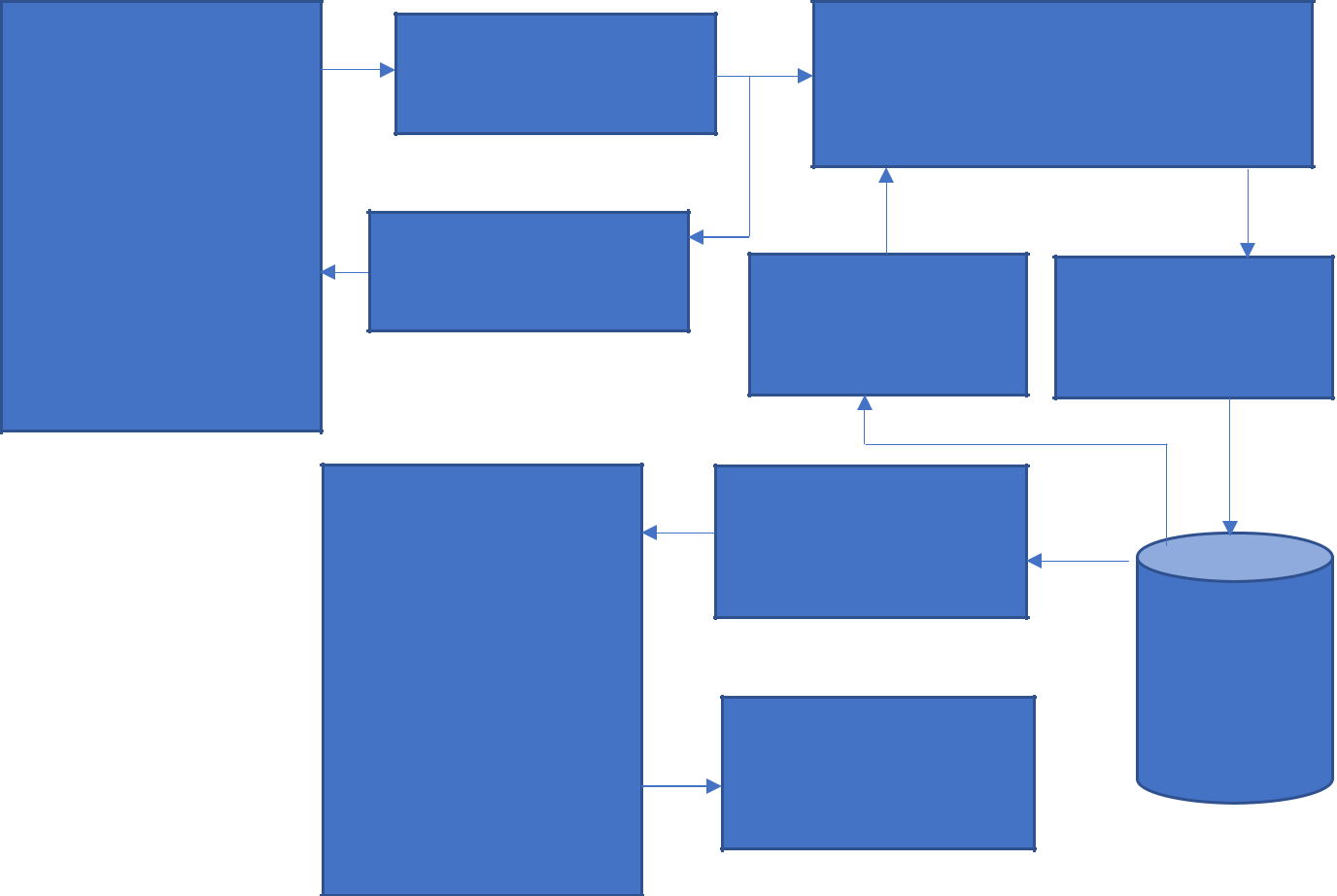
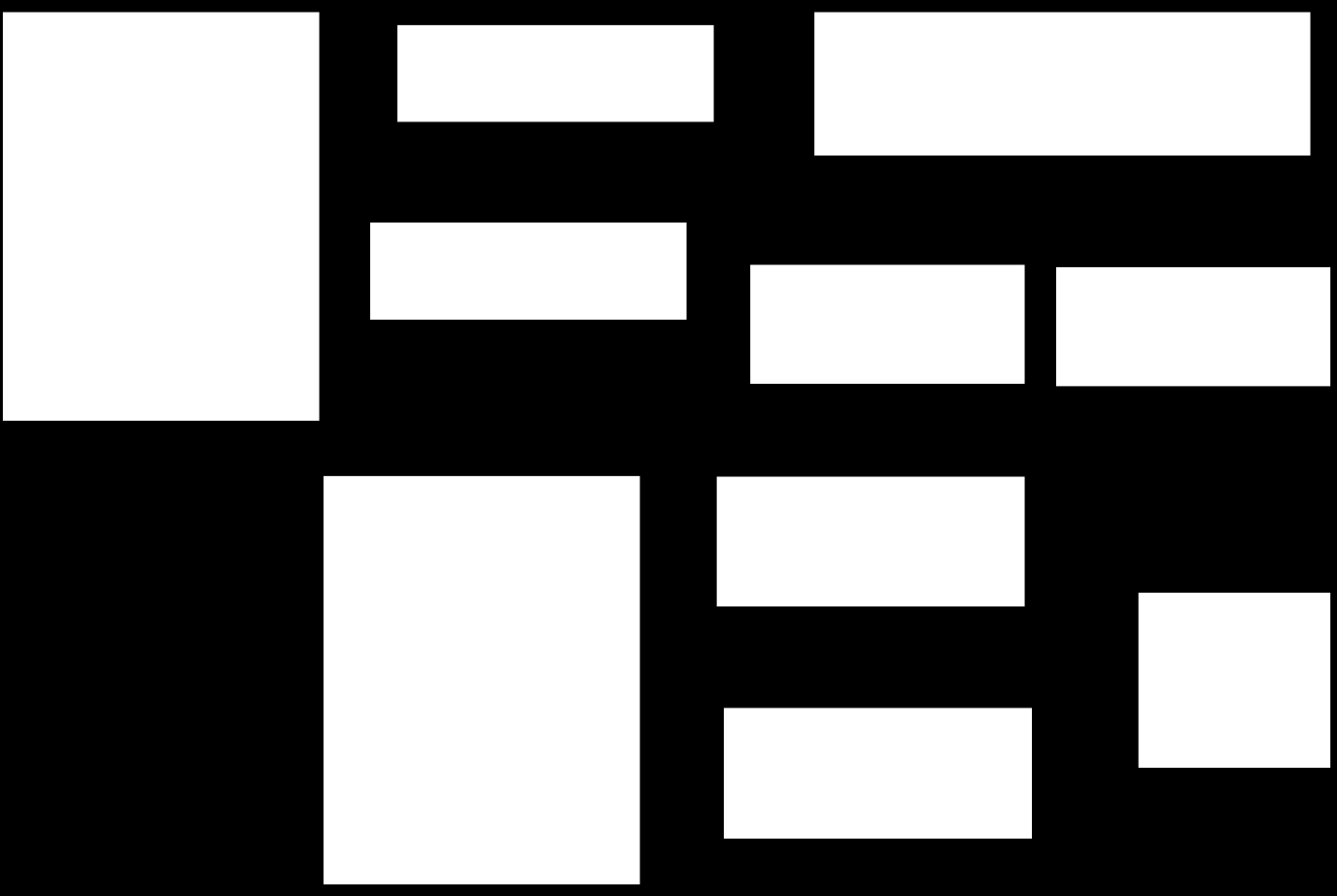
***c. Website***

The website stack is an essential part of the project as it serves as the Human Machine Interface (HMI) between the AI algorithms and the user (Admin). It integrates all the technologies used in this project into one system that is accessible from any device with access to the internet, secure web browser and account login details.

31



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Deep Learning Web App** | |  | |  | |
|  | Client sends data to | Web Server (Django) | |  | |
|  | web server |  | |
| Client (cURL, Python |  |  |  | |  | |
| app, etc.) | Web server sends |  |  | |  | |
|  |  |  | |  | |
|  | results to client | Web server polls | Web server stores | |  | |
|  |  |  | |
|  |  | for results | data in queue | |  | |
|  |  | Model server Polls |  | |  | |
|  |  | batch of data form |  | |  | |
|  |  | queue |  | |  | |
|  | Model Server/Process |  | Message | |  | |
|  |  | Broker | |  | |
|  | (Keras) |  |  | |
|  |  | (Redis) | |  | |
|  |  |  |  | |
|  |  | Model server stores |  | |  | |
|  |  | results in database |  | |  | |



*Figure 4.3 Website block diagram*

The diagram above shows the structure of the web stack application. This block diagram above shows the API of the website which constitutes the back-end development model.

The next image below shows the outline of the front-end and full stack structure of the web application.

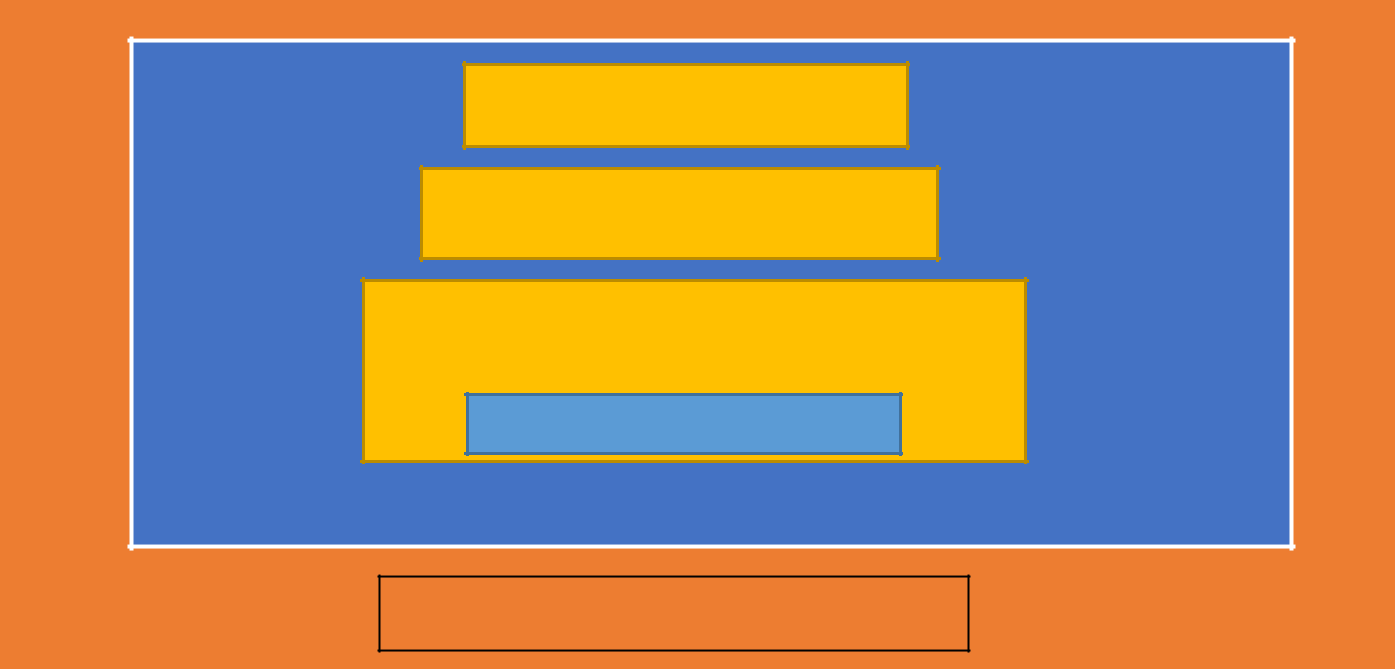
32



User (Admin)



Web Browser



Web Server

Application Server

**Application Framework**

APPLICATION

Server Operating System

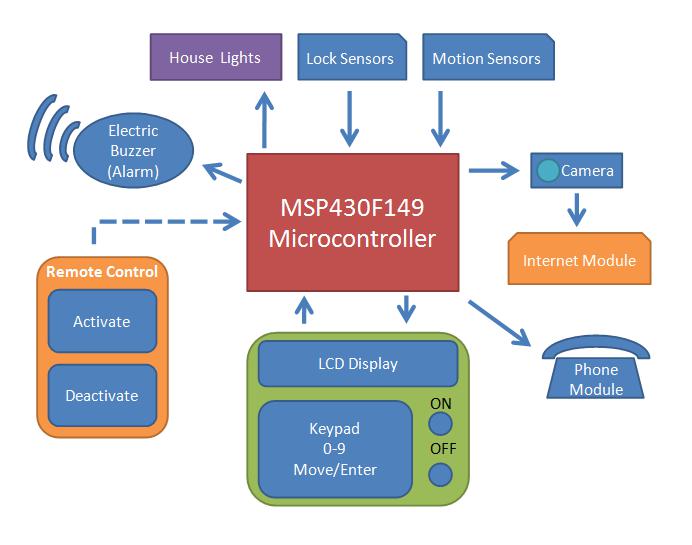
Nvidia Jetson Nano Hardware

*Figure 4.4 Website Application Stack*

***d.*** ***Alarm System***

This is the final and just as important component of this system. It is an alarm system designed to trip off when a protocol is breached, for example, when someone is not wearing a face mask. Below is the block diagram for the system.

33



*Figure 4.5 Alarm system block diagram*

The design compromises of LED lights, jumper wirers (for connectivity), sound system, etc., which are all integrated with the microcontroller to allow manipulation from the software. This alarm will be connected to various external electronic devices like a phone module, lock and motion sensors, and depending on the settings, the lighting of a building.

**4.3** **Problems Encountered**

During the development of this project work, there were a number of problems encountered and obstacles in the way which called for creativity, critical thinking and intensive research work.

The first and most important problem was the issue of lack of time. Due to the lockdown, the school’s calendar was affected and that disrupted the ongoing schedule of the final year and the whole academic institution. Working with the new time frame meant that the development cycle had to be sped up.

34

Another issue I encountered is with the hardware parts I ordered from Amazon. It came in three batches, I was able to receive the first two packages, however, the third package was held back by the Nigerian Customs.

Hence, I wasn’t able to complete the project as I had planned for.

As I am a novice in the field of computer vision, I faced multiple problems in the installation of packages and libraries to support machine learning and deep learning. So many times, my device crashed and I had to start the installation process from start.

**4.4** **Solving Problems**

As for providing solutions for the problems, I had to be resourceful and provide means to walk around and through the obstacles. Many times, this meant that they turned into advantages for me, showing me new perspectives and pathways to arrive at an objective. I learnt new methods for testing programs and algorithms.

Also, this was a great experience for me to become a better programmer, developer and problem solver in general.

**4.5** **Testing**

In order to ensure that all the programs are working as planned there will be a set of tests on the system and the algorithms to spot and fix potential bugs and errors in datasets. Below is the test layout table used in the project development.

35

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test** |  | **Test Case** | **Test Steps** | | **Test Data** | **Expected Results** | |
| **Case** |  |  |  |  |  |  |  | |
| **No** |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  | |
| **1.** |  | **Connect to** | **1.** | **Switch on Microcomputer** | **IP address:** | • | **LED indicator showing** | |
|  |  | **microcomput** | **2.** | **Run Server Python Script** | **10.0.0.90:800** |  | **power status** | |
|  |  | **er (Jetson** |  | **containing CV/DL** | **0** | • **Website display with video** | |
|  |  | **Nano/** |  | **algorithms** |  |  | **stream** | |
|  |  | **Raspberry pi)** | **3.** | **Launch website** |  |  |  | |
|  |  |  |  | **application** |  |  |  | |
|  |  |  | **4.** | **Verify Address and port** |  |  |  | |
|  |  |  |  |  |  |  |  | |
| **2.** |  | **Social** | **1.** | **Run python line in** | **Output.avi** | • | **Squared boxes around** | |
|  |  | **Distancing** |  | **terminal:** | **build up** |  | **people. Red box shows no** | |
|  |  | **Algorithm** |  | ***Time python*** |  |  | **social distancing and green** | |
|  |  |  |  | ***social\_distance\_detector.p*** |  |  | **shows social distancing.** | |
|  |  |  |  | ***y –input pedestruans.mp4*** |  |  |  | |
|  |  |  |  | ***–output.avi –display 0*** |  |  |  | |
|  |  |  | **2.** | **Watch and analyze video** |  |  |  | |
|  |  |  |  | **stream** |  |  |  | |
|  |  |  |  |  |  |  |  | |
| **3.** |  | **Face Mask** | **1.** | **Train dataset in terminal** | **Training Loss** | • | **Accuracy report** | |
|  |  | **Detection** |  | **to get model for future** | **and Accuracy** | • **Training loss and accuracy** | |
|  |  | **Algorithm** |  | **training.** | **on COVID-19** |  | **graph display.** | |
|  |  |  |  |  | **Dataset graph** | • **Box over human face. Red** | |
|  |  |  |  |  |  |  | **shows no facemask, green** | |
|  |  |  |  |  |  |  | **shows there is a face mask.** | |
|  |  |  |  |  |  |  |  | |
|  | *Table 4.1* | |  |  |  |  |  | |

36

**4.6** **Use Guide**

Once the administration user registers an account and inputs their password, they will have full access to all video streams and data. The admin user will be able to take recordings and save them; add new streams to the video collection in the add stream section; download the data analytics from the video recordings and more features.

Specific data will be saved for people that breach protocol, like screenshots of their faces and any records available about them.

Afterwards, the admin user can log out of the web application add log back in at any time. The login details can be changed from the settings.

**4.7** **Summary**

So far, this chapter has summarized the implementation process of the development life cycle of the project.

In the next chapter I will discuss about the conclusions and final recommendations for the project.

37

**CHAPTER FIVE**

**CONCLUSION**

**5.1** **Overview**

This chapter of the thesis will discuss project’s objectives, how well they were met and the final recommendations and comments on this body of work.

**5.2** **Achievement**

In my estimation, this project hit many of its milestones and long terms goals. My main objective for this work was to create an integrated system that will help reduce and track the spread of contagious viruses and influenza alike. This was achieved by developing AI and CV codes and algorithms that are trained to check that protocols such as wearing face masks and keeping social distance are met. In this way it can do the work that was previously done by humans with a high rate of certainty and efficiency.

Another major achievement was crossing the bridge from the development phase to production phase by assembling all the computer barebones parts and components into a fully functioning system capable of computing large data sets for artificial intelligence with a wide range of application in the real world.

**5.3** **Limitations and Challenges**

As a result of the problems stated in section 4.3 of this thesis, there were some constraints as to how far I could go with this project in regards to its completion.

Due to the limits in time and resources, I was not able to fully complete the website development process.

38

However, I was able to run multiple tests to ensure that the video streams were able to play from a web browser.

Furthermore, the alarm and sound system, the thermal camera and some other components did not arrive as they were held back by Nigerian Customs. These all contributed to the setbacks encountered during the development lifecycle.

**5.4** **Continuous Development and Integration (CD/CI)**

As earlier stated, this is an MVP (minimum viable product) meaning that it is at its beginning stages in the development life cycle. There are many iterations of continuous development and integration for this project as technology is always improving.

Below is a list of future development and alterations for this project:

* Fully integrated Alarm System
* Integration of thermal camera for heat signatures
* Development of security software
* Incorporation of wheels and engine to make this an autonomous movable system (something like an AI vacuum cleaner) this is constantly moving around running checks.
* Creation of a mobile application for both Android and Apple devices.

**5.5** **Summary**

The chapter discusses about the successes and setbacks of the project work and provides some insight on how these obstacles were fixed and some further enhancements that could be made in the near future.

39

**References**

* Tanner, G., 2021. *Getting Started with NVIDIA Jetson Nano Developer Kit*. [online] Gilberttanner.com. Available at: <https://gilberttanner.com/blog/jetson-nano-getting-started> [Accessed 1 June 2021].
* Education, I., 2021. *What is Artificial Intelligence (AI)?*. [online] Ibm.com. Available at: <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence> [Accessed 7 June 2021].
* Cds.cern.ch. 2021. [online] Available at: <https://cds.cern.ch/record/400313/files/p21.pdf> [Accessed 7 June 2021].
* NVIDIA Developer. 2021. *Getting Started with Jetson Nano Developer Kit*. [online] Available at: <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write> [Accessed 1 June 2021].
* SearchDataCenter. 2021. *What is Parallel Processing?*. [online] Available at: <https://bit.ly/34uU9u9> [Accessed 1 June 2021].
* Developer.mozilla.org. 2021. *Django introduction - Learn web development | MDN*. [online] Available at: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction> [Accessed 1 June 2021].
* Runestone.academy. 2021. *1.2. History — Fundamentals of Web Programming*. [online] Available at: <https://runestone.academy/runestone/books/published/webfundamentals/WWW/history. html> [Accessed 1 June 2021].
* Exyte.com. 2021. *A brief history of Python*. [online] Available at: <https://exyte.com/blog/a-brief-history-of-python> [Accessed 1 June 2021].
* Lepton.flir.com. 2021. *Getting started – Lepton Maker Community*. [online] Available at: <https://lepton.flir.com/getting-started/> [Accessed 1 June 2021].
* Inviqa. 2021. *An introduction to Lean Kanban software development*. [online] Available at: <https://inviqa.com/blog/introduction-lean-kanban-software-development> [Accessed 8 June 2021].
* AltexSoft. 2021. *Functional and Nonfunctional Requirements: Specification and Types*.

[online] Available at: <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/> [Accessed 1 June 2021].

* Safe Home Security System Group. 2021. *Block Diagram*. [online] Available at: <https://securitygroupicom5217.wordpress.com/block-diagram/> [Accessed 1 June 2021].
* Development of building security integration system using sensors, microcontroller and GPS (Global Positioning System) based android smartphone - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-block-diagram-of-Building-Security-system\_fig2\_323803855 [accessed 1 Jun, 2021]
* Henshaw, J., 1979. *This is our chance*. London: Hodder and Stoughton.

40

**APPENDICES**

**APPENDIX A**

**Source Codes**

# import the necessary packages

from tensorflow.keras.preprocessing.image import ImageDataGenerator from tensorflow.keras.applications import MobileNetV2 from tensorflow.keras.layers import AveragePooling2D

from tensorflow.keras.layers import Dropout

from tensorflow.keras.layers import Flatten

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Input

from tensorflow.keras.models import Model

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input from tensorflow.keras.preprocessing.image import img\_to\_array from tensorflow.keras.preprocessing.image import load\_img

from tensorflow.keras.utils import to\_categorical

from sklearn.preprocessing import LabelBinarizer

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report

from imutils import paths

import matplotlib.pyplot as plt

import numpy as np

import argparse

import os

* construct the argument parser and parse the arguments ap = argparse.ArgumentParser() ap.add\_argument("-d", "--dataset", required=True, help="path to input dataset")

ap.add\_argument("-p", "--plot", type=str, default="plot.png", help="path to output loss/accuracy plot") ap.add\_argument("-m", "--model", type=str, default="mask\_detector.model",

help="path to output face mask detector model") args = vars(ap.parse\_args())

* initialize the initial learning rate, number of epochs to train for,
* and batch size

41

INIT\_LR = 1e-4

EPOCHS = 20

BS=32

* grab the list of images in our dataset directory, then initialize
* the list of data (i.e., images) and class images

print("[INFO] loading images...")

imagePaths = list(paths.list\_images(args["dataset"]))

data = []

labels = []

* loop over the image paths

for imagePath in imagePaths:

* extract the class label from the filename label = imagePath.split(os.path.sep)[-2]
* load the input image (224x224) and preprocess it image = load\_img(imagePath, target\_size=(224, 224)) image = img\_to\_array(image)

image = preprocess\_input(image)

* update the data and labels lists, respectively

data.append(image)

labels.append(label)

* convert the data and labels to NumPy arrays data = np.array(data, dtype="float32") labels = np.array(labels)
* perform one-hot encoding on the labels

lb = LabelBinarizer()

labels = lb.fit\_transform(labels)

labels = to\_categorical(labels)

* partition the data into training and testing splits using 80% of
* the data for training and the remaining 20% for testing (trainX, testX, trainY, testY) = train\_test\_split(data, labels, test\_size=0.20, stratify=labels, random\_state=42)
* construct the training image generator for data augmentation aug = ImageDataGenerator(

rotation\_range=20, zoom\_range=0.15, width\_shift\_range=0.2, height\_shift\_range=0.2, shear\_range=0.15, horizontal\_flip=True, fill\_mode="nearest")

$ python train\_mask\_detector.py --dataset dataset

[INFO] loading images...

42

-> (trainX, testX, trainY, testY) = train\_test\_split(data, labels,

(Pdb) labels[500:]

array([[1., 0.],

[1., 0.],

[1., 0.],

...,

[0., 1.],

[0., 1.],

[0., 1.]], dtype=float32)

(Pdb)

* load the MobileNetV2 network, ensuring the head FC layer sets are
* left off

baseModel = MobileNetV2(weights="imagenet", include\_top=False, input\_tensor=Input(shape=(224, 224, 3)))

* construct the head of the model that will be placed on top of the
* the base model

headModel = baseModel.output

headModel = AveragePooling2D(pool\_size=(7, 7))(headModel)

headModel = Flatten(name="flatten")(headModel)

headModel = Dense(128, activation="relu")(headModel)

headModel = Dropout(0.5)(headModel)

headModel = Dense(2, activation="softmax")(headModel)

* place the head FC model on top of the base model (this will become
* the actual model we will train)

model = Model(inputs=baseModel.input, outputs=headModel)

* loop over all layers in the base model and freeze them so they will
* \*not\* be updated during the first training process

for layer in baseModel.layers:

layer.trainable = False

# compile our model

print("[INFO] compiling model...")

opt = Adam(lr=INIT\_LR, decay=INIT\_LR / EPOCHS)

model.compile(loss="binary\_crossentropy", optimizer=opt,

metrics=["accuracy"])

* train the head of the network print("[INFO] training head...") H = model.fit(

aug.flow(trainX, trainY, batch\_size=BS), steps\_per\_epoch=len(trainX) // BS, validation\_data=(testX, testY),

validation\_steps=len(testX) // BS,

epochs=EPOCHS)

43

* make predictions on the testing set print("[INFO] evaluating network...") predIdxs = model.predict(testX, batch\_size=BS)
* for each image in the testing set we need to find the index of the
* label with corresponding largest predicted probability

predIdxs = np.argmax(predIdxs, axis=1)

# show a nicely formatted classification report

print(classification\_report(testY.argmax(axis=1), predIdxs,

target\_names=lb.classes\_))

# serialize the model to disk

print("[INFO] saving mask detector model...")

model.save(args["model"], save\_format="h5")

* plot the training loss and accuracy

N = EPOCHS plt.style.use("ggplot") plt.figure()

plt.plot(np.arange(0, N), H.history["loss"], label="train\_loss") plt.plot(np.arange(0, N), H.history["val\_loss"], label="val\_loss") plt.plot(np.arange(0, N), H.history["accuracy"], label="train\_acc") plt.plot(np.arange(0, N), H.history["val\_accuracy"], label="val\_acc") plt.title("Training Loss and Accuracy")

plt.xlabel("Epoch #")

plt.ylabel("Loss/Accuracy") plt.legend(loc="lower left") plt.savefig(args["plot"])

* import the necessary packages

from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input from tensorflow.keras.preprocessing.image import img\_to\_array from tensorflow.keras.models import load\_model

import numpy as np

import argparse

import cv2

* construct the argument parser and parse the arguments ap = argparse.ArgumentParser() ap.add\_argument("-i", "--image", required=True, help="path to input image") ap.add\_argument("-f", "--face", type=str, default="face\_detector",

help="path to face detector model directory")

44

ap.add\_argument("-m", "--model", type=str,

default="mask\_detector.model",

help="path to trained face mask detector model")

ap.add\_argument("-c", "--confidence", type=float, default=0.5,

help="minimum probability to filter weak detections")

args = vars(ap.parse\_args())

* load our serialized face detector model from disk print("[INFO] loading face detector model...")

prototxtPath = os.path.sep.join([args["face"], "deploy.prototxt"]) weightsPath = os.path.sep.join([args["face"], "res10\_300x300\_ssd\_iter\_140000.caffemodel"])

net = cv2.dnn.readNet(prototxtPath, weightsPath)

* load the face mask detector model from disk

print("[INFO] loading face mask detector model...")

model = load\_model(args["model"])

* load the input image from disk, clone it, and grab the image spatial
* dimensions

image = cv2.imread(args["image"])

orig = image.copy()

(h, w) = image.shape[:2]

# construct a blob from the image

blob = cv2.dnn.blobFromImage(image, 1.0, (300, 300),

(104.0, 177.0, 123.0))

* pass the blob through the network and obtain the face detections print("[INFO] computing face detections...") net.setInput(blob)

detections = net.forward()

* loop over the detections

for i in range(0, detections.shape[2]):

* extract the confidence (i.e., probability) associated with
* the detection

confidence = detections[0, 0, i, 2]

* filter out weak detections by ensuring the confidence is
* greater than the minimum confidence

if confidence > args["confidence"]:

* compute the (x, y)-coordinates of the bounding box for
* the object

box = detections[0, 0, i, 3:7] \* np.array([w, h, w, h])

(startX, startY, endX, endY) = box.astype("int")

# ensure the bounding boxes fall within the dimensions of

45

# the frame

(startX, startY) = (max(0, startX), max(0, startY))

(endX, endY) = (min(w - 1, endX), min(h - 1, endY))

* extract the face ROI, convert it from BGR to RGB channel
* ordering, resize it to 224x224, and preprocess it

face = image[startY:endY, startX:endX]

face = cv2.cvtColor(face, cv2.COLOR\_BGR2RGB)

face = cv2.resize(face, (224, 224))

face = img\_to\_array(face)

face = preprocess\_input(face)

face = np.expand\_dims(face, axis=0)

* pass the face through the model to determine if the face
* has a mask or not

(mask, withoutMask) = model.predict(face)[0]

* + determine the class label and color we'll use to draw
* the bounding box and text

label = "Mask" if mask > withoutMask else "No Mask"

color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

# include the probability in the label

label = "{}: {:.2f}%".format(label, max(mask, withoutMask) \* 100)

* display the label and bounding box rectangle on the output
* frame

cv2.putText(image, label, (startX, startY - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, color, 2)

cv2.rectangle(image, (startX, startY), (endX, endY), color, 2)

# show the output image

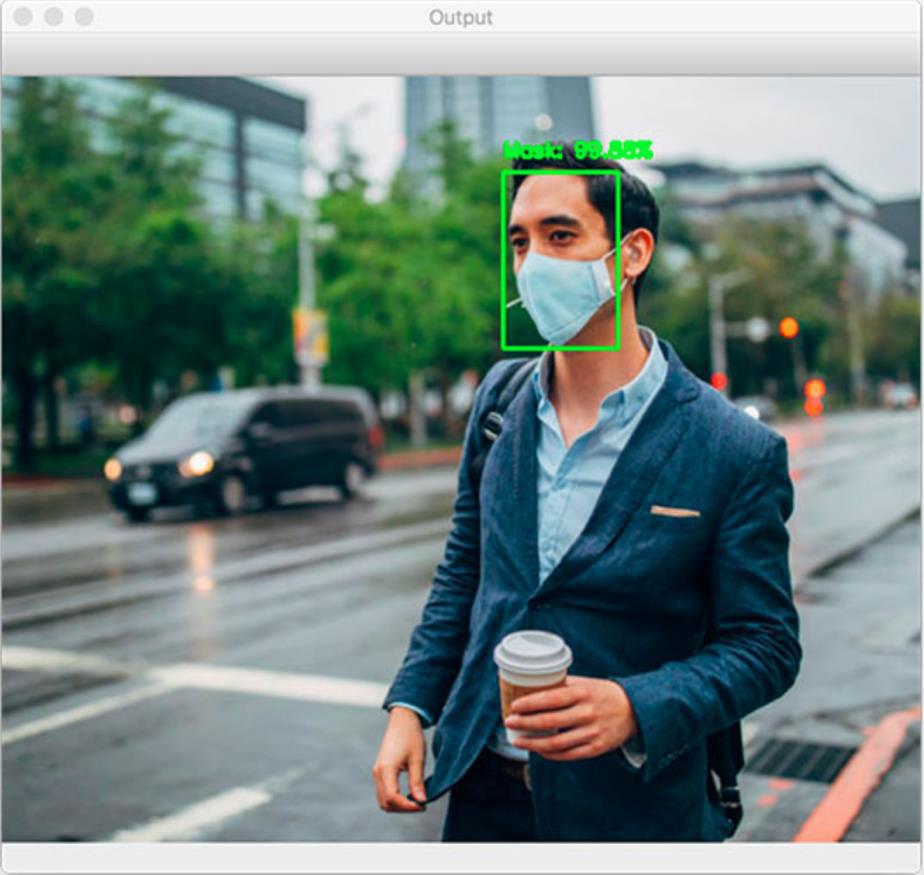
cv2.imshow("Output", image)

cv2.waitKey(0)

46

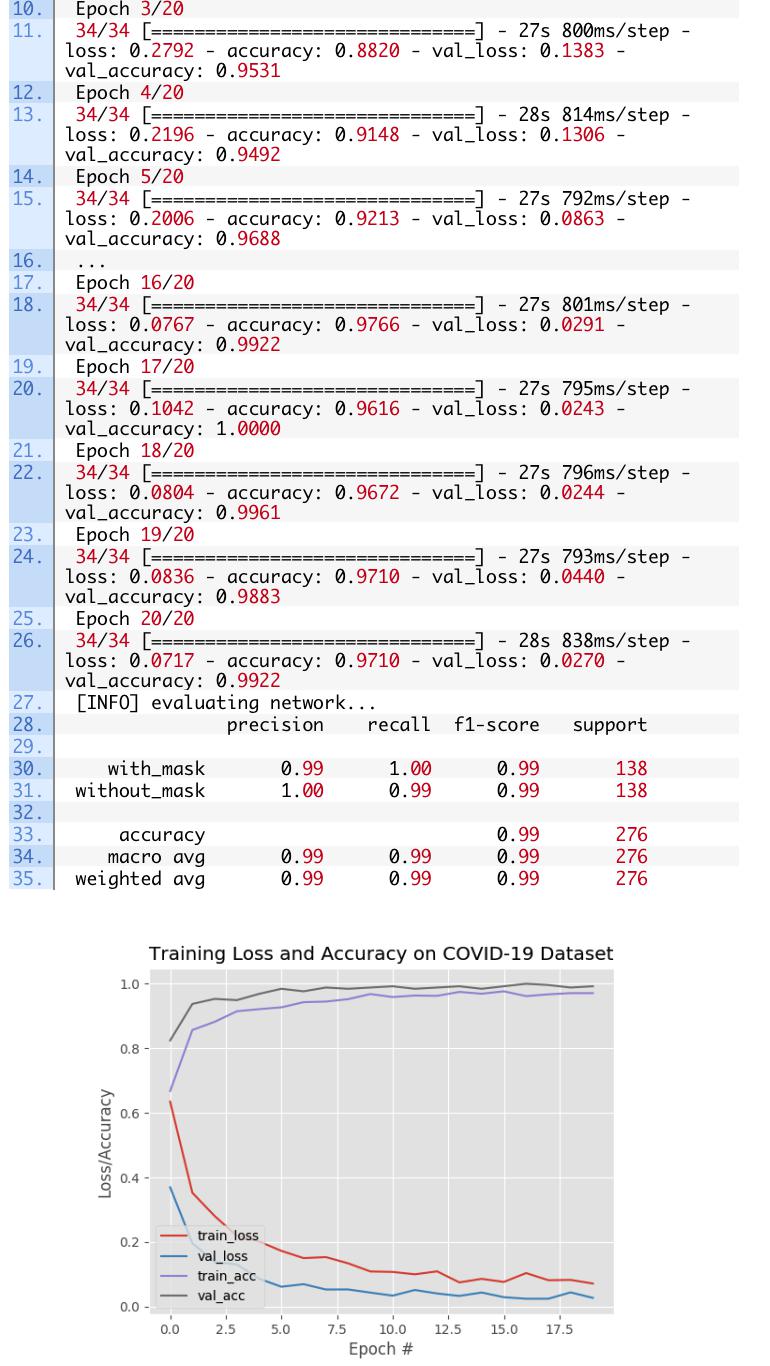
**APPENDIX B**

Below are the project artifacts which compromise of some pictures and screenshots of the hardware and software for this project.



*Figure 5 (A person wearing a face mask has been detected by the Computer Vision algorithm)*

47



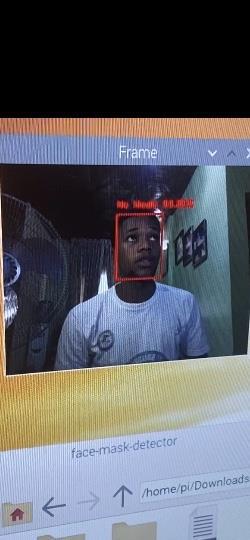
*Figure 6*

*[This image shows the program run during the start of the algorithm to show to show the testing of the model]*

48



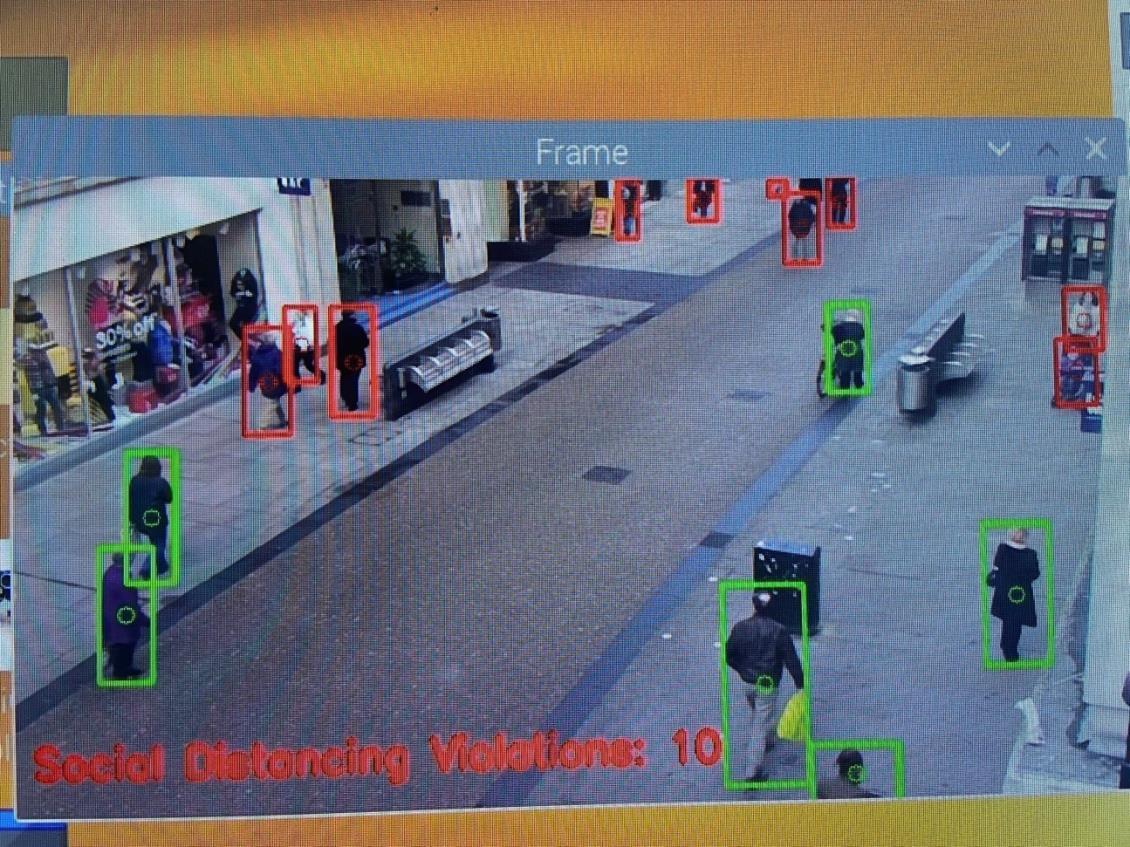
*Figure 7 (Face mask detected)*



*Figure 8 (Face mask not detected)*

Figures 7 & 8 shows two instances, one where I am wearing a face mask and the other, I do not have a face mask. The algorithm works as it encloses my face with a green box in figure 7 and a red box in figure 8. Both showing the level of accuracy of the visual output.

49



*Figure 9 (Social distancing application)*

This is the social distancing application. This computer vision algorithm is calculating the distance between several people in the video and detects and records all those who are not observing the protocol.

50



*Figure 10 (Hardware Prototype Display)*

The picture above displays the hardware device used which is an integration of the Nvidia microcomputer, a motherboard, several camera, heat and other sensors, circuit boards, casing, etc.

Rounding up with an unforgettable quote I took away from Henshaw’s This is Our chance from a character name Bambulu: “This is the child of my brain, the product of my endeavor, and the materialization of my inventive genius.”

51