MEDBOX-AI: AN AI-POWERED MEDICINE SCANNER USING DEEP LEARNING-BASED TRANSLATION OF DOCTOR'S PRESCRIPTION

By:

Amican, Jevanelle

Cabie, Michael Mar

Capillo, Antonio Miguelle

Dumlao, Gilbert

Junio, Alvin Paulo

Ramiro, Dustin Kyle

BS in Computer Engineering

A Final Documentation Submitted to the Department of Computer Engineering, in

Partial Fulfillment of the Requirements of **Project Design** Course

March 2020

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

1338 Arlegui St., Quiapo, Manila

DEPARTMENT OF COMPUTER ENGINEERING

APPROVAL SHEET

The proposed system/project/design entitled **MedBox-AI: An AI-Powered Medicine Scanner using Deep Learning-Based Translation of Doctor's Prescription** which was presented in January 2021, by the proponents:

Amican, Jevanelle

Cabie, Michael Mar

Capillo, Antonio Miguelle

Dumlao, Gilbert

Junio, Alvin Paulo

Ramiro, Dustin Kyle

is now APPROVED by the following members of the committee:

_____	_____
Engr. Mon Arjay F. Malbog	**Engr. Julie Anne Susa**
Panel Member	Panel Member
_____	_____
Dr. Rufo I. Marasigan Jr.	**Dr. Ragnar Alon**
Head Panel Member	Project Adviser

_____

Engr. Jennifer Enriquez

Chairperson, Computer Engineer

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

1338 Arlegui St., Quiapo, Manila

DEPARTMENT OF COMPUTER ENGINEERING

APPROVAL SHEET

The proposed system/project/design entitled **MedBox-AI: An AI-Powered Medicine Scanner using Deep Learning-Based Translation of Doctor's Prescription** has been prepared and submitted by the proponents.

Amican, Jevanelle

Cabie, Michael Mar

Capillo, Antonio Miguelle

Dumlao, Gilbert

Junio, Alvin Paulo

Ramiro, Dustin Kyle

For approval to the committee for the Project Design

After a thorough review and evaluation of the proposed system/design/project, the committee has accepted the proposed design based on the required criteria.

The acceptance is valid to the information being presented January 2021 / 1st semester of the school year 2020 – 2021

Engr. Mon Arjay F. Malbog	**Engr. Julie Anne Susa**
Panel Member	Panel Member
Dr. Rufo I. Marasigan Jr.	**Dr. Ragnar Alon**
Head Panel Member	Project Adviser

Engr. Jennifer Enriquez

Chairperson, Computer Engineer

ACKNOWLEDGEMENT

This project is made possible through the help and support from everyone, including parents, professors, family, and friends, and in essence, all sentient beings. Also, the proponents, dedicate this acknowledgment of gratitude towards the following significant advisors and contributors:

First and foremost, we, The Frontlinerz, gladly appreciate the goodness of our Lord, God for giving us the knowledge we need and a great motivation to strive for the best throughout this project. Indeed, without the presence and guidance of the Lord we would not be able to finish this system on time, so Almighty God we will be always in your debt.

Also, our class adviser, Engr. Jennifer Enriquez and project adviser Dr. Alvin S. Alon for imparting their knowledge to us and for giving us the best advice they could share just to make our project more reliable and effective. Without their conduction and guidance, we will never be able to successfully do our project and present it to the panel.

To our beloved friends, colleagues, and classmates who had a huge help in persuading and giving us great advice and motivation to do much better on this project. For letting us know how time management and patience is really important for us to finish this on due time.

And lastly, we would like to greatly appreciate our family for giving their undying love, guidance, and support during the uneasiness of project making. We won't be this dedicated and determined without them as our inspiration. So, with due regards, we express our gratitude and love to them.

Thank you!

Amican, Jevanelle

Cabie, Michael Mar

Capillo, Antonio Miguelle

Dumlao, Gilbert

Junio, Alvin Paulo

Ramiro, Dustin Kyle

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The project, **MedBox-AI: An AI-Powered Medicine Scanner using Deep Learning-Based Translation of Doctor's Prescription** is a translation device that uses object detection to translate the sloppily handwritten of the doctor's prescription. Medbox AI was specifically designed to help pharmacists in translating the medicine names that are present on the prescription

Keywords: Doctors prescription, Object Detection, Deep learning, Pharmacist, AI-Powered, Python, Google Colab, Medicine.

CHAPTER I: PROJECT BACKGROUND

This chapter includes the project background. The ideas gathered significantly influence the project's development, the target clients, the prospective scope and limitations, and totality development.

The Project

Pharmacy is accepted as a part of medical practice back in Sumerian times. Cuneiform tablets preserved recording prescription medicines and a variety of ancient Egyptian papyri on topic theoretical aspects. In 1752 AD, the first pharmacy attached to a hospital in Philadelphia began operations. As a pharmacist at the hospital from 1755 to 1756, John Morgan was an early promoter of prescription writing.[1]

A prescription is a legal document; (handwritten or electronic) issued by a licensed physician to the pharmacist to prepare or deliver pharmacological agents/medications. to diagnose, prevent, or treat a disease. There are primarily two types of prescriptions. First is the Pre-compounding prescriptions, which is the prescription for the prepared drug by a pharmaceutical company, e.g., Cap. Ciprofloxacin (500 mg) etc. This form of medicine is more common these days and includes prescriptions distributed in pre-compound form and ready to be given as it is. The pharmacist or formulations delivering the medication are not changed. The other types of prescriptions were the extemporaneous prescription, the kind of prescription in which the pharmacist prepares the medicine according to the physician's drugs and dosages. Mixing of the ingredients of a prescription or drug formula and generally refers to a manual process performed for individual orders by a dispenser or pharmacist[2]

There are two ways to do a prescription, which are the written prescription and e-prescription. A written prescription is when the doctor writes all the details in a prescription form and on his/her record. E-prescription is also a way to do a prescription. E-prescription or electronic prescription is the practice of submitting a prescription directly to a pharmacy via apps on a computer or other electronic devices, such as a tablet, smartphone, or printer.

A written prescription is a common way to prescribe medicine in the Philippines. But more than half (55%) of the adult Filipino patients have difficulty understanding their doctor's prescription, according to a study at the University of the Philippines Manila (UPM). Research proves that this widespread allegation is accurate since that doctors' handwriting is not worse nor better than that of other professionals. The unreadability of a doctor's handwriting in prescribing medication causes medication errors such as administration of improper dosage of medicine or even death. It explained that most doctors' poor handwriting is attributed to when doctors are in a rush when writing prescriptions, during their rounds or peak hours, or experienced fatigue.[3]. That causes doctors to rely on stenography or shorthand lettering to write important information on the prescription as fast as possible.[4] Stenography or shorthand is a handwriting system that uses symbols and abbreviations for letters, words, or phrases to write the information faster. [5]

Validating if the problem identified is worth solving, a survey was conducted. Below are the survey results with the pharmacist as respondents.

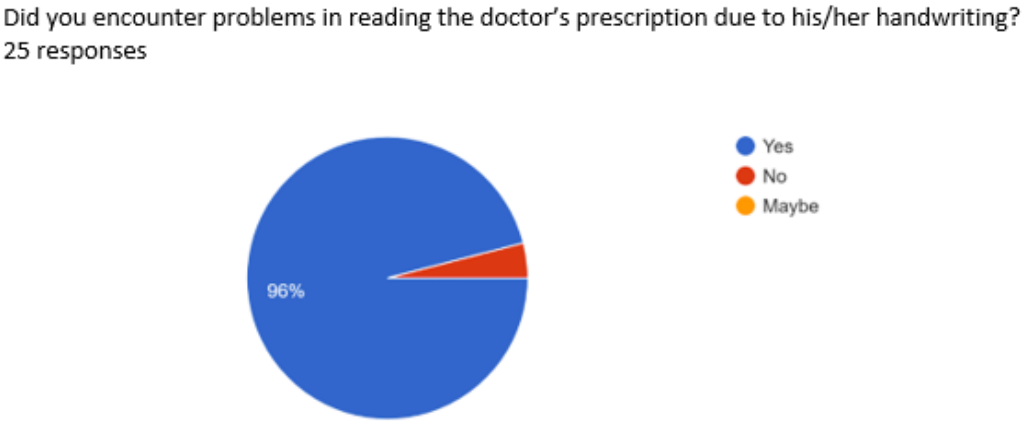


Figure 1.1 Survey responses to question 1

Figure 1.1 is the result of the survey conducted last February 2020 regarding the encountered problems in reading the doctor's prescriptions due to handwriting.

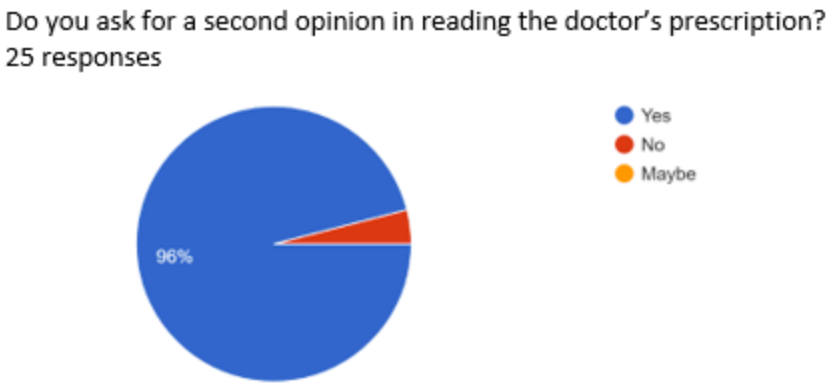


Figure 1.2 Survey responses to question 2

Figure 1.2 is the survey result that talks about if pharmacists asked for help or others' second opinion when reading the doctor's prescription.

Based on the proponents' survey, pharmacists need to learn and understand the sloppy handwriting present in the prescription. It was accepted that doctors have illegible handwriting. The writer usually knows what was written, but when other parties are involved, they often have problems reading and interpreting the text.[6] The only solution is to verify it with their supervisor. If they have the same reading or understanding of the prescription, then they would proceed; otherwise, they would contact the doctor directly to confirm the prescription. They would ensure and ask the doctor who gave the medication if they read the prescription correctly on the pharmacists. And based on the conducted interview, other doctors would type out the prescriptions with their assistant's help. Also, worth noting that not all doctors have this kind of privilege.

Based on the proponents' interview, there are instances of difficulty in translating the prescription especially if the written prescriptions are written sloppily. It consumes much time to verify it with other pharmaceutical personnel. When this method is used, it would usually take around 5 minutes or more to confirm the correct reading of the sloppily written prescription and they usually use based on their translation from experience. There is no assurance that they can contact the doctor who gave the prescription, which causes the patient to wait and, in the end, they look for another drug store.

Through the survey, the following design problems were formulated:

1. There is no existing platform that helps the pharmacy to translate the sloppily written prescriptions.
2. The turn around time of verifying the context of the prescriptions is high.
3. High probability of incorrect interpretation of the doctor's prescription by the pharmacist.

Project Objectives

The project aims to develop a platform that helps pharmacists in performing their job effectively. Specifically, to:

1. Provide a system that helps the pharmacy to translate the sloppily written prescriptions;
2. Decrease the turnaround time in verifying the context of the prescriptions; and
3. Decrease the probability of incorrect interpretation of the doctor’s prescription by the pharmacist.

The Client

The potential clients for this project are the pharmaceutical drug store currently not attached to any hospital. It means that any pharmaceutical drug stores outside of the hospital can be potential clients of this project. When the pharmacists in a hospital were unable to read the prescription correctly, the usual solution is to consult with the doctor who made the prescription to verify the prescription's contents. While the pharmaceutical drug stores outside of hospitals would have no means of ascertaining the prescription contents if they read the correct medicine present on the said prescription correctly. With this project, the pharmacist would easily read any prescription regardless of how sloppy the handwriting is and avoid making prescription errors. It would result in the fact that they would provide the correct medicine to the customer without the trouble of reading the doctor's sloppy handwriting in the prescription. Based on the proponents' interview, the client mentioned that the device should guarantee reliability and when it comes to cost the clients agreed that they would buy the platform for less than 15,000 PHP as long as the platform would prove useful.

Project Scope and Limitations

The project is about designing an AI-Powered Medicine Scanner using Deep Learning-Based Translation of a Doctor's Prescription. To ensure the attainment of the objectives, this project covers the following:

- Analyzing the generic name of the medicine indicated on the prescription, accurately identify the captured prescription using deep learning.
- Determine and interpret the medicine's handwritten generic name on the prescription without consuming much time.
- The GUI of the platform must be user friendly.

However, due to limited time and resources, here are the limitations of the design project

- Is not able to check the legitimacy of the prescription provided.
- Only translate the written generic name of the medicine present on the prescription.
- The platform has limited update capabilities.
- Only translate 5 antibiotics, including Amoxicillin, Doxycycline, Cephalexin, Ciprofloxacin, and Clindamycin.

- This project would only provide the design for the hardware and not the hardware itself.

Project Development

The proponents used the waterfall model, as shown in Figure 1.3. It is essential to make a systematized structure of the venture to avoid conflicts in developing the design.

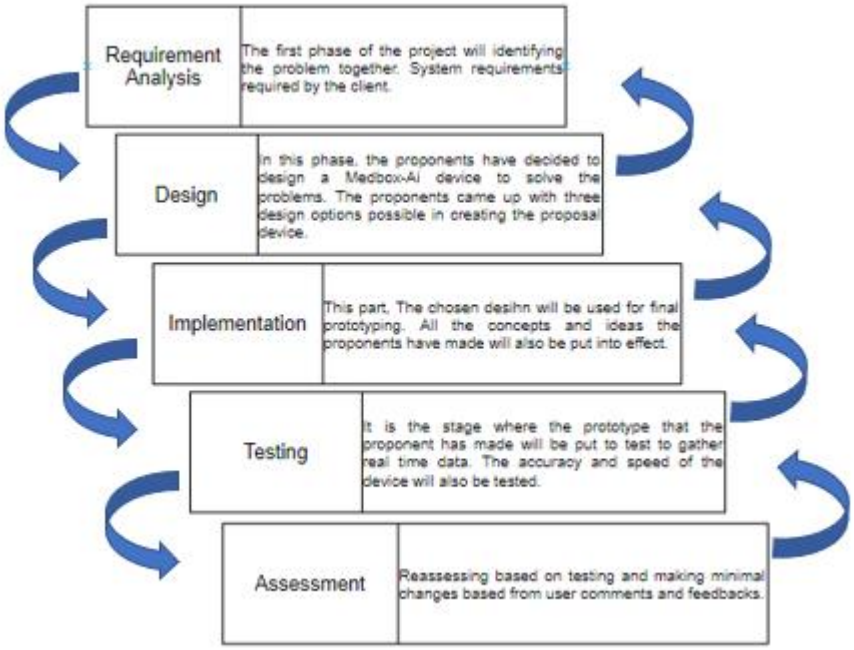


Figure 1.3 Medbox-Ai Waterfall Model

Figure 1.3 shows the Project Development model used in MedBox-Ai. In the first part, the proponents have identified the problem to be solved. The proponents have obtained the parameters to translate the prescriptions by gathering data from studies and research. Proponents were conducting a survey and interview. After getting the parameters, the next phase is the design, wherein the proponents utilized the obtained information to create several design options and design constraints for the project. The proponents conducted a basic design ranking to arrive at the most efficient method in developing the project. Subsequently, the proponent started designing the actual flow and operation of the project. With the construction of the device's initial design, the proponents could now finalize its creation. The proponents carry on to the implementation and testing. Further Analysis is done while testing the device, which could drive the proponents to see some errors and direct design correction and improvement to the said design.

CHAPTER II

This chapter includes the client's requirement that lists all the features of MedBox-AI. It also includes related studies and articles that have significant relation and influence on the project's development. Details about the devices and technologies involved in the projects are also stated.

Client Requirement

After verifying the problem and setting the design project's objectives, below are the client's requirements and implemented in the AI-Powered Medicine Scanner using Deep Learning. The device should/can:

- Determine the medicine on the prescription with the correct translation.
- Display the accurate prescribed medicine.
- The processing time must not cause inconvenience to the pharmacy.
- Affordable Price.

Design Criteria And Design Constraints

The design contains the explicit goals the project must achieve to be successful—the criteria based on the client's requirement for the project to be done. The design constraints are the limitations set according to the design criteria defined by the client.

Table 2.1 Design Criteria And Constraints

DESIGN CRITERIA	DESIGN CONSTRAINTS
Accuracy	The accuracy must not be lower than 70%.
Speed	Based on the interview the speed for the translation should not be higher than 4 seconds.
Economical.	The total cost excluding miscellaneous must be below 15,000 PHP.

Table 2.1 shows the client's requirements, design criteria were derived, and the corresponding constraints presented in the table.

Accuracy

This project is concerned with the output accuracy of the overall system. The design to be implemented must be with high precision and an acceptable rate of failure. Inadequate equipment, low data processing, or human error can lead to inaccurate results that are not very close to the truth.[7] Accuracy can help increase users' satisfaction.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

(equation 1)

Or for Binary classification, it can be calculated in terms of positive and negatives:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

(equation 2)

Where:

- True Positive (TP): A true positive test result detects the condition when the condition is present.
- True Negative (TN): A true negative test result does not detect the condition when the condition is absent.
- False-positive (FP): A false-positive test result is one that detects the condition when the condition is absent.
- False-negative (FN): A false-negative test result is one that does not detect the condition when the condition is present.

Speed

The proponents must create a device that translates the prescription in a short period. It avoids the hassle for the users and helps them utilize their time in using the platform. Significant delays may cause errors and unnecessary waiting time for the users. The proponents should measure the speed based on the algorithm used. Factors such as algorithm efficiency and simulation environment can affect the rate of the system. Do note that the simulation environment that the proponents used is google colab that offers 2 different GPU models since 2019, which are the K80 and T4. These 2 GPUs offer 12GB - 16GB of GPU memory. However, the user cannot select which of the 2 GPUs would be given to the user due to the availability issues. Each of the users has a limited quota for T4 GPUs. This would mean that taking advantage of the more powerful GPU in Google Colab is not always guaranteed.

$$Speed = speed\ of\ algorithm + Google\ Colab\ free\ GPU$$

(equation 3)

Economical

This design constraint requires achieving the product by improving the business environment. The design must cover a cost that can guarantee the client's requirements. It must also finance and facilitate the maintenance of the product for the long term. For this project, the materials must be cost-efficient as possible, with the design's overall performance justifying the cost. The total cost can define as:

$$Cost = \sum cost\ of\ components\ (PHP)$$

(equation 4)

To compute the system's cost, the proponents must create a Bill of Materials (BOM), a comprehensive inventory of a material specification: In addition to direct costs. [8]

Storyboard

Storyboard created to visualize the solution to the problem. Based on the information gathered, effective solutions can be designed to attain the client's requirement mentioned.

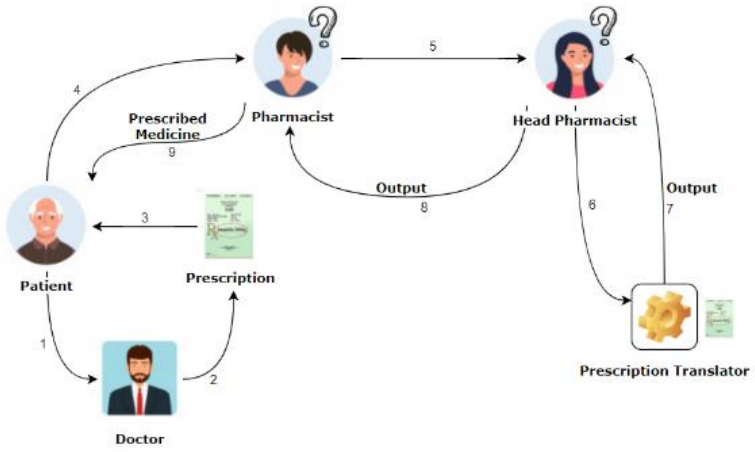


Figure 2.1 Storyboard Medbox-Ai

This scenario would start with the patient going to the doctor to ask for a prescription. Which, in turn, the doctor would give the patient the prescription that they would need. After receiving the doctor's prescription, the patient would then go to a pharmacy to buy the doctor's medicines. For the patient to buy the prescribed medicine, they would need to present the doctor's prescription to the pharmacist. Then the pharmacists would try to read and decipher the prescription provided by the patient. If the pharmacist weren't able to understand the prescription, the pharmacists would go to the head pharmacist to ask for an opinion regarding reading the prescription. If the pharmacist understood the prescription, then the pharmacists would give the correct medicine from the prescription. If not, then this is when our solution, a prescription translation device, would come in, and the pharmacist would just put the prescription in the device. The machine would translate the medicine written in the prescription and display it on the screen. After seeing the device's translation, the pharmacist would now give the correct medication in the prescription.

Relevant Information

There are many studies regarding Illegible handwriting and other prescription errors. It was accepted that doctors have illegible handwriting. The writer usually knows what was written, but when other parties are involved, they often have problems reading and interpreting the text [6]

An article found in the British Medical Journal investigated that a 43 years old American died after the pharmacist dispensed Isordil medicine, which he thought was Plendil medicine, due to the doctors' illegible handwriting on the doctors' prescription. [9]January 2007 report from Time Magazine that more than 7,000 people annually kills due to doctors' sloppy handwriting. Therefore a petition has been made for doctors to use digital notes and prescriptions to prevent these errors. [10]

However, a studies report shows that electronic prescriptions do not prevent common prescription errors in manual handwritten prescriptions. [11]Regardless of the computer revolution, most information in clinical records is still written by hand.

The lack of a translation platform leads to a platform that can use as a prescription translator. In the pharmaceutical field, people rely on the present solution, asking their coworkers, or contacting the doctor who prescribed the prescription. To create such a platform that helps to understand the illegible handwritten prescription. Technical requirements must be present and defined.

Python is a recognized high-level programming language that can use in a wide variety of applications. It has high-level data structures, dynamic typing, dynamic binding, and many more complex application development features. It can also use for scripting that would connect the different components. One of its features is that it can extend to run on almost all available operating systems today. It also supports thousands of 3rd party modules that can support different kinds of developments. [12]

Object Detection is needed for designing the solution. It is a computer vision technique for detecting and locating objects in images or videos. Object detection algorithms usually use machine learning or deep learning to produce meaningful results. It is understood that when humans see an object, they can automatically recognize and locate the object they are looking at in a matter of moments. The mimicking of this instance of human intelligence using a computer is the end goal of object detection technology. One example of its implementation is the Advance Drivers Assistance System (ADAS), which would enable cars to detect the objects in front of them, which would result in improved traffic safety. [13]

Many variables can significantly affect the image quality of the captured images, potentially affecting the algorithm's accuracy. Sharpness is the sum of the image that can be transmitted. When the picture's subject is sharp, the image appears clear and vibrant, with detail, contrast, and texture made in great detail. And when the picture lacks sharpness, it would look blurred and lacking in detail. Next is camera noise, a random variation of the brightness or color information in the images captured. It is a loss of image signals caused by external sources. Images containing multiplicative noise have the characteristic that the brighter the field, the louder it is. But it's mostly additive, in any case. The next one is distortion, which is observed when a person takes a picture with many straight lines, but it looks as if the straight lines were curved when taken by a camera. There are two types of distortion, optical distortion and perspective distortion, both of which result in image deformation.[14]

Optical distortion leans more on the lens itself, while the camera's location distorts the viewpoint. Next is uniformity. One example of a uniformity situation is that the image taken should have the same amount of brightness. If there are any brightness variations or other variables on the image, the result is non-uniformity. And finally, data compression can have a significant impact on the quality of the image. For example, if the image is saved as a low-quality JPEG, it would be lost, and there would be much information in the image that would be lost.[14]

TensorFlow is a machine learning method that works. on a broad scale and in a heterogeneous world. It's an open-source, artificial intelligence platform created by Google.TensorFlow uses data flow graphs to represent computation, shared state, and the operations that mutate that state. It maps the nodes of a data flow Graph across multiple computers in a cluster and across multiple computing devices, like multi-core CPUs, general-purpose GPU's and custom-designed ASIC's Named Tensor Processing Units (TPUs). This architecture gives the application developer versatility. TensorFlow enables proponents to Experiment with new optimizations and training algorithms. TensorFlow supports several applications, with an emphasis on training and inference in deep neural networks. Several Google services are using TensorFlow in development. [15]

Convolutional Neural Network is a Machine learning technique that uses deep, feed-forward, artificial neural networks. It's been commonly used in the study of visual images. They are also classified as shift invariant or space invariant artificial neural networks (SIANN) based on their shared-weight design and translation invariance characteristics.

CSPDarknet53 is a convolutional neural network and backbone for object detection that uses Darknet53. It employs a CSPNet strategy to partition the base layer's feature map into two parts and then merges them through a cross-stage hierarchy. The use of a split and merge strategy allows for more gradient flow through the network. Also worth noting that this is the backbone used by YOLOV4.[16]

Inception v2 is a deep neural network with an architectural architecture consisting of replicating components referred to as inception modules. It typically consists of several convolutional layers: a 1x1 convolutional layer, a 3x3 convolutional layer, and a 5x5 convolutional layer. [17] And Inception v2 is the second generation of Inception convolutional neural network architectures that use batch normalization.

Mobilenet v2 is a family of high-performance computer vision neural networks. It was developed for mobile devices designed to support classification, identification, and more. It can operate deep neural networks on mobile devices. It also provides benefits, such as protection, privacy, and power consumption. Mobilenet v2 significantly improves MobileNetV1 and drives the state of art for mobile visual recognition, including classification, object detection, and semantic segmentation.[18]

SSD is a single-shot detector. It has no delegated region proposal network and predicts the boundary boxes and the classes directly from feature maps in one single pass. SSD introduces: small convolutional filters to predict object classes and offsets to default boundary boxes to improve accuracy.[19]

YOLO stands for You Only Look Once, an object detection algorithm that can rapidly detect multiple objects within a single image. This detection can also be done in real-time without compromising its accuracy and precision. YOLOv4 is one of the latest YOLO iterations that significantly improve the existing performance of previous versions of YOLO. In this case, this version has dramatically improved detection performance and superior speed.[20]

Faster RCNN is the 3rd iteration of the RCNN. R stands for Region, and CNN stands for Convolutional Neural Networks. One of its goals is to reduce computational cost and to be able to work in real-time. It has 2 regions; the first one is a convolutional network that proposes the Region. In contrast, the second one is the one that uses the proposed Region. It uses the Region Proposal Network, which would tell where the Faster RCNN would look [21]

The microprocessor is the fundamental unit of a computer system that performs arithmetic and logical operations., including adding, subtracting, transferring numbers from one area to another, and comparing two numbers. It's often known merely as a processor, a central processing unit, or a logic chip. It's essentially the engine or the computer's brain that goes into motion when It's switched on now. It is a programmable, multipurpose computer that integrates the functions of a CPU (central processing unit) on a single IC (integrated circuit)[22]

Colaboratory or “Colab” in short, it's a Google Research product. Colab allows anyone to write and execute arbitrary python code via a browser and is specifically suited to machine learning, data analysis, and education. More technically, Colab is a hosted Jupyter notebook service that needs no setup while providing free access to computing resources, including GPUs.

CHAPTER III: PROJECT DESIGN

This chapter thoroughly discusses the designs for the proposed device, the MedBox-AI. Each proposed design is presented through the graphical illustration of the system architecture that shows the study's system flow and, finally, the design options. The process and components used in every design option would also present in this chapter.

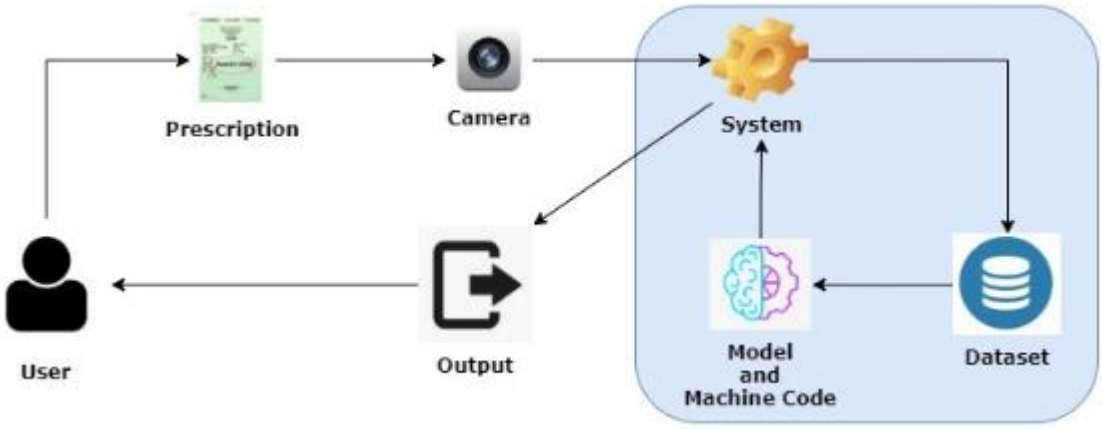


Figure 3.1 Medbox-Ai System Architecture

Figure 3.1 indicates that the MedBox-AI device would start with a doctor's order, and then a prescription would be put in front of the camera. After the camera captured the prescription, the machine would interpret the prescription's written portion using the chosen algorithm. The algorithm you want must be accurate in terms of precision, and the pace must be consistent. After that, the device would display the translated output.

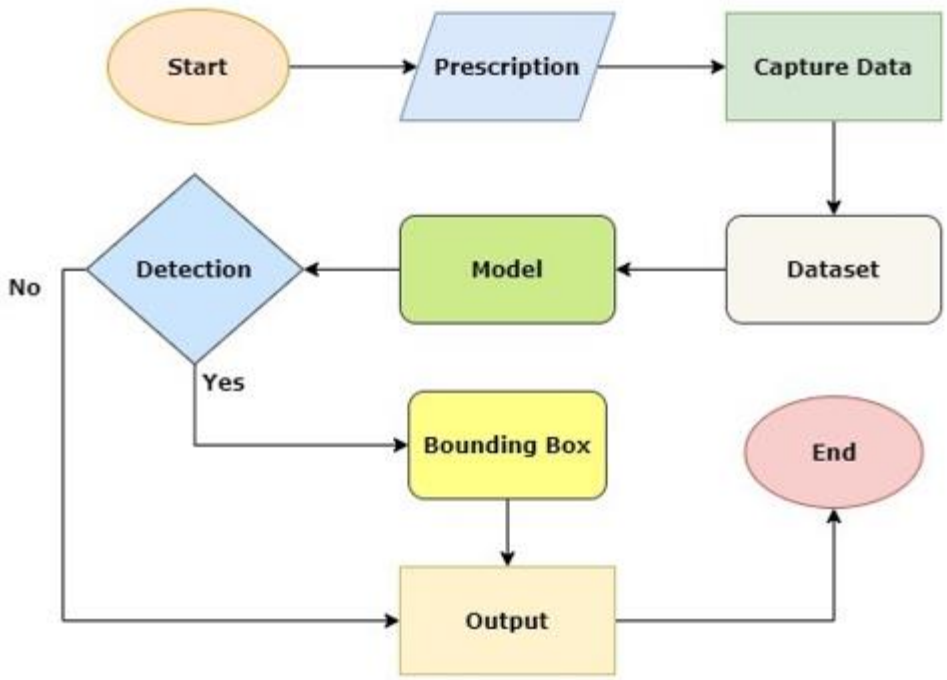


Figure 3.2 Medbox-Ai System Flowchart

Figure 3.2 demonstrates the flow of the MedBox-AI system when the user introduces the prescription into the system. The device captures a prescription image, and the captured image feeds it to the algorithm, then the algorithm would detect the name of the antibiotics prescribed, and then the algorithm would position a bounding box in the name of the antibiotic detected. And finally, the algorithm would display the translated name and bounding box on the LCD. If the algorithm could not detect the antibiotic's name, it would be displayed without any translation.

This section talks about the process of producing a dataset and the process of training the model. Do note that the images captured for the dataset were captured using Huawei Mate 30 Pro, a 40-megapixel camera.

A. Process of Producing the Dataset

Data Gathering

Before the training can begin, the proponents first contacted 2 doctor friends of a member and asked them to provide our datasets. In this case, 5 antibiotic names were handwritten by 2 doctors about 200 times, which would get the proponents 1000 handwritten names

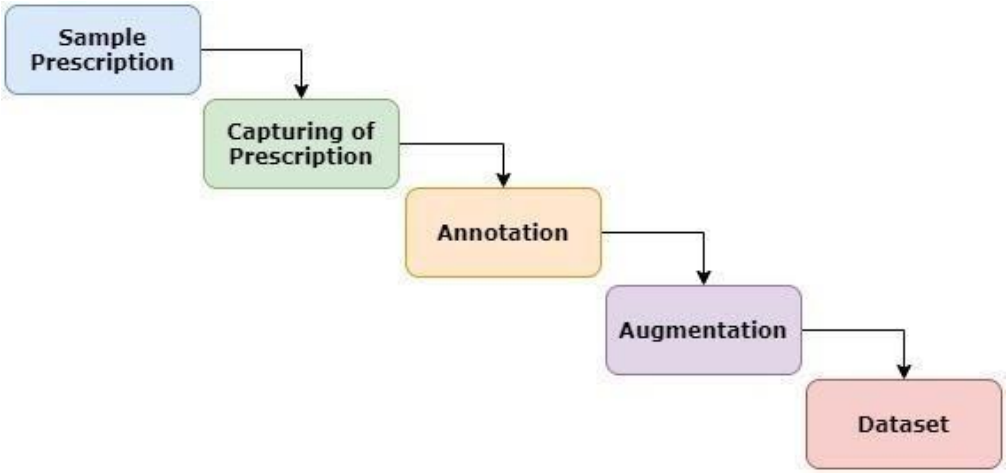


Figure 3.3 Process Flow Of Dataset

In figure 3.3 shows the process flow of creating the dataset needed for the algorithm. It would start with the sample prescription, capturing prescriptions, annotation, augmentation, and finally, the output dataset.

Figure 3.4 Amoxicillin Sample

Figure 3.5 Doxycycline Sample

Figure 3.6 Ciprofloxacin Sample

Figure 3.7 Clindamycin Sample

Figure 3.8 CEPHALEXIN SAMPLE

1. **Sample Prescription** - The process of creating a dataset would start with the gathering of data. The developers ask for help through one of the member's doctor friends in providing the sample prescription.

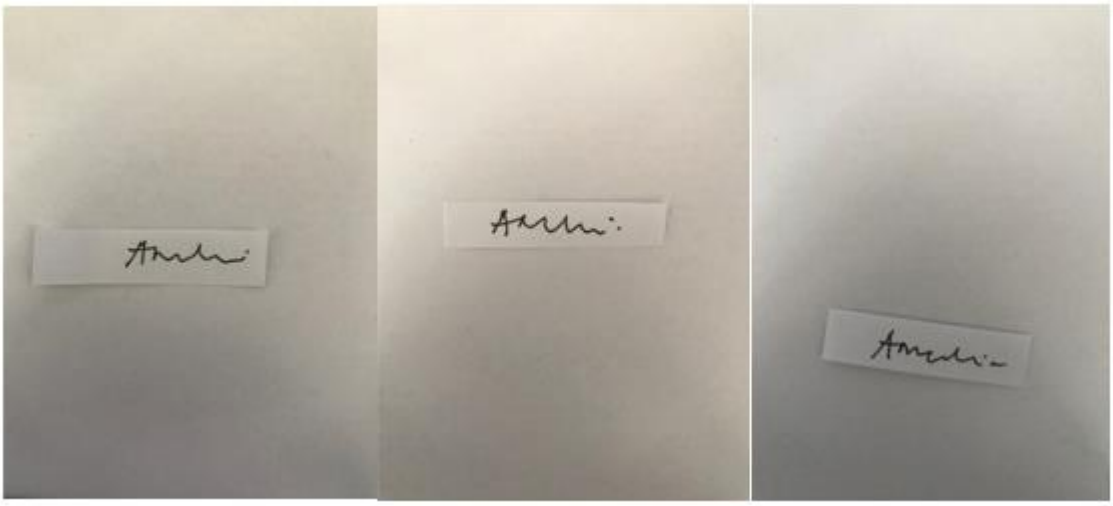


Figure 3.9 Amoxicillin

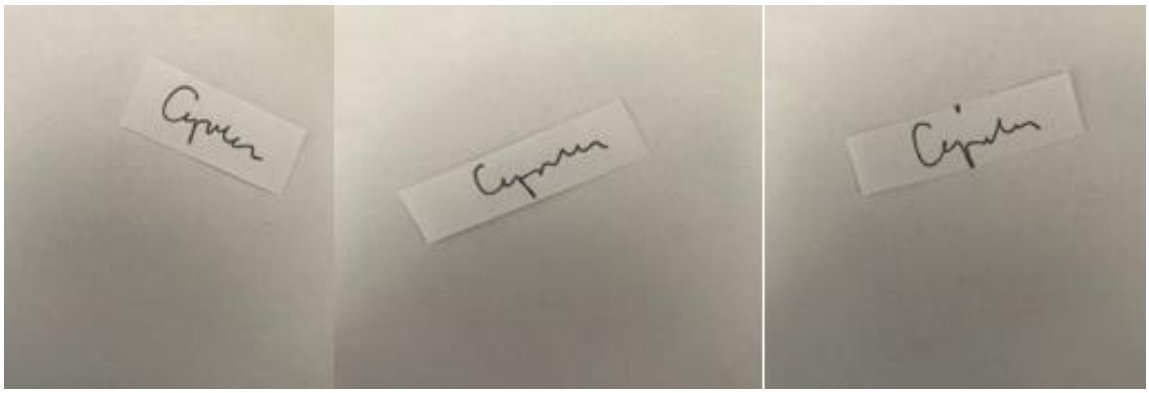


Figure 3.10 Cephalexin

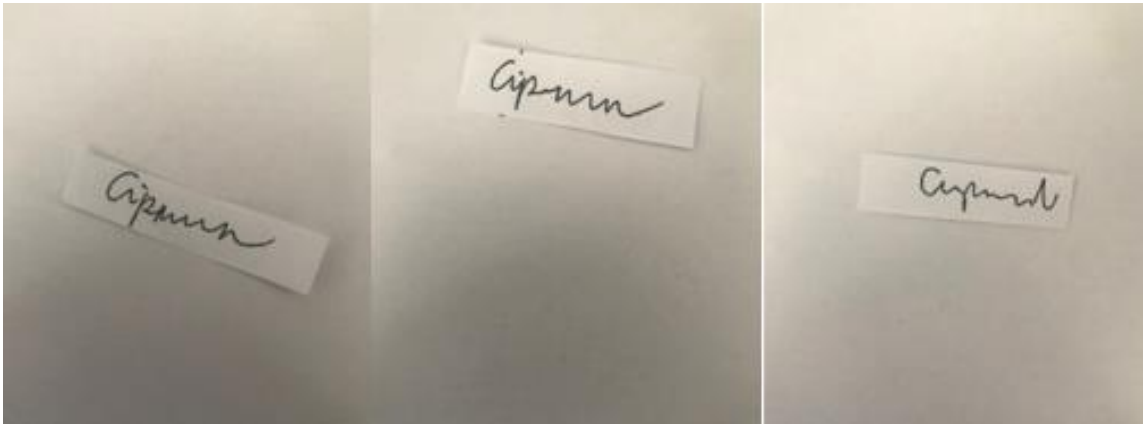


Figure 3.11 Ciprofloxacin

Figure 3.12 Clindamycin

Figure 3.13 Doxycycline

2. **Capturing of Prescriptions** - After enough data gathered, the developers cut the antibiotic's names for the dataset. And after cutting the sample prescriptions, the developers captured images of the sample prescriptions.

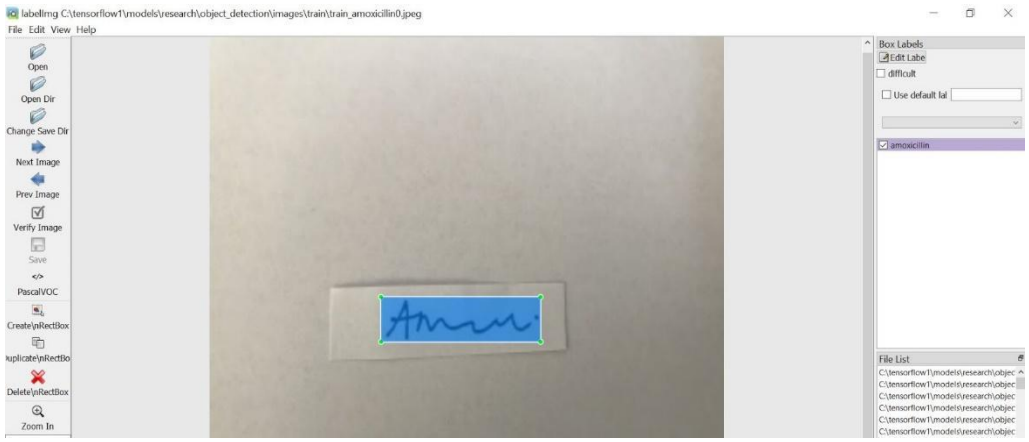


Figure 3.14 Amoxicillin Annotation

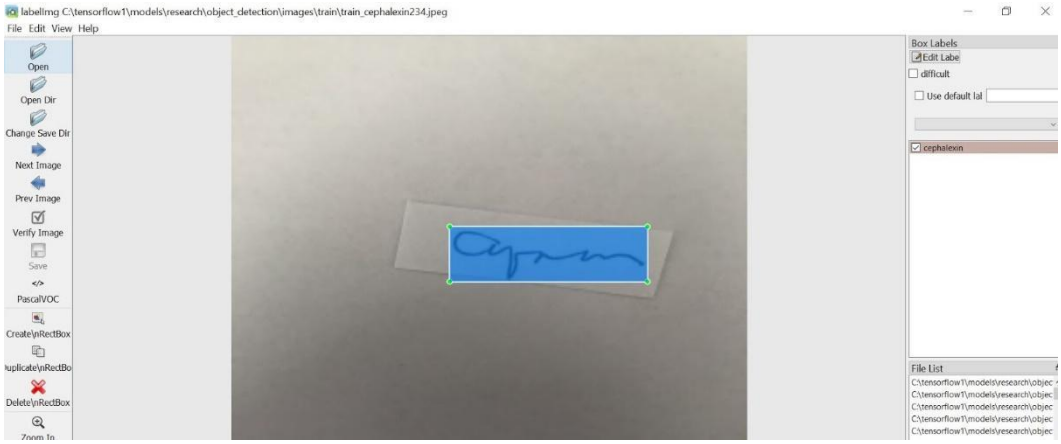


Figure 3.15 Cephalexin Annotation

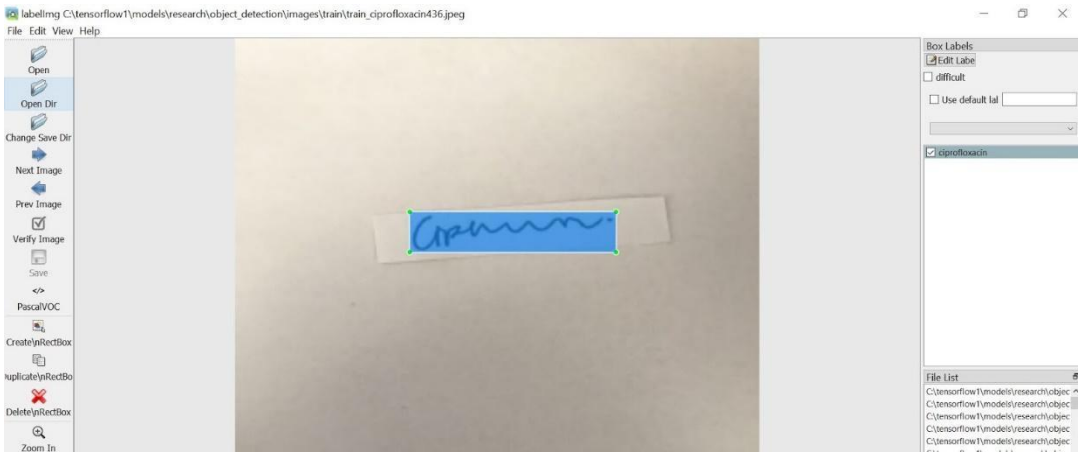


Figure 3.16 Ciprofloxacin Annotation

Figure 3.17 Doxycycline Annotation

Figure 3.18 Clindamycin Annotation

3. **Annotation** - Then, the capture images would be annotated or labeled in regards to the data needed. The role of labeling data with metadata is annotation. Typically, this metadata takes the form of tags. That can add to different data in the dataset. The process would first select and identify the data to be tagged and then add it to the image's selected portion.[23]

Figure 3.19 Augmented Sample

4. **Augmentation** - After that, the final step would be the augmentation of the data, which would create a modified version of the Dataset images. Doing this would help the system be more accurate despite the different states that the images are in.[24]. The data augmentation done in the existing dataset is that the developers changed the position, brightness, and contrast of the existing dataset. This process also helped improve the algorithm by providing additional datasets for training.

5. **Dataset** - is the final output in the process. It is the one that would be used to train the algorithm. In this case, the proponents were able to generate 1,200 images for the algorithm's data set.

B. Process of Training the Model

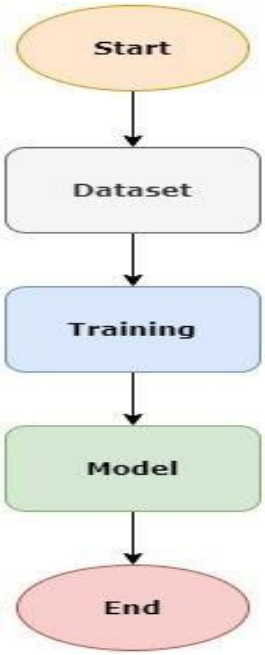


Figure 3.20 Training Flowchart

The training of the dataset would start by gathering lots of images for the dataset. The next step would be training, which could take much time regarding the number of images used. After training the dataset, the output would be the model that could be used for the respective algorithms.

Selection of Processor

In Deep Learning and Machine Learning, it has often been connected to massive computers with fast CPUs and GPUs, large RAM sizes, or Cloud-based algorithms. Microcontrollers processors are widespread. It's all around, like household appliances, toys, cars, etc. The possibilities are limitless when it comes to machine learning brought to the microcontroller. The proponent selected the two best microcontrollers that may suit the platform. These microcontrollers are raspberry pi 4b and Jetson Nano.

Figure 3.21 Raspberry Pi 4b

The Raspberry Pi 4b is the latest version of the low-cost Raspberry Pi computer. It has different RAM sizes, 4 GB and 8 GB. It is like a mini-computer that can run some applications. [25]

Figure 3.22 Nvidia Jetson Nano

Jetson Nano is a small, rugged computer for embedded applications and AI IoT that delivers modern AI's power. Jetson Nano has the efficiency and capabilities that you need to run modern AI workloads. offering you a quick and easy way to add advanced AI to your next product.. [26]

Table 3.1 Specification Compare

Specification	Jetson Nano	Raspberry Pi 4b
Processor	128-core NVIDIA Maxwell™ architecture-based GPU, Quad-core ARM® A57 @ 1.42GHz	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	4 GB 64-bit LPDDR4; 25.6 GB/s	2GB, 4GB or 8GB LPDDR4 (depending on model)
Operating System	Linux for Tegra®, Ubuntu	Raspbian, Ubuntu MATE/Core, Ubuntu Server, Windows 10 IoT Core, OSMC, RetroPie
Connectivity	Gigabit Ethernet, M.2 Key E, 802.11ac Wi-Fi, Bluetooth LE 4.1, Ethernet (max 300 Mbps)	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports
GPIO	40-pin GPIO, I2C, I2S, SPI, UART	Standard 40-pin GPIO header (fully backward-compatible with previous boards)
Video and Sound	4x USB 3.0, USB 2.0 Micro-B, HDMI 2.0 & eDP 1.4, Support of MIPI CSI-2 DPHY lanes, 12x (Module) and 1x (Developer Kit) and PCIe Gen2 high-speed I/O	2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Multimedia	4K @ 30 fps (H.264/H.265) / 4K @ 60 fps (H.264/H.265) encode and decode	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphic
SD Card Support	Micro SD card slot	Micro SD card slot for loading operating system and data storage
Input Power	DC Barrel jack for 5V	5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment	Extended temperature range	Operating temperature 0–50ºC

In this section, the proponents determine the best microcontroller using the pairwise comparison: Each value is assessed using the formula below:

$$value = \sum(i * ranking)$$

(equation 5)

Table 3.2 Trade-Off Constraint Value In Microcontroller Processor

Microcontrollers	Cost (Php)	Speed (GHz)
Raspberry Pi 4b	4, 749.75	1.5
Nvidia Jetson Nano	7, 699.00	1.42

In table 3.2, shows the comparison between the cost and the speed for both microcontroller processors. Both of these options have advantages and disadvantages to each other.

Table 3.3 Trade-Off Of Microcontroller Processor

Criteria	Importance Factor	Raspberry Pi 4b	Jetson Nano
Cost (Php)	0.6	5	3.08
Speed (GHz)	0.4	5	4.73
value= i*(ranking)		5	3.74

In table 3.3, the microcontroller processor's comparison shows the best microcontroller, which is the Raspberry Pi 4b. The winning microcontroller processor is chosen to be the processor of the platform design.

The other components used in the design of the project prototype are:

Figure 3.23 Sd Card

Figure 3.23 is a Class 10 UHS 1 microSD card that is one of the best options if you are looking for fast application performance. The 32 GB microSD card features a maximum transfer speed of up to 98 MBps. This SD card would act as the Rom of both of the microprocessors.

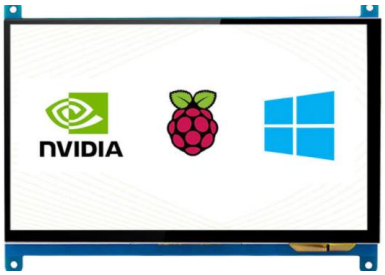


Figure 3.24 Touchscreen Lcd

Figure 3.24 is a 7-inch touchscreen LCD that serves as the display and the user's primary way to interact with the platform.

Figure 3.25 Led Strip

Figure 3.25 is the LED Module that would serve as the lighting for the platform's tray that would also serve as a guide.

Figure 3.26 Casing (Abs Plastic Filament)

Figure 3.26 is the Casing (ABS plastic filament) that would be used for 3d printers. It would serve as the casing of the platform.

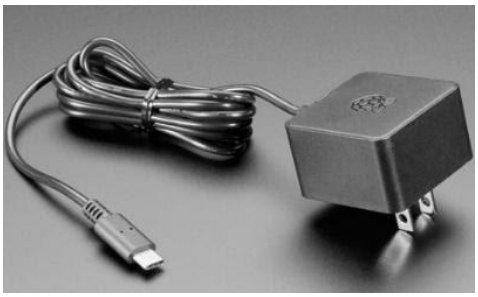


Figure 3.27 Raspberry Pi Power Adapter

Figure 3.27 is the Raspberry Pi Power Adapter. It would be used to power the raspberry pi with its micro USB port used for power connection.

Computation Cost

It includes all the system's raw parts, including the memory, display, power supply, camera, Lightning, casing, and the microcontroller processor.

Table 3.4 Bill Of Materials

Components	Device	Picture	Cost
Processor	Raspberry pi 4b		4,749.75 Php
Memory	SD card		269.00 Php
Display	LCD Touch Screen		3,000.00 Php
Power Supply	Raspberry Pi Power Supply		574.75 Php

Camera	IMX219		1,650.00 Php
Lighting	3 module led light		9.00 Php
Casing	ABS plastic filament		1,000.00 Ph
Total Cost			**11,252.50 Php**

Table 3.4 shows the components and cost when using Raspberry Pi 4 B as the processor. The total cost would be 11,252.50 PHP. Some of these prices may be subject to change.

To get the total cost, you must take the sum of all the hardware that the design option uses.

$$Cost = \sum cost\ of\ components\ (PHP)$$

(equation 4)

Design Option

The proponents have developed 3 design options that would use 3 different algorithms. Regarding the algorithms, the proponents have decided to choose between the SSD using MOBILENET_V2 and Tensorflow, YOLOV4 using CSPDARKNET53, and finally Faster_RCNN using InceptionV2 and Tensorflow. Due to limited resources, the proponents could not test the different algorithms to different processors, so the proponents resorted to using google colab to have a stable and uniform environment.

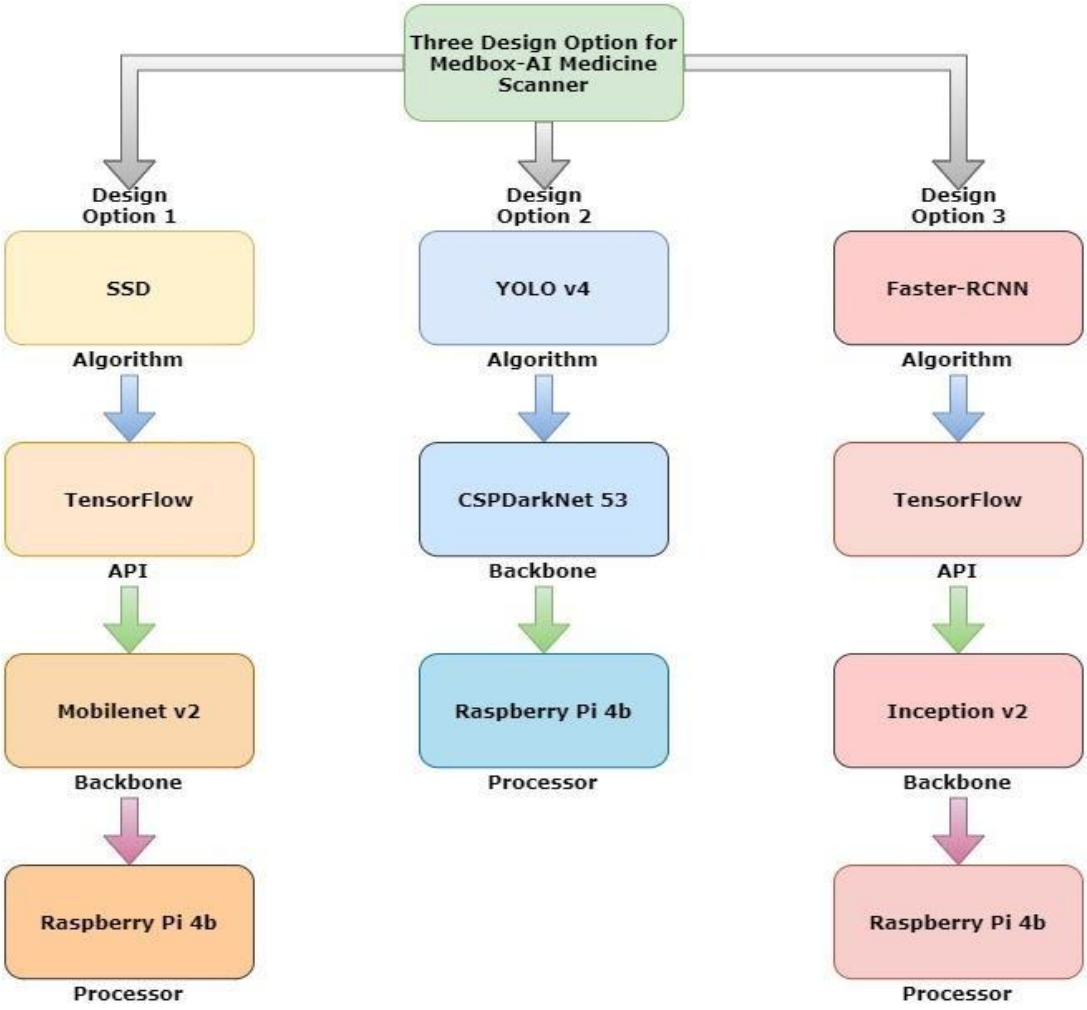


Figure 3.27 Design Tree Hierarchy

Figure 3.27 Illustrates the following algorithms that could be used for different design options. Each of these algorithms has its pros and cons compared to each other.

Design Option 1

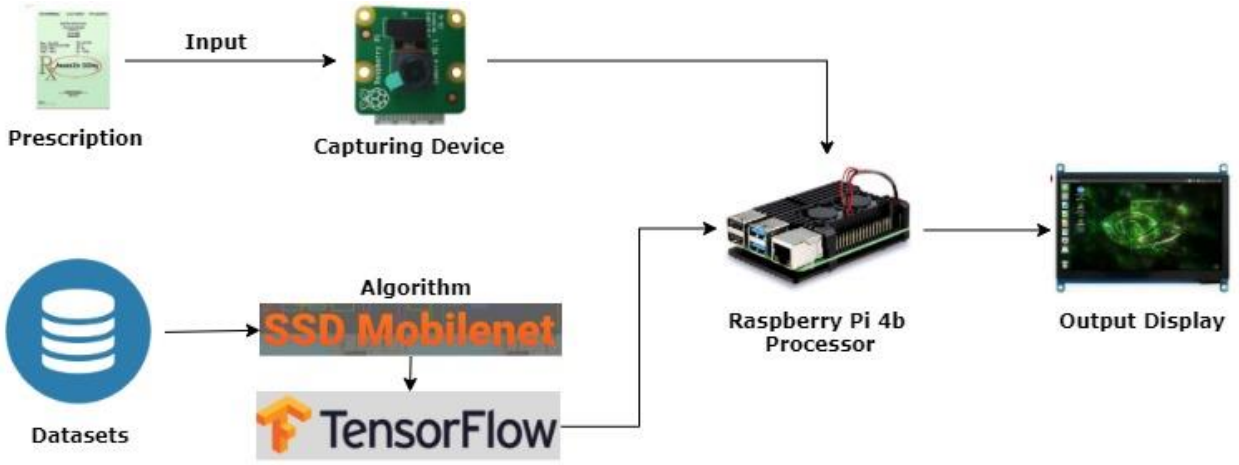


Figure 3.28 Design Option 1

This design choice uses the combination of SSD Mobilenet V2 and the Raspberry Pi processor 4. And again, in terms of the CPU, the specs are high enough to run SSD Mobilenet V2 properly. This choice also runs on SSD Mobilenet V2 to detect multiple objects inside an image without compromising its high accuracy. The raspberry pi 4 is a versatile processor that can run this algorithm correctly and has a reasonably low price for the user's computing power. The benefit of the IMX219 specification is a high-quality camera that can improve the performance of the algorithm. When it comes to price, this design option is on the cheaper side of the spectrum.

Figure 3.29 Ssd Mobilenet Testing Process

This figure shows the process of algorithm testing, which would start with the dataset and feed the said Dataset into SSD Mobilenet V2 that runs on the Google colab environment.

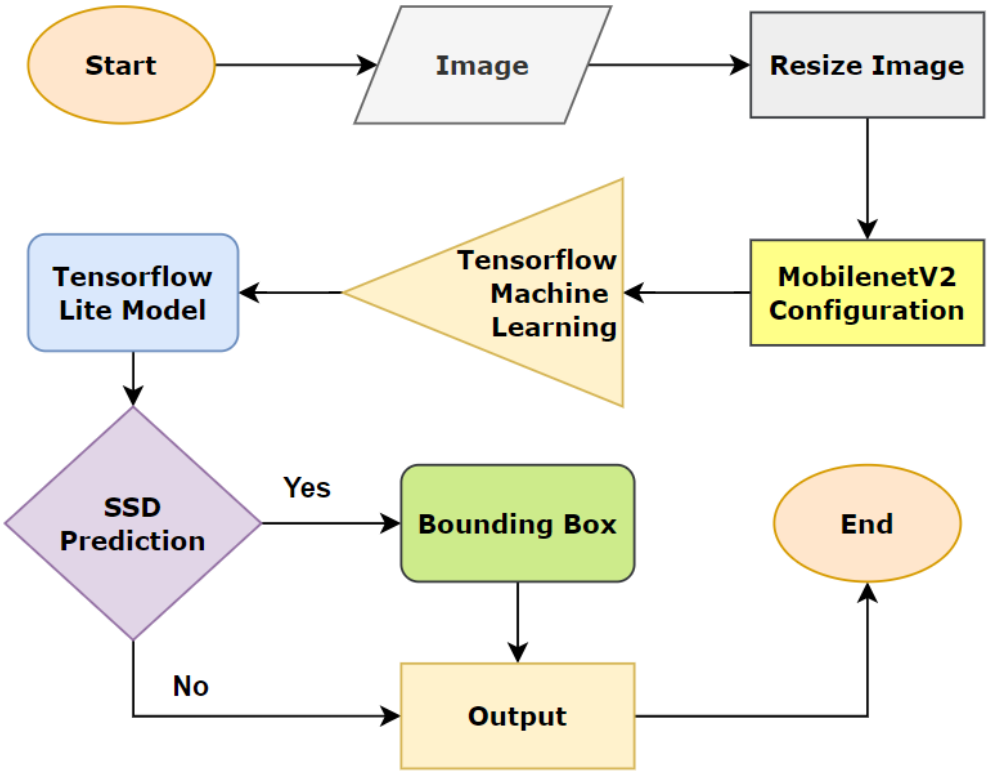


Figure 3.30 Ssd Flowchart

In SSD Flowchart, start inputting the model's image; first, it would resize the image. Then do the different stages of the image's convolutional process and run detections to display the output at the end.

The formula shown below produces average accuracy and speed, which utilizes data input to test the algorithm.

$$Accuracy\ Average = \frac{Total\ Success\ \%}{Number\ of\ Test}$$

(equation 1)

$$Time\ Average = \frac{Total\ Sum\ of\ Test}{Number\ of\ Test}$$

(equation 3)

Table 3.5 Ssd Mobilenet Accuracy And Speed Testing

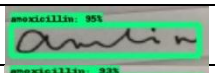
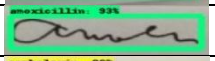
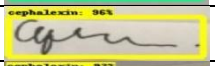
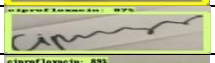
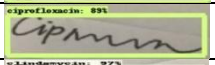
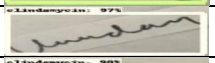
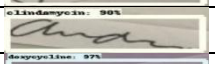
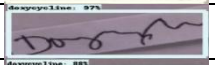
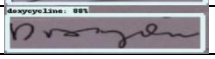
SSD Mobilenet Algorithm					
Number of Tests	**Test Case**	**Expected output (%)**	**Test Picture**	**Accuracy Output (%)**	**Speed Output (ms)**
1	Amoxicillin	100		95	52.4
2	Amoxicillin	100		93	58.2
3	Cephalexin	100		96	54.8
4	Cephalexin	100		92	56.7
5	Ciprofloxacin	100		87	58.5
6	Ciprofloxacin	100		89	54.5
7	Clindamycin	100		97	59.8
8	Clindamycin	100		90	56.6
9	Doxycycline	100		97	60.6
10	Doxycycline	100		88	61.5
Average				**92.4**	**57.4**

Table 3.5 the testing results of SSD Mobilenet. The tests were done twice per antibiotic name. The results displayed were the accuracy and speed of the algorithm for different testings. And based on the results of the accuracy testing, it has an average of 92.4%. Simultaneously, the result of the speed has resulted in an average time of 57.4 ms.

Table 3.6 Design Option 1 Test Result Of Constraints

Constraints	**Result**
Accuracy	92.4 %
Cost	11,252.50 Php
Speed	57.4 ms

The total result of the cost would be Php 11,252.50 that includes all of the components and materials to create the project's hardware side. Do note that these components' costs came from various online shops, and some of them are a direct conversion of cost from USD to PHP. And finally, some of the prices of these components may be subject to change. The accuracy was from the algorithm itself, and after 10 testings, the proponents added all of the accuracies acquired from the testing and divided them by 10 to get the mean average. The mean average that would result from the tests would be 92.4%. The same can be said for the speed, where the proponents recorded the results from the 10 tests and then divided them by 10 to get the mean average. The result of the speed tests would yield 57.4 milliseconds.

Design Option 2

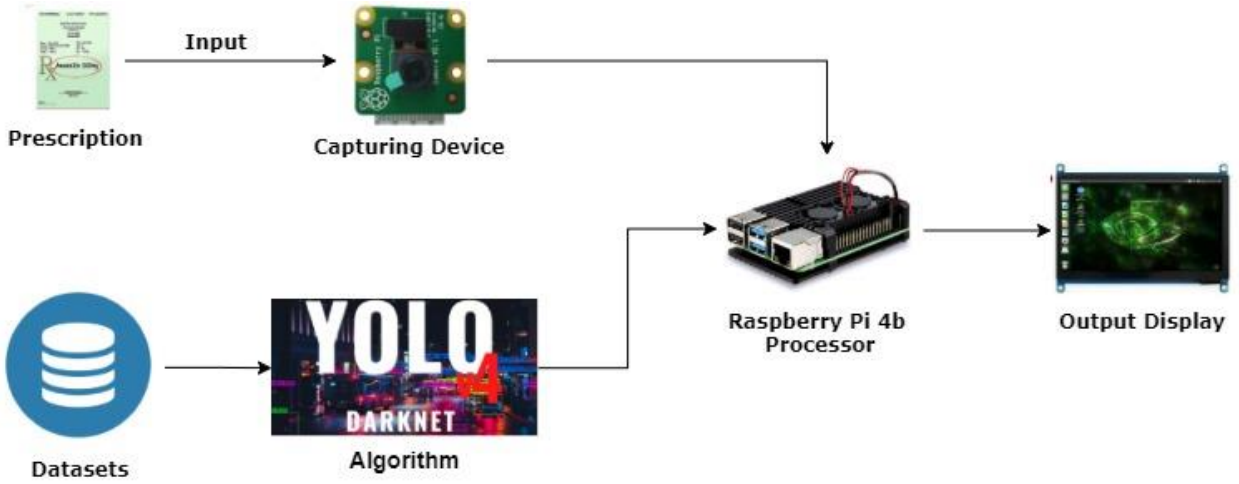


Figure 3.31 Design Option 2

This design option uses the combination of YOLOV4 and Raspberry Pi 4 B. In regards to the processor, the Raspberry Pi 4 has enough processing power to be able to run YOLOV4 properly. This version of Raspberry Pi 4 has an 8GB version that can help with its processing performance. Next would be the algorithm, which is the YOLOV4, which can detect multiple objects at once. In terms of efficiency, its specialty is more on the speed side, which would make this algorithm quicker, but at the cost of accuracy. While the camera, which is the IMX219, has enough specs to provide high-quality images in theory, it would improve the algorithm's accuracy.

Figure 3.32 Yolov4 Testing Process

Figure 3.32 shows the process of algorithm testing for YOLOV4. The process would start with the dataset then the said datasets would be fed to YOLOV4, which is run by Google colab environment.

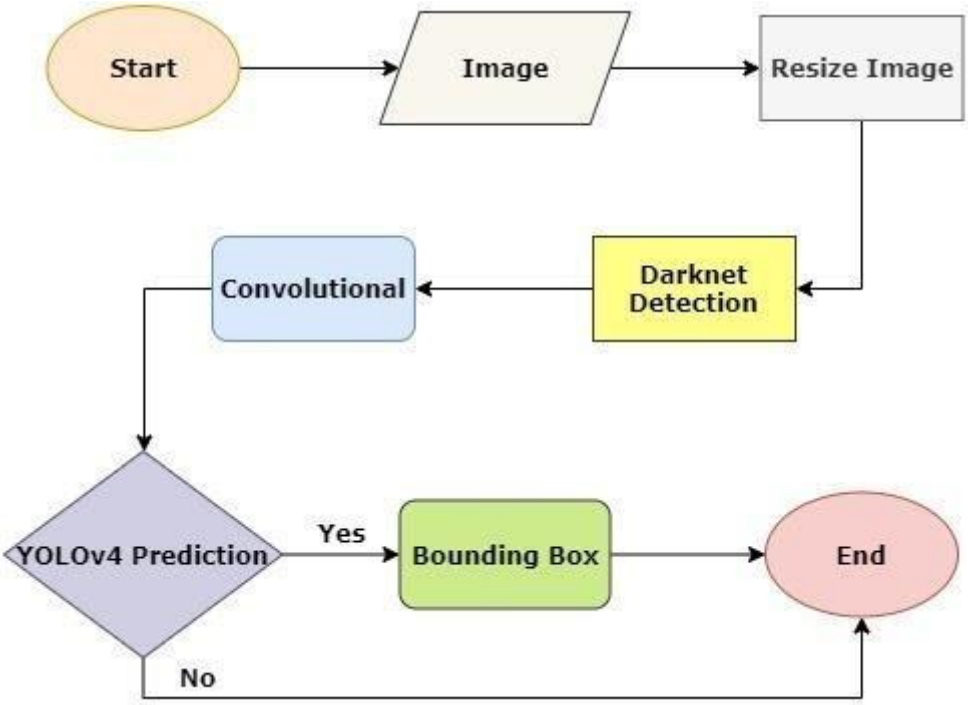


Figure 3.33 Yolov4 Flowchart

This flowchart represents the process that YOLOV4 would go through. It would start by inputting the image and then editing its size after coming to the YOLOV4 detection and predictions. If the Yolo prediction is correct, then the algorithm would end, if not, then the algorithm would put bounding boxes on the image, and the algorithm would end.

The formula shown below produces average accuracy and speed, which utilizes data input to test the algorithm.

$$Accuracy\ Average = \frac{Total\ Success\ \%}{Number\ of\ Test}$$

(equation 1)

$$Time\ Average = \frac{Total\ Sum\ of\ Test}{Number\ of\ Test}$$

(equation 3)

Table 3.7 Yolov4 Accuracy And Speed Testing

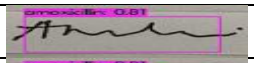
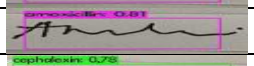
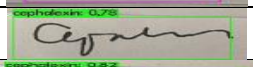
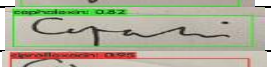
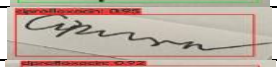
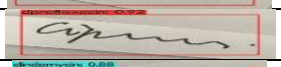
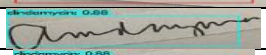
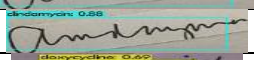
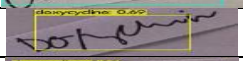
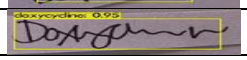
Yolov4 Algorithm					
Number of Tests	**Test Case**	**Expected output (%)**	**Test Picture**	**Accuracy Output (%)**	**Speed Output (ms)**
1	Amoxicillin	100		81	40.8
2	Amoxicillin	100		81	40.7
3	Cephalexin	100		78	40.8
4	Cephalexin	100		82	40.8
5	Ciprofloxacin	100		95	40.8
6	Ciprofloxacin	100		92	40.8
7	Clindamycin	100		88	40.8
8	Clindamycin	100		68	40.7
9	Doxycycline	100		69	40.8
10	Doxycycline	100		95	40.9
Average				**82.9**	**40.8**

Table 3.7 shows the testing results of YOLOV4. The tests were done twice per antibiotic name. The results displayed were the accuracy and speed of the algorithm for different testings. And based on the results of the accuracy testing, it has an average of 82.9%. Simultaneously, the result of the speed has resulted in an average time of 40.8 ms.

Table 3.8 Design Option 2 Test Result Of Constraintstest Result Of Constraints

Constraints	**Result**
Accuracy	82.9 %
Cost	11,252.50 Php
Speed	40.8 ms

This design option's total cost would be 11,252.50 Php that includes the processor and any other components needed for the design option. The team confirmed that some of each component's prices have already directly translated from USD to PHP and that some of the components' prices may be subject to change. The following result would be its average accuracy, which is at 82.9%; which the proponents added all 10 results then divided the total by 10 to get the average accuracy. The same would be valid for the average time consumption, in which the results of each assessment were added together, and the total was divided by 10 to achieve average time consumption. The average time consumption of this design option is 40.8 milliseconds.

Design Option 3

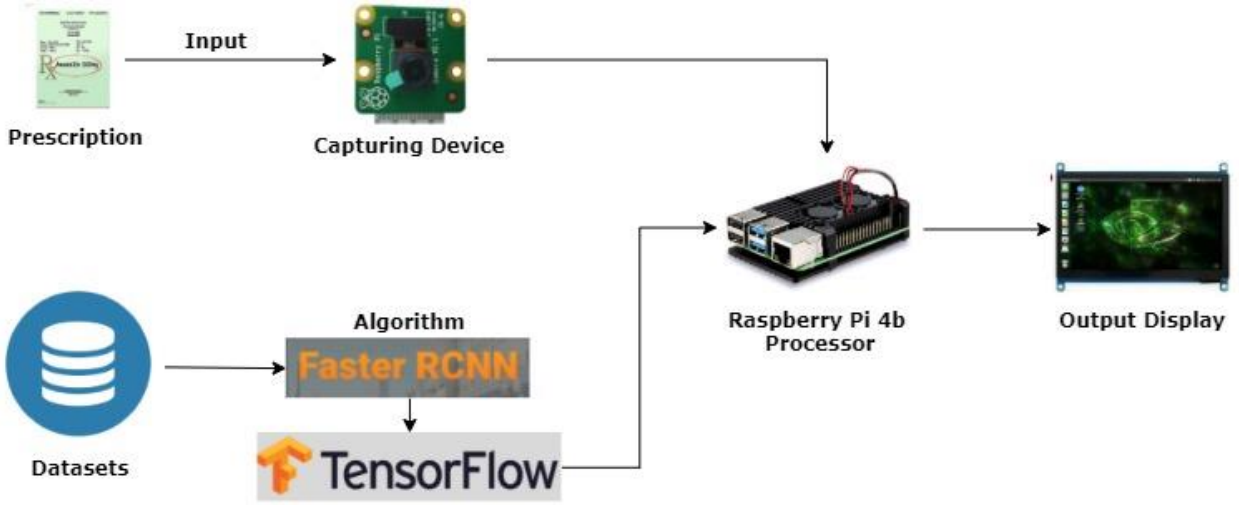


Figure 3.34 Design Option 3

This design option would make use of the Faster RCNN and the Raspberry Pi 4 B. In terms of processors, the latest variant of Raspberry Pi 4 has the capabilities to be able to run the algorithm correctly. And because of its specs, it has the real possibility to improve further the low processing speed of the Faster RCNN, which would result in the perfect balance between speed and accuracy. In terms of algorithm, the Faster RCNN sacrifices speed to have a rating inaccuracy. Theoretically, the algorithm's lack of speed can be accounted for by Raspberry Pi 4 B's efficiency. In the camera portion, the IMX219 can provide high-quality images that can further aid with the algorithm's accuracy.

Figure 3.35 Faster Rcnn Testing Process

Figure 3.35 shows the algorithm testing process for Faster RCNN. The process would start with the datasets and then said datasets would be fed into the Faster RCNN with the backbone of Inception_V2 and Tensorflow as its API, which is run by the Google colab environment.

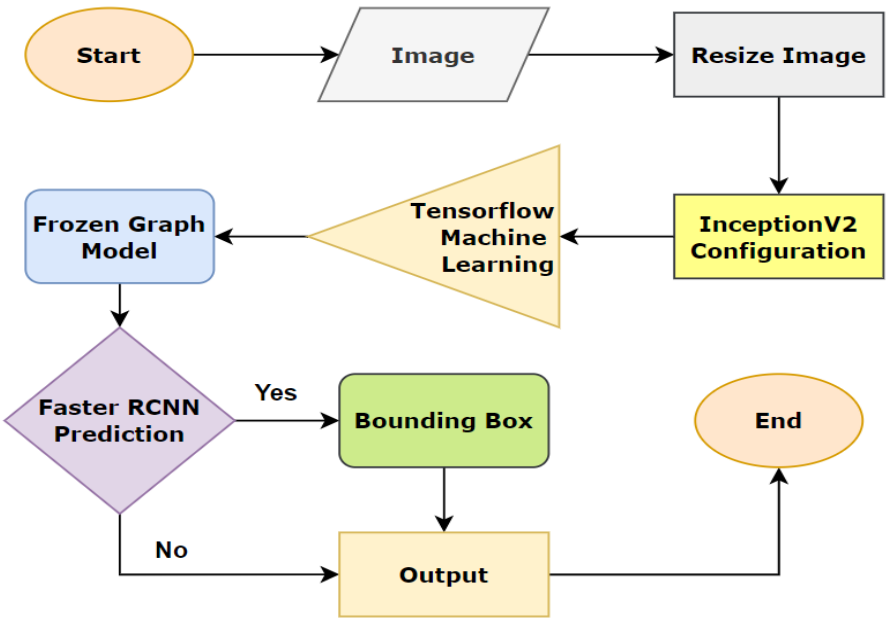


Figure 3.36 Faster Rcnn Flowchart

Figure 3.36 represents the process flow of the Faster - RCNN algorithm. It would with the image in the dataset, which would be resized by the algorithm. Then the resized image would go into the convolutional process and after the convolutional process. And after that is the Faster - RCNN detection. After the algorithm successfully detected the object, then it would proceed to the prediction. If the Faster - RCNN has the correct prediction, then the algorithm would place a bounding box on the object, and if not, then the process would end.

The formula shown below produces average accuracy and speed, which utilizes data input to test the algorithm.

$$Accuracy\ Average = \frac{Total\ Success\ \%}{Number\ of\ Test}$$

(equation 1)

$$Time\ Average = \frac{Total\ Sum\ of\ Test}{Number\ of\ Test}$$

(equation 3)

Table 3.9 Faster Rcnn Accuracy And Speed Testing

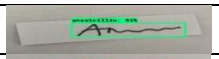
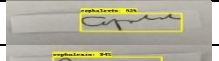
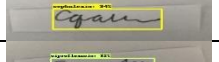
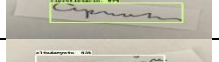
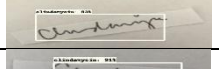
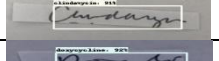
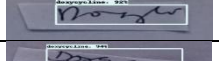
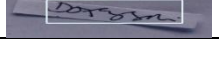
Faster RCNN Algorithm					
Number of Tests	**Test Case**	**Expected output (%)**	**Test Picture**	**Accuracy Output (%)**	**Speed Output (ms)**
1	Amoxicillin	100		93	110.4
2	Amoxicillin	100		90	140.5
3	Cephalexin	100		92	120.6
4	Cephalexin	100		94	120.3
5	Ciprofloxacin	100		82	120.3
6	Ciprofloxacin	100		87	120.1
7	Clindamycin	100		93	120.4
8	Clindamycin	100		91	130.2
9	Doxycycline	100		92	120.4
10	Doxycycline	100		94	130.2
Average				**90.8**	**134.38**

Table 3.9 shows the testing results of Faster RCNN. The tests were done twice per antibiotic name. The results displayed were the accuracy and speed of the algorithm for different testings. And based on the results of the accuracy testing, it has an average of 90.8. Simultaneously, the result of the speed has resulted in an average time of 134.38 ms.

Table 3.10 Design Option 3 Test Result Of Constraints

Constraints	Result
Accuracy	90.8 %
Cost	11,252.50 Php
Speed	134.38 ms

The total costs of this design option would be at 11,252.50 Php. Some components' prices were directly converted from USD to Php, and that their prices may be subject to change. The next is average accuracy, where the total results of the accuracy were divided by 10 to get the average. The average accuracy of this design option is 90.8%. The last one would be the average time consumption, which was

computed by adding all the speed results in the testing and dividing the total results by 10. The average time consumption would be 134.38 milliseconds.

Table 3.11 Summary Of All Design Options

Design Constraints	Design Option 1	Design Option 2	Design Option 3
Accuracy (%)	92.4 %	82.9%	90.8%
Cost (Php)	11,252.50	11,252.50	11,252.50
Time Consumption (ms)	57.4	40.8	134.4

Table 3.11 shows all of the results done per design options. In design option 1, it has an average accuracy of 92.4%, the total cost of 11,252.50 PHP, and average time consumption of 57.4 milliseconds. Design option 2 has an average accuracy of 82.9%, a total cost of 11,252.50 PHP, and average time consumption of 40.8 milliseconds. Design option 3 has an average accuracy of 90.8%, a total cost of 11,252.50 PHP, and average time consumption of 134.38 milliseconds. And by analyzing the table, it is possible to see each design option's strengths and disadvantages, and in Chapter 4, where each design option is compared to each other, to find the best design option for the solution.

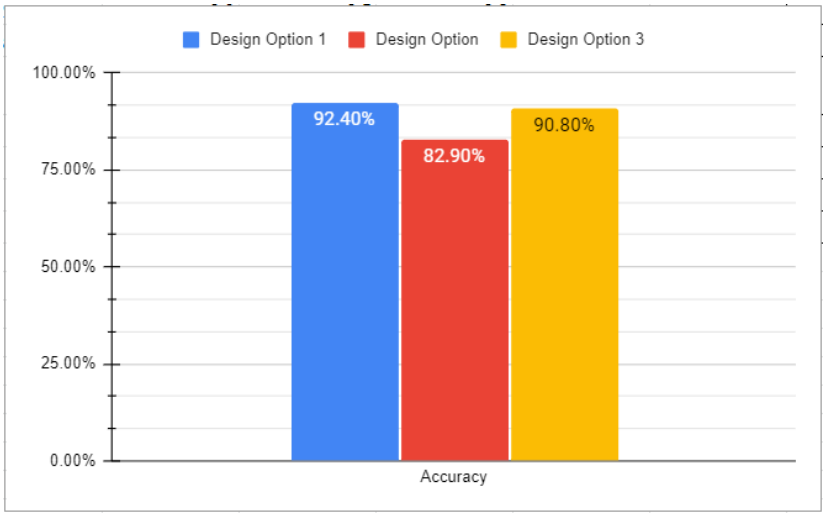


Figure 3.37 Accuracy Value For All Design Option

Figure 3.37 shows the performance of the algorithms based on accuracy. In this case, design option 1 and design option 3 have a clear advantage with 92.40% and 90.80%, while design option 2 lags behind 82.90% of the average accuracy.

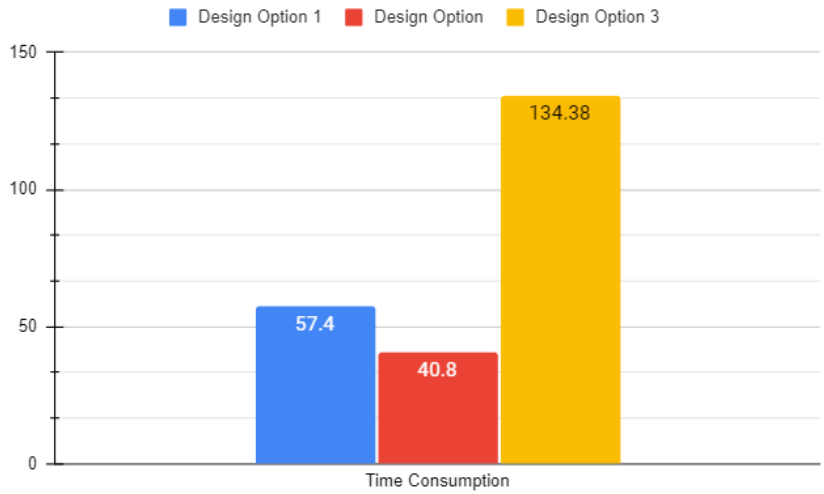


Figure 3.38 Speed Value For All Design Option

Figure 3.38 shows the performance of the different design options in terms of time consumption. Preferably the lower the value, the faster the design option is. In this case, design option 3 has the highest time consumption between the 3, which is 134.38 milliseconds. It means that this design option is the slowest among the 3. While design option 1 has 57.4 milliseconds, only lagging behind design option 2 has 40.8 milliseconds. Design option 2 is the fastest design option among the 3.

CHAPTER IV: CONSTRAINTS, TRADE OFFS, AND STANDARDS

This chapter points out the best design choice, taking into account several constraints as seen. This chapter would also concentrate more on computations that would help proponents select which design choice would be better suited to the solution.

Design Criteria And Constraints

Accuracy

This project is concerned with the output accuracy of the overall system. The design to be implemented must be with high precision and an acceptable rate of failure. Inadequate equipment, low data processing, or human error can lead to inaccurate results that are not very close to the truth. Accuracy can help increase users' satisfaction.

Speed

The proponents must create a device that translates the prescription in a short period. It avoids the hassle for the users and helps them utilize their time in using the platform. Significant delays may cause errors and unnecessary waiting time for the users. The proponents should measure the speed based on the algorithm used. Factors such as algorithm efficiency and simulation environment can affect the rate of the system.

Economical

This design constraint requires achieving the product by improving the business environment. The design must cover a cost that can guarantee the client's requirements. It must also finance and facilitate the maintenance of the product for the long term. For this project, the materials must be cost-efficient as possible, with the design's overall performance justifying the cost.

Design Trade-Offs

Design trade-off is a decision-making process that involves losing one factor or quality to test the design's attainability. This decision allows the advocates to select the best design used for the research, considering both the limitations and the study's goals. In finding the most feasible design, the trade-off analysis was carried out on the design and on the criteria for evaluating which design is ideal for the final design implementation.

Based on Figure 3.27 Design Tree Hierarchy, displays the trade-off flow chart between the various design choices. There are 3 design options, which are the same as the number of algorithms used for the design options.

Table 4.1 Trade-Off Analysis Constraints Value

Constraint	Unit	Design 1	Design 2	Design 3
Accuracy	**%**	**92.4**	**82.9**	**90.8**
Speed	**ms**	**57.4**	**40.8**	**134.4**
Cost	**Php**	**11,252.50**	**11,252.50**	**11,252.50**

Table 4.1, shown above, shows the importance of the proponent testing for each design alternative. The performance value helps assess the best design choices for the device when considering the constraints. Each constraint was chosen to help shape the project to suit the exact needs of the client.

The formula shown below sets out the rank of each design choice. Each value is evaluated using the methods of Kirkwood and Anton Son.

$$Ranking\ Value = \left[\ 1 - \left(\frac{Highest\ Value - Lowest\ Value}{Governing\ Value}\right)\right]\ x\ 5$$

(equation 6)

Accuracy Constraints:

Based on the proponents' interview, the client's goal in the project is to have a high level of reliability in its results, which means that the project must always be consistent and accurate. Therefore, the proponents have agreed to make accuracy their main priority when choosing the project's components and algorithms.

Table 4.2 Accuracy Constraints Ranking For All The Design Options

Accuracy	Value	Ranking Score
Design 1	92.4	5
Design 2	82.9	4.49
Design 3	90.8	4.91

Table 4.2 shows the ranking of the accuracy constraints for all of the existing design options. And based on the data on the table, design option 1 won in accuracy constraints.

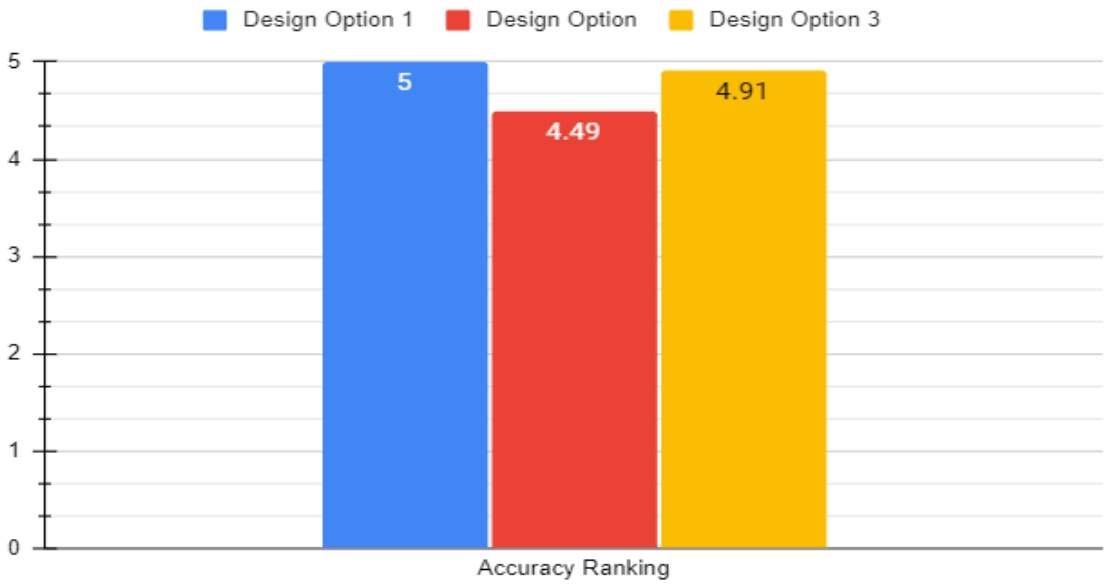


Figure 4.1 Accuracy Ranking Score

Figure 4.1 displays the accuracy ranking score in line graph form. It shows that design option 1 and design option 3 are in the lead in regards to accuracy.

Speed Constraints

During the interview, the proponents found out that the clients don't mind if the algorithm would instantly translate the unreadable prescription, as they prefer to have better accuracy and consistency rather than faster processing speed. So the proponents have decided to lower the importance of speed and sacrifice the processing speed to attain a higher percentage of accuracy.

Table 4.3 Speed Constraints Ranking For All The Design Options

Speed	Value	Ranking Score
Design 1	57.4	3.55
Design 2	40.8	5
Design 3	134.4	1.52

Table 4.3 shows the ranking of all design options in regards to speed. And when it comes to speed, design option 2 is ahead of all of them.

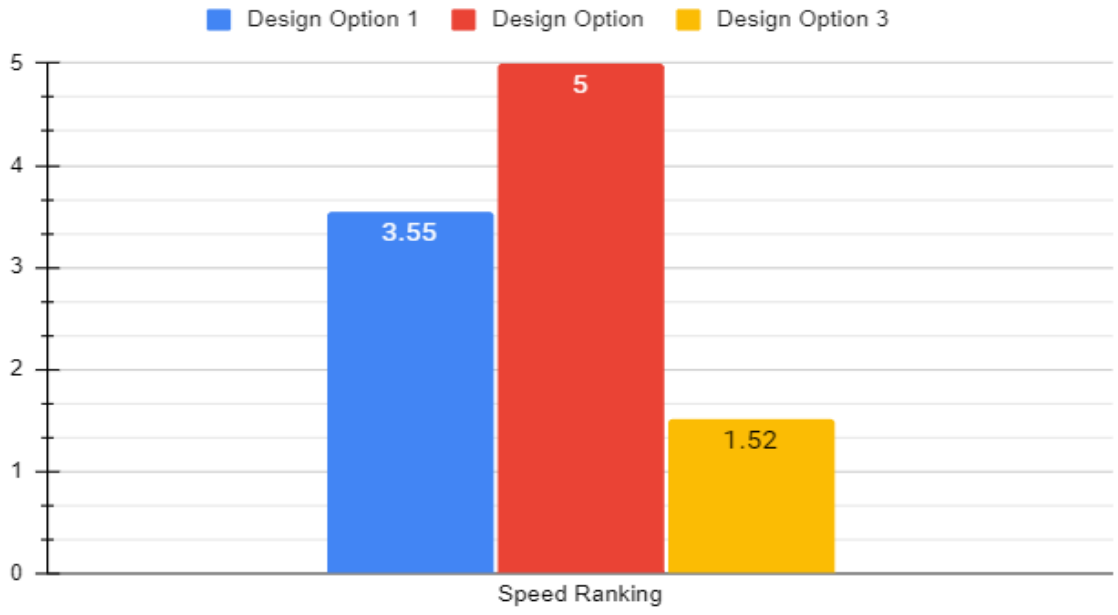


Figure 4.2 Speed Ranking Score

Figure 4.2 is the graph that shows the ranking in regards to speed. If speed compares different design options, then design option 2 is preferred.

Economic Constraints:

The last one would be Economic constraints, which can affect the project based on the interviews. During the client's interview, most interviewees were somewhat worried about the project's selling price. They prefer that as long as the project's price is reasonable, they would gladly buy the project. So the proponents have decided that the Economic constraints would come next in priority after the reliability.

Table 4.4 Economic Constraints Ranking For All The Design Options

Cost	Value	Ranking Score
Design 1	11,252.50	5
Design 2	11,252.50	5
Design 3	11,252.50	5

Table 4.4 shows the ranking score of the design options regarding economic constraints. Regarding economic constraint, all design options are preferred in terms of the economic cost, for they all have the same components.

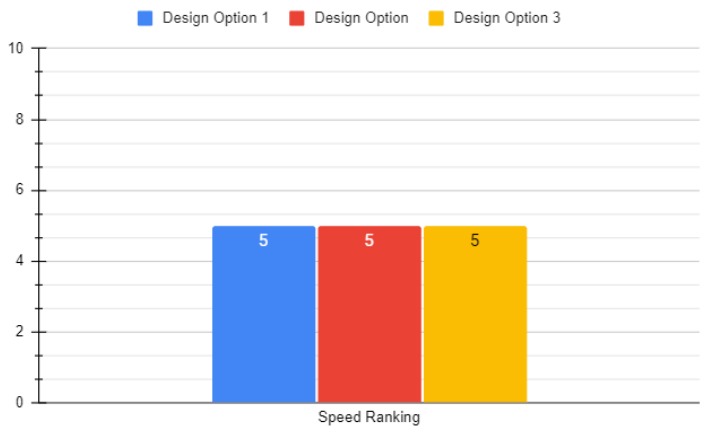


Figure 4.3 Economic Ranking Score

Figure 4.3 shows the Economic Ranking Score in line graph form. It clearly shows that design option 2, design option 3, and design option 6 are the ones in the lead concerning economic constraints

Table 4.5 Summary Ranking Of Design Constraints For Each Design Option.

Design Option	Accuracy Rank	Speed Ranks	Cost Rank
Design 1	5	3.55	5
Design 2	4.49	5	5
Design 3	4.91	1.52	5

Table 4.5 shows the summary ranking of design constraints for each design option. Different design options lead to different design constraints. When it comes to accuracy, design option 1 is preferable. In regards to speed, design option 2 is preferable in this category. Finally, in terms of cost, they all have the same ranking.

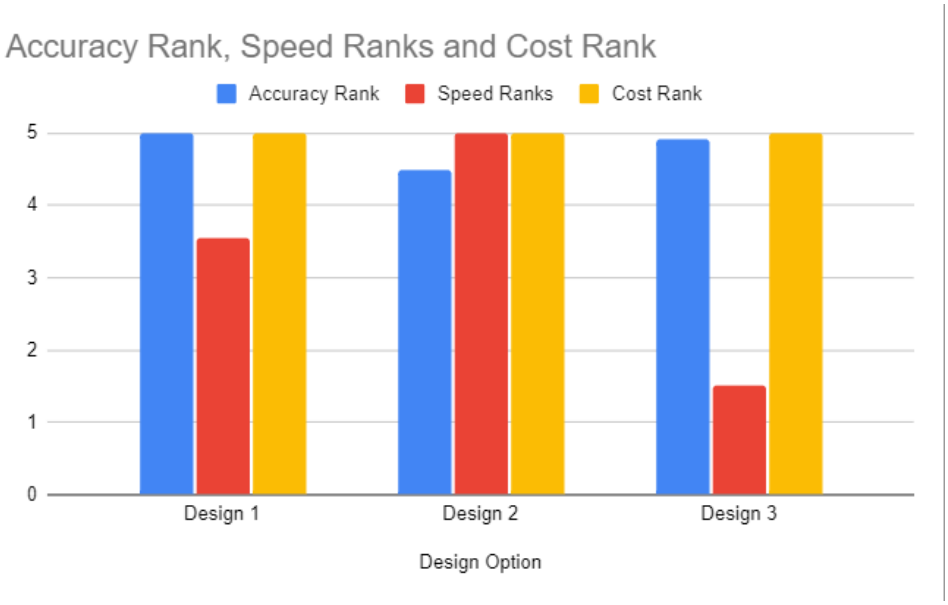


Figure 4.4 Summary Ranking Of Design Constraints For Each Design Option

Figure 4.4 shows the ranking of design constraints for each design option. It also displays which design options have the advantage in regards to the different constraints. And as displayed on the table, no design option has a clear advantage over each other.

Trade-Off Analysis

Table 4.6 Design Constraints Importance Factor

Criteria	Importance Factor
Accuracy	0.6
Speed	0.2
Cost	0.2

Table 4.6 indicates the precision, speed, and cost of the design restriction. The client needs more precision and less importance in cost and speed, which gives a criterion importance factor of 60% in accuracy and a criterion importance factor of 20% for each in speed and cost

Table 4.7 Trade-Off Analysis Of All Designs

Importance Factor	Design 1	Design 2	Design 3
0.6	5	4.49	4.91
0.2	3.55	5	5
0.2	5	5	5
Value = $\sum (i * ranking)$	3	2.69	2.95
	0.71	1	0.30
	1	1	1
Total	**4.71**	**4.69**	**4.25**

Table 4.7 shows the overall ranking using the criterion importance factor. Design 1 got the highest ranking with 4.71, while design 3 got the lowest at 4.25. Each value is assessed using the formula below:

$$Value = \sum(i * ranking)$$

(equation 5)

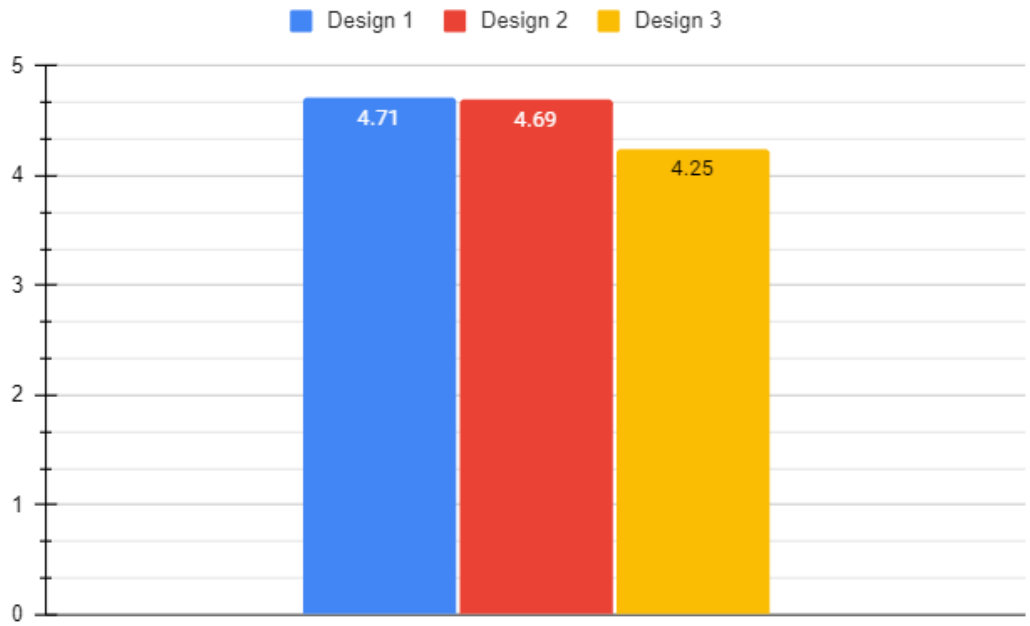


Figure 4.5 Overall Ranking Of Trade-Off Analysis

Figure 4.5 shows the overall ranking of the trade-off analysis. Design 1 got the highest ranking with 4.71, while design 3 got the lowest at 4.25.

Sensitivity Analysis 1

Table 4.8 Design Constraints Importance Factor Of Sensitivity Analysis 1

Criteria	Importance Factor
Accuracy	0.4
Speed	0.1
Cost	0.5

Table 4.8 shows the first sensitivity analysis of all design options. In the Analysis, the priority criteria of importance are the cost with the value of 50%. The accuracy is the next priority of importance of 40%, and the less priority is the speed with the value of 10%.

Table 4.9 Sensitivity Analysis 1 Of All Designs

Importance Factor	Design 1	Design 2	Design 3
0.4	5	4.49	4.91
0.1	3.55	5	1.52
0.5	5	5	5

Value = (i*ranking)	2	1.895	1.96
	0.36	0.5	0.15
	2.50	2.5	2.5
Total	**4.86**	**4.80**	**4.62**

Table 4.9 shows the overall ranking in sensitivity 1 using the criterion importance factor. Design 1 got the highest ranking with 4.86, while design 3 got the lowest at 4.62. Each value was evaluated using the formula below:

$$Value = \sum(i * ranking)$$

(equation 5)

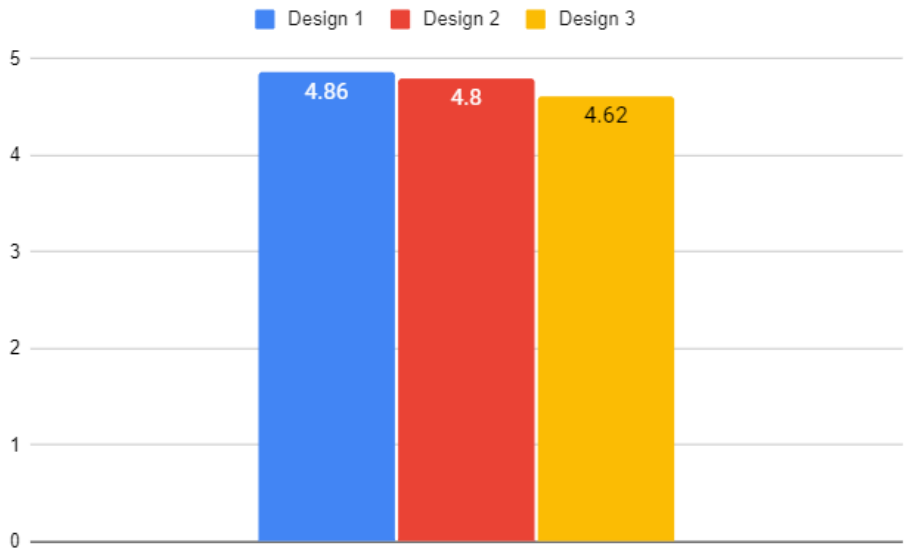


Figure 4.6 Overall Ranking Of Sensitivity Analysis 1

Figure 4.6 shows the overall ranking of sensitivity analysis 1. Design 1 got the highest ranking with 4.86, while design 3 got the lowest at 4.62.

Sensitivity Analysis 2

Table 4.10 Design Constraints Importance Factor Of Sensitivity Analysis 2

Criteria	Importance Factor
Accuracy	0.3
Speed	0.5
Cost	0.2

Table 4.10 shows the second sensitivity analysis of all design options. In the Analysis, the priority criteria of importance are the speed with the value of 50%. The accuracy is the next priority of importance of 30%, and the less priority is the cost with the value of 20%.

Table 4.11 Sensitivity Analysis 2 Of All Designs

Importance Factor	Design 1	Design 2	Design 3
0.3	5	4.49	4.91
0.5	3.55	5	1.52
0.2	5	5	5

Value = (i*ranking)	1.5	1.35	1.47
	1.78	2.50	0.76
	1	1	1
Total	**4.28**	**4.85**	**3.23**

Table 4.11 shows the overall ranking in sensitivity analysis 2 using the criterion importance factor. Design 2 got the highest ranking with 4.85, while design 3 got the lowest at 3.23. Each value was evaluated using the formula below:

$$Value = \sum(i * ranking)$$

(equation 5)

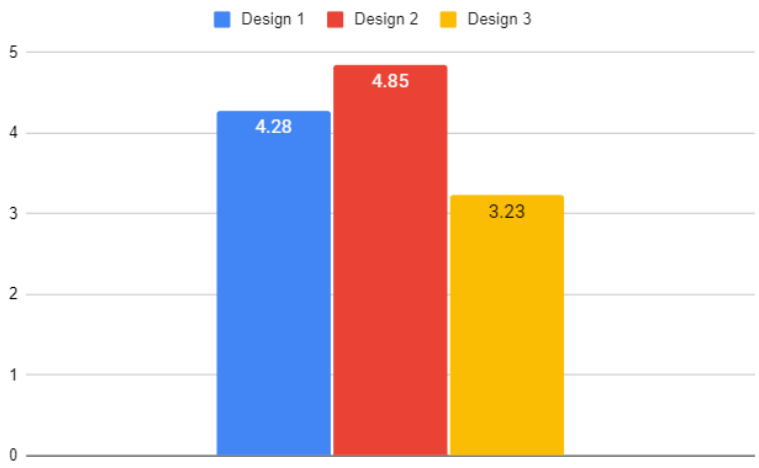


Figure 4.7 Overall Ranking Of Sensitivity Analysis 2

Figure 4.7 shows the overall ranking of sensitivity analysis 2. Design 2 got the highest ranking with 4.85 while design 3 got the lowest with 3.23

Sensitivity Analysis 3

Table 4.12 Design Constraints Importance Factor Sensitivity Analysis 2

Criteria	Importance Factor
Accuracy	0.3
Speed	0.3
Cost	0.3

Table 4.12 shows the third sensitivity analysis of all design options. In the Analysis, the priority criteria of importance adjust such that the different constraints have an equal value of 33%.

Table 4.13 Sensitivity Analysis 3 Of All Designs

Importance Factor	Design 1	Design 2	Design 3
0.3	5	4.49	4.91
0.3	3.55	5	1.52
0.3	5	5	5

Value = (i*ranking)	1.5	1.35	1.47
	1.07	1.5	0.46
	1.5	1.5	1.5
Total	**4.07**	**4.35**	**3.43**

Table 4.13 shows the overall ranking in sensitivity analysis 3 using the criterion importance factor. Design 2 got the highest ranking with 4.35, while design 3 got the lowest at 3.43. Each value was evaluated using the formula below:

$$Value = \sum(i * ranking)$$

(equation 5)

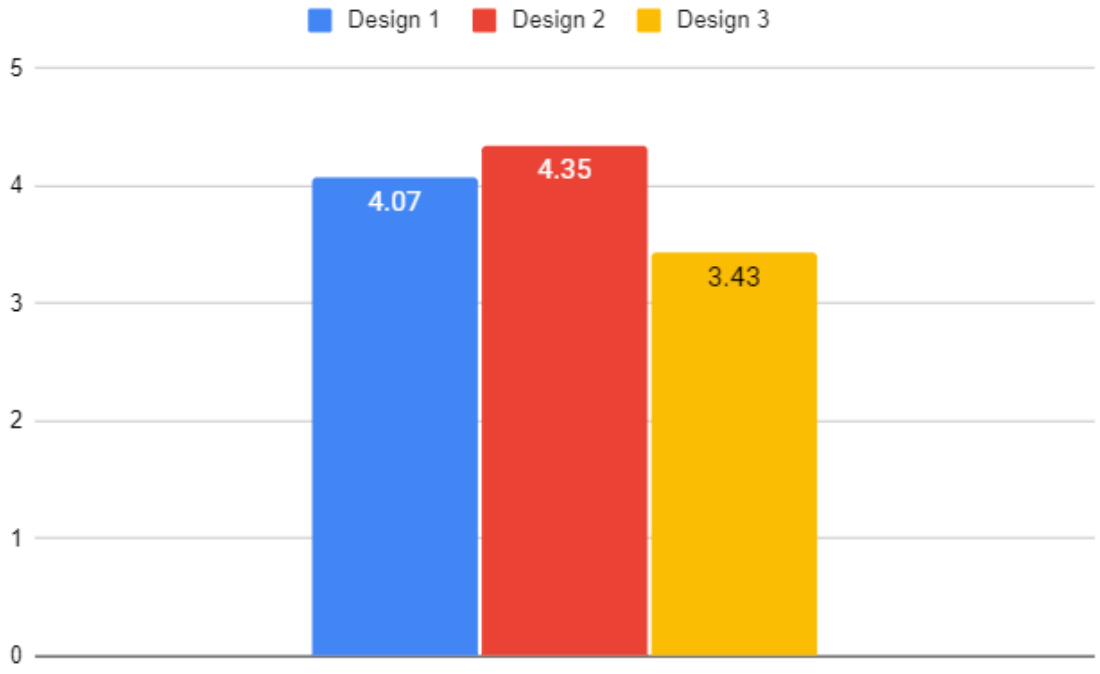


Figure 4.8 Overall Ranking Of Sensitivity Analysis 3

Figure 4.8 shows the overall ranking of sensitivity analysis 3. Design 2 got the highest ranking with 4.35, while design 3 got the lowest at 3.43.

Table 4.14 Summary Of Overall Ranking Of Trade-Off, Sensitivity Analysis Of All Design Options

Overall Ranking	Design Option 1	Design Option 2	Design Option 3
Trade-off Analysis	4.71	4.69	2.30
SA1	4.86	4.80	4.62
SA2	4.28	4.85	3.23
SA3	4.07	4.35	3.43

Table 4.14 shows the overall ranking of trade-off, sensitivity analysis of all design options. Considering the criterion of importance given by the client, design option 4 is the best design. Applying the Kirkwood and Anton Son formula, the corresponding ranking is computed, which resulted in design option 4 is the best design.

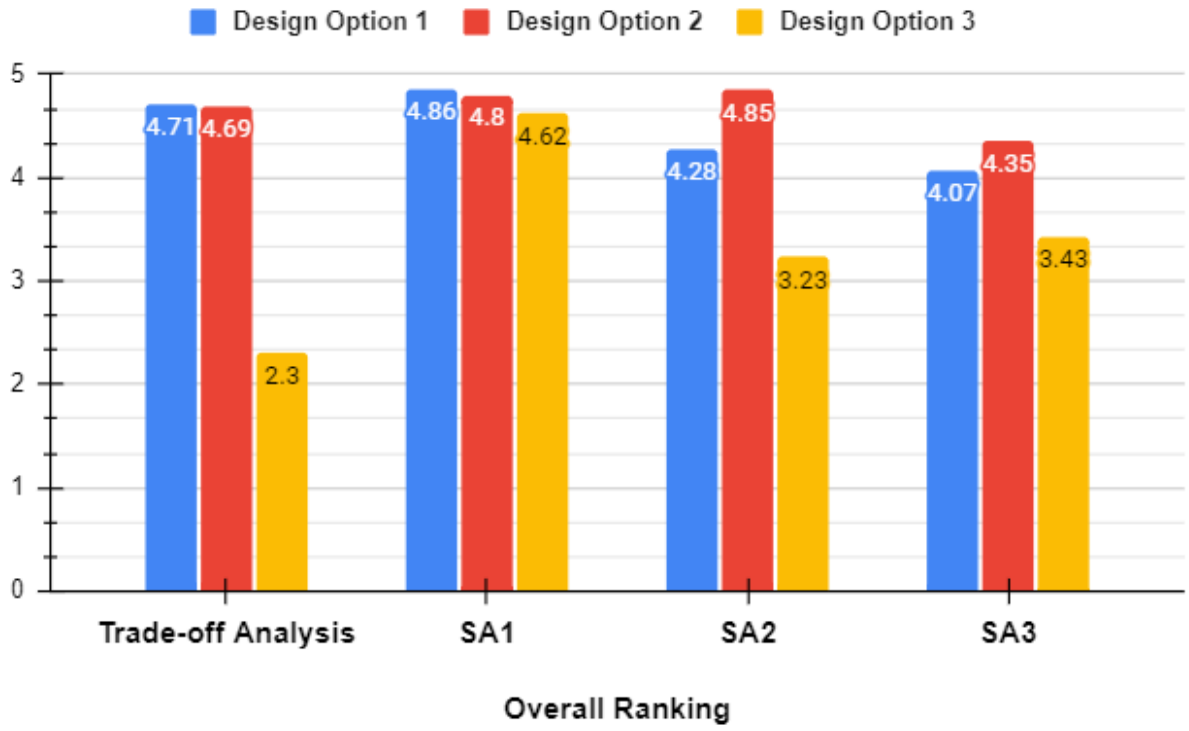


Figure 4.9 Summary Of Overall Ranking Of Trade-Off, Sensitivity Analysis Of All Design Options

As shown in figure 4.9, there are 3 Sensitivity Analyses performed aside from the Trade of Analysis for all the design options. The yellow bar indicates the overall ranking in the Trade-off analysis. Gray bar for Sensitivity analysis 1. The orange bar for Sensitivity analysis 2, and the blue bar for sensitivity analysis 3. From this bar chart, design option 1 won the trade-off and each sensitivity analysis.

CHAPTER V: FINAL DESIGN

This chapter would give more technical information regarding the design option chosen based on the trade-off analysis results. This chapter would also consist of the final design's system flowchart, the gathering of datasets, the algorithm's training process, the GUI, prototype, user manual, and the code. The prototype section consists of a schematic diagram, components, and the prototype overview. But due to recent events, the proponents could not procure the hardware needed for the winning design option and could not test the algorithm to their prospective hardware. The only testing done was in regards to the algorithm for the different design options. The proponents can only provide the GUI, technical drawing of the final device, and the different components needed for the winning design option.

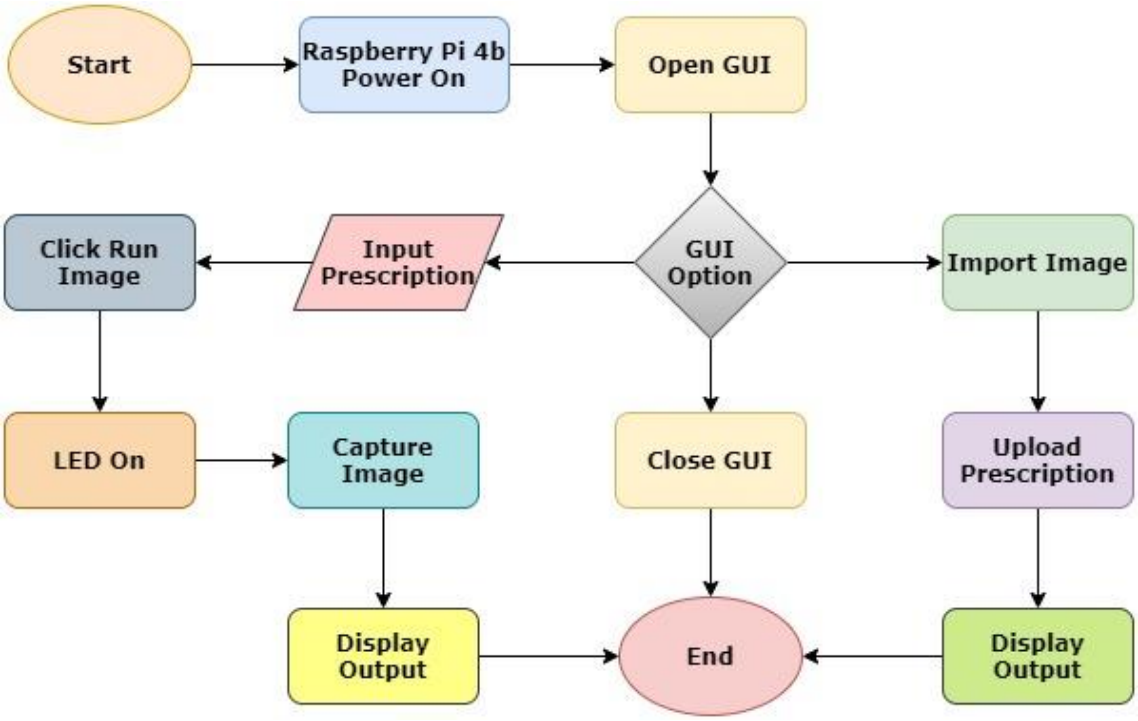


Figure 5.1 Final System Flowchart Of Medbox - Ai

Figure 5.1 shows the whole process of the device itself. It would start with the powering on of the raspberry pi. After powering up, it would open a GUI that would let the user choose between either importing an image to be translated or directly inputting the image in front of the device's camera. If the user wishes to import a part of the image, the system prompts the user to upload the prescription they wished to translate. After the translation of the file, The translated Image is shown immediately on the LCD, and the process would end. If the user prefers a different choice, the user also needs to place the picture in the prescription tray directly below the camera. After placing a prescription and clicking run, the image would also be captured by the system, and the system would then translate the captured image and present the output to the LCD screen. After that, the process would end.

Based on the trade analysis, the winning design option would be design option 1, which has the SSD mobilenet as the algorithm with Raspberry Pi 4 Model B. In the trade analysis, design option 1 always comes out on top except from the trade analysis with a higher priority on speed where it would come out as second with almost little difference in the value of speed compared to the fastest algorithm.

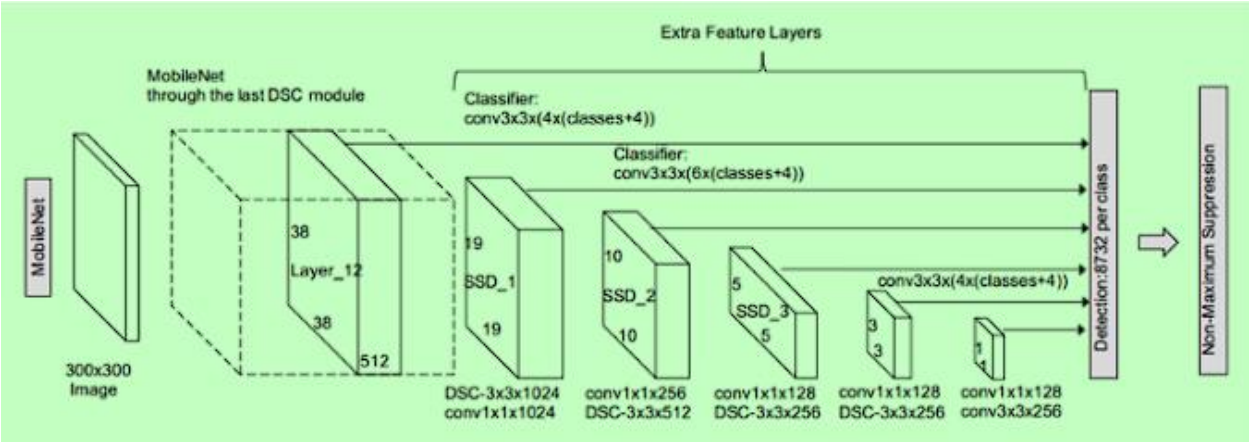


Figure 5.2 Ssd Mobilenet Architecture

The SSD architecture is a single convolution network that learns to predict bounding box locations and classify these locations in one pass. Hence, SSD can be trained end-to-end. The SSD network consists of base architecture (MobileNet in this case) followed by several convolution layers.

Gathering Of Dataset

Before the training can begin, the developers first contacted 2 doctor friends of a member and asked them to provide our datasets. In this case, 5 antibiotic names were handwritten by 2 doctors about 200 times, which would get the proponents 1000 handwritten names

In figure 3.3 shows the process flow of creating the dataset needed for the algorithm. It would start with the sample prescription, capturing prescriptions, annotation, augmentation, and finally, the gathered dataset.

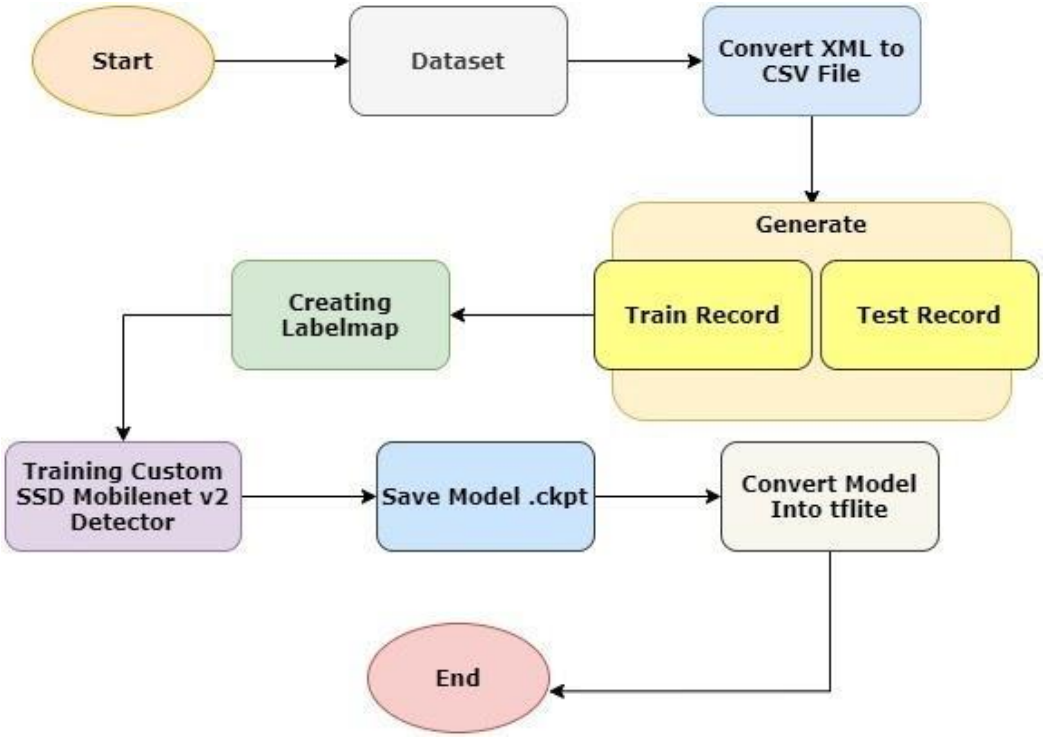


Figure 5.3 Training Flowchart Of Ssd Mobilenetv2

This figure shows the training process for the SSD Mobilenet. It would start with the dataset, and then the dataset would be converted from XML to CSV File. After that, the dataset would be generated into the training record and test record. The next step would be the creation of the label map. The next step would be the training of the Custom SSD Mobilenet v2 Detector, and after the training, the output would be the model itself, and finally, the model output would be converted into tf lite then the training process would end.

Training Process

```
[17] !python xml_to_csv.py

     Successfully converted xml to csv.
     Successfully converted xml to csv.
```

Figure 5.4 Converting Xml To Csv File

This figure shows the first process of training, which is the conversion process of XML to CSV file.

```
[18] !python generate_tfrecord.py --csv_input=images/train_labels.csv --image_dir=images/train --output_path=train.record
     WARNING:tensorflow:From generate_tfrecord.py:108: The name tf.app.run is deprecated. Please use tf.compat.v1.app.run instead.
     WARNING:tensorflow:From generate_tfrecord.py:94: The name tf.python_io.TFRecordWriter is deprecated. Please use tf.io.TFRecordWriter instead.
     W0112 16:54:28.403234 139775415768960 module_wrapper.py:139] From generate_tfrecord.py:94: The name tf.python_io.TFRecordWriter is deprecated. Please use tf.io.TFRecordWriter instead.
     WARNING:tensorflow:From generate_tfrecord.py:53: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile instead.
     W0112 16:54:28.659104 139775415768960 module_wrapper.py:139] From generate_tfrecord.py:53: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile instead.
     Successfully created the TFRecords: /content/gdrive/My Drive/tensorflow/models/research/object_detection/train.record
[19] !python generate_tfrecord.py --csv_input=images/test_labels.csv --image_dir=images/test --output_path=test.record
     WARNING:tensorflow:From generate_tfrecord.py:108: The name tf.app.run is deprecated. Please use tf.compat.v1.app.run instead.
     WARNING:tensorflow:From generate_tfrecord.py:94: The name tf.python_io.TFRecordWriter is deprecated. Please use tf.io.TFRecordWriter instead.
     W0112 17:04:25.811920 140037019096960 module_wrapper.py:139] From generate_tfrecord.py:94: The name tf.python_io.TFRecordWriter is deprecated. Please use tf.io.TFRecordWriter instead.
     WARNING:tensorflow:From generate_tfrecord.py:53: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile instead.
     W0112 17:04:25.884523 140037019096960 module_wrapper.py:139] From generate_tfrecord.py:53: The name tf.gfile.GFile is deprecated. Please use tf.io.gfile.GFile instead.
     Successfully created the TFRecords: /content/gdrive/My Drive/tensorflow/models/research/object_detection/test.record
```

Figure 5.5 Training Generation

This figure shows the second part of the training process ,which is generating the train record file.

```
I0112 17:19:39.804836 140253677963136 learning.py:507] global step 406: loss = 4.7685 (0.961 sec/step)
INFO:tensorflow:global step 407: loss = 5.5816 (0.989 sec/step)
I0112 17:19:40.795563 140253677963136 learning.py:507] global step 407: loss = 5.5816 (0.989 sec/step)
INFO:tensorflow:global step 408: loss = 4.8085 (1.000 sec/step)
I0112 17:19:41.797538 140253677963136 learning.py:507] global step 408: loss = 4.8085 (1.000 sec/step)
INFO:tensorflow:global step 409: loss = 4.9517 (0.963 sec/step)
I0112 17:19:42.762539 140253677963136 learning.py:507] global step 409: loss = 4.9517 (0.963 sec/step)
INFO:tensorflow:global step 410: loss = 5.2803 (0.960 sec/step)
I0112 17:19:43.724421 140253677963136 learning.py:507] global step 410: loss = 5.2803 (0.960 sec/step)
INFO:tensorflow:global step 411: loss = 5.3811 (0.965 sec/step)
I0112 17:19:44.691484 140253677963136 learning.py:507] global step 411: loss = 5.3811 (0.965 sec/step)
INFO:tensorflow:global step 412: loss = 5.1918 (0.962 sec/step)
I0112 17:19:45.655139 140253677963136 learning.py:507] global step 412: loss = 5.1918 (0.962 sec/step)
INFO:tensorflow:global step 413: loss = 4.9327 (0.954 sec/step)
I0112 17:19:46.610570 140253677963136 learning.py:507] global step 413: loss = 4.9327 (0.954 sec/step)
INFO:tensorflow:global step 414: loss = 3.4051 (0.951 sec/step)
I0112 17:19:47.563237 140253677963136 learning.py:507] global step 414: loss = 3.4051 (0.951 sec/step)
INFO:tensorflow:global step 415: loss = 5.4794 (0.971 sec/step)
I0112 17:19:48.536370 140253677963136 learning.py:507] global step 415: loss = 5.4794 (0.971 sec/step)
INFO:tensorflow:global step 416: loss = 4.6818 (0.965 sec/step)
I0112 17:19:49.502850 140253677963136 learning.py:507] global step 416: loss = 4.6818 (0.965 sec/step)
INFO:tensorflow:global step 417: loss = 3.1425 (0.957 sec/step)
I0112 17:19:50.460990 140253677963136 learning.py:507] global step 417: loss = 3.1425 (0.957 sec/step)
INFO:tensorflow:global step 418: loss = 7.6951 (0.990 sec/step)
I0112 17:19:51.453106 140253677963136 learning.py:507] global step 418: loss = 7.6951 (0.990 sec/step)
INFO:tensorflow:global step 419: loss = 3.7623 (0.981 sec/step)
I0112 17:19:52.435721 140253677963136 learning.py:507] global step 419: loss = 3.7623 (0.981 sec/step)
INFO:tensorflow:global step 420: loss = 5.7922 (0.969 sec/step)
I0112 17:19:53.406541 140253677963136 learning.py:507] global step 420: loss = 5.7922 (0.969 sec/step)
INFO:tensorflow:global step 421: loss = 5.0560 (0.962 sec/step)
I0112 17:19:54.369972 140253677963136 learning.py:507] global step 421: loss = 5.0560 (0.962 sec/step)
INFO:tensorflow:global step 422: loss = 4.7174 (0.973 sec/step)
I0112 17:19:55.344832 140253677963136 learning.py:507] global step 422: loss = 4.7174 (0.973 sec/step)
```

Figure 5.6 Training Process

This figure shows the training process and saving the cpkt model.

```
[46] !set CONFIG_FILE=/content//gdrive/My Drive/tensorflow/models/research/object_detection/training/ssd_mobilenet_v2_quantized_300x300_coco.config
     !set CHECKPOINT_PATH+=/content//gdrive/My Drive/tensorflow/models/research/object_detection/training/training/model.ckpt-577
     !set OUTPUT_DIR+=/content//gdrive/My Drive/tensorflow/models/research/object_detection/TFLite_model

[47] !python export_tflite_ssd_graph.py --pipeline_config_path=%CONFIG_FILE% --trained_checkpoint_prefix=%CHECKPOINT_PATH% --output_directory=%OUTPUT_DIR% --add_postprocessing_op=true
```

Figure 5.7 Conversion Of The Model

This figure shows the conversion of the model into a pb model.

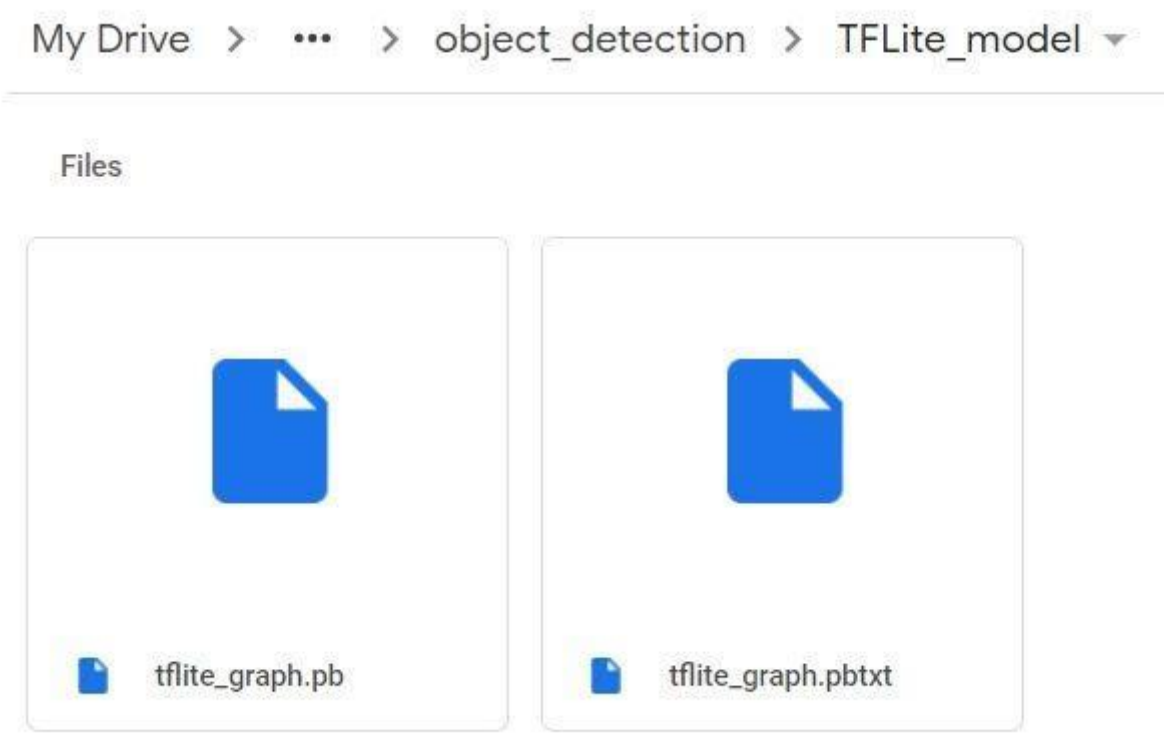


Figure 5.7 Conversion Pb Output
This figure shows the converted pb model

```
[19] toco --graph_def_file='TFLite_model/tflite_graph.pb' --output_file=output.tflite --input_format=TENSORFLOW_GRAPHDEF --output_format=TFLITE --input_shape=300,300 --input_array=ResizeBi
     Successfully converted pb to tflite
```

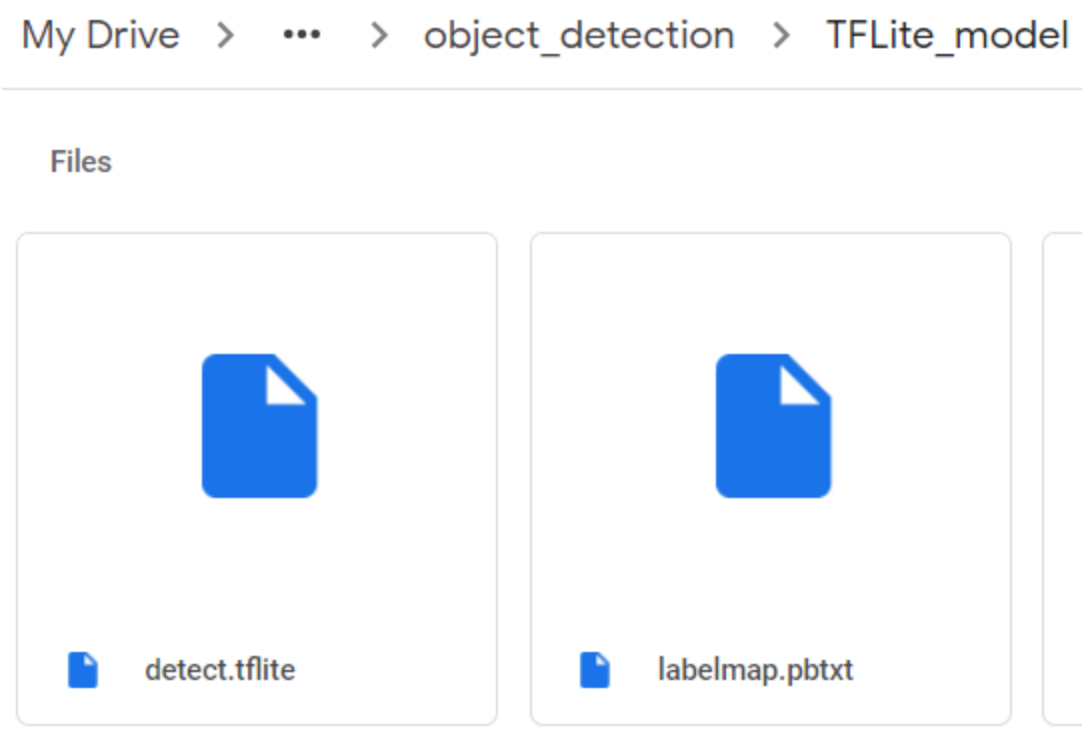


Figure 5.8 Conversion Tflite Output

And finally, this figure shows the converted tflite model with labelmap pbtxt

Graphical User Interface (Gui)

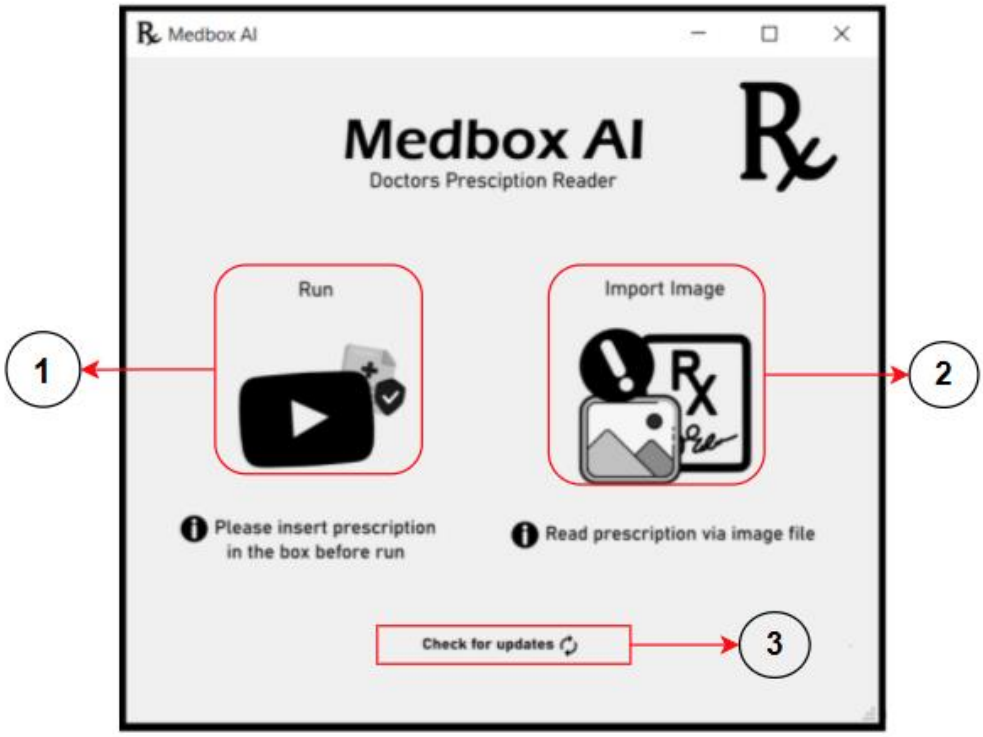


Figure 5.9 Medbox-Ai Main Interface

Figure 5.9 shows the main GUI of the platform. The GUI was pretty much straightforward and only came with 3 buttons. And these are the run button, import image button, and then check for updates button.

1. **Run button** - **Run button** - When the run button is pressed, the algorithm would immediately capture the prescription placed on the prescription tray.

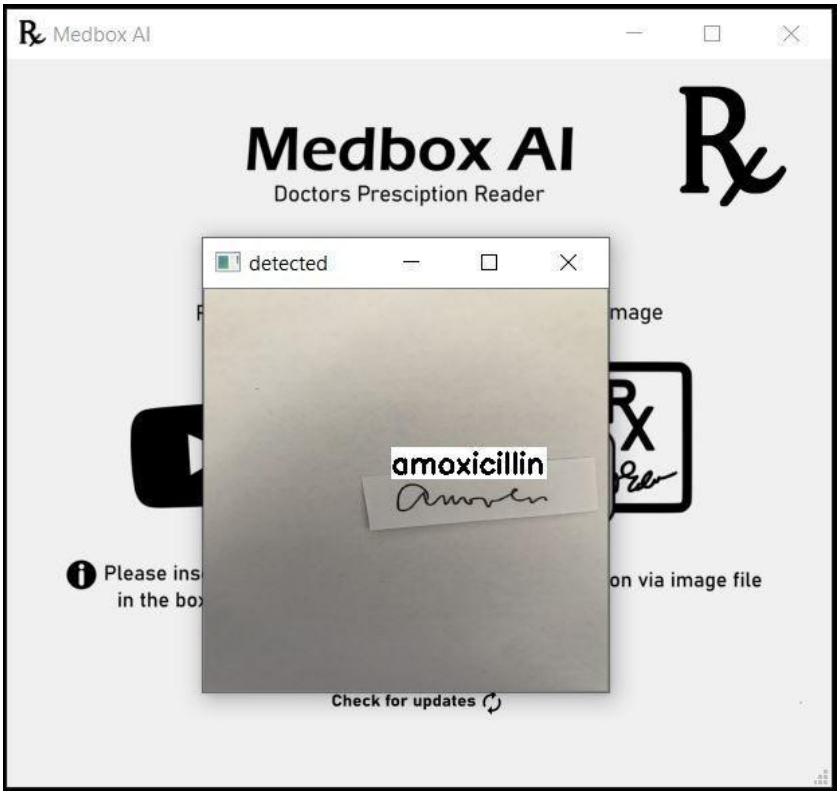


Figure 5.10 Medbox-Ai Run Interface

.

2. **Import Image** - When the import image is pressed, the system will prompt the user to upload the prescription image into the system, then the algorithm would translate the medicine name in the prescription.

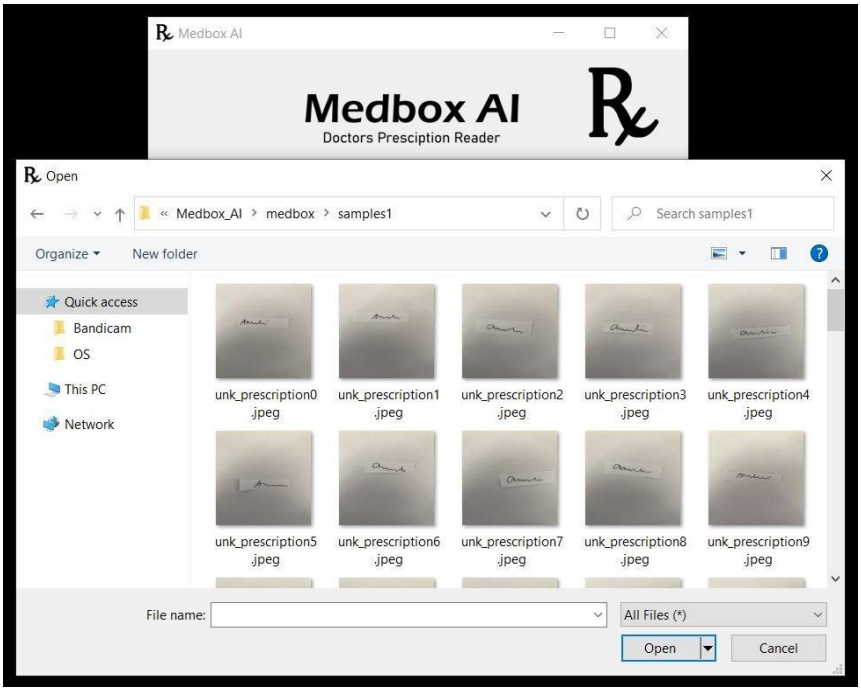


Figure 5.11 Medbox Import Interface

3. **Check for an update** - This button would make the system check for updates in Github's repository. If there are new updates, the system will update, which requires an internet connection to function.

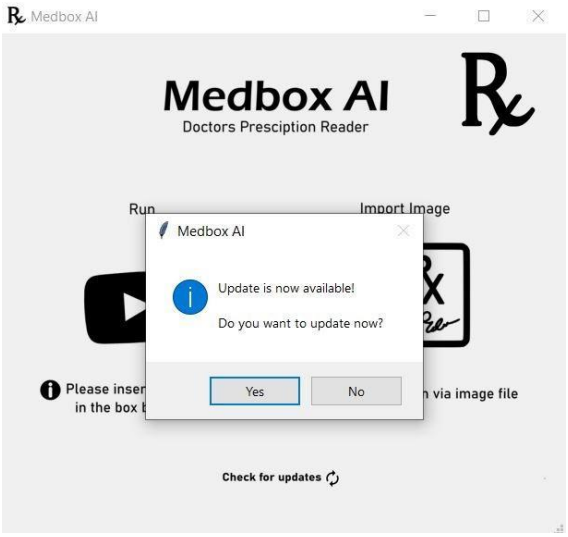


Figure 5.12 Notification Update Interface

Figure 5.13 is an interface of the MedBox-AI that lets the user notify the system's available update.

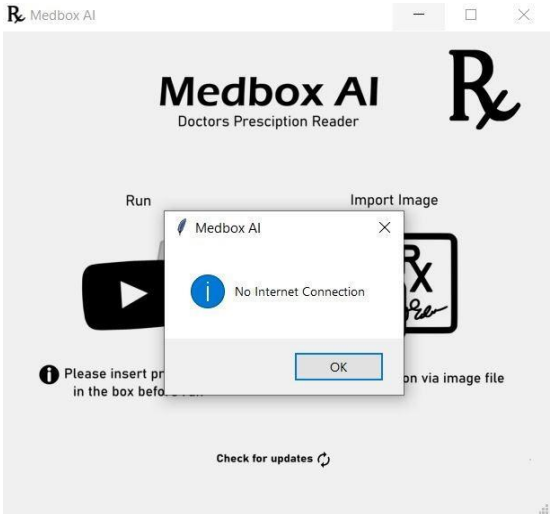


Figure 5.13 No Internet Connection Interface

Figure 5.13 is an interface to the MedBox-AI interface that indicates that the system is not linked to the Internet.

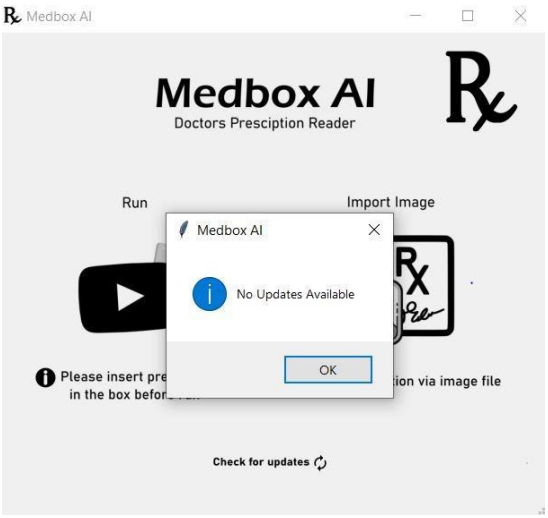


Figure 5.14 No Updates Available Interface

Figure 5.14 is an interface to the MedBox-AI interface that indicates that there is no update available.

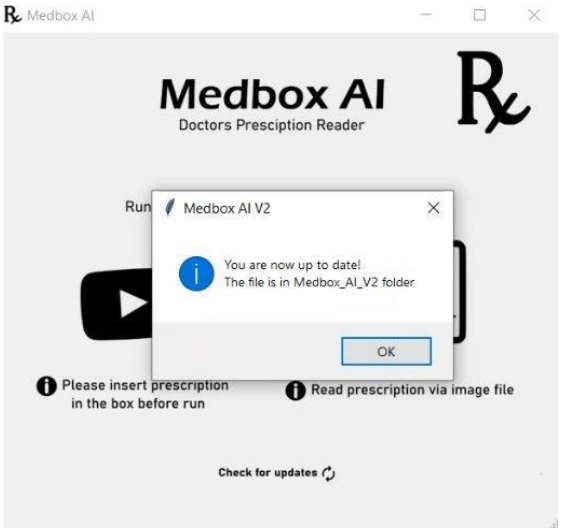


Figure 5.15 Up To Date Interface

Figure 5.15 is an interface to the MedBox-AI interface that shows the user that the device is up to date.

Testing Of The System

Table 5.1: Testing Of Functions From The User View

Functions	Expected Output	Actual Output	Status
Run Image	• Display Output	• Display Output	Successful
Import Image	• Import-file window • Display Output	• Import-file window • Display Output	Successful
Check for an update	• Display Notification Update • Display No Internet Connection • Display No Updates Available • Display Up to Date	• Display Notification Update • Display No Internet Connection • Display No Updates Available • Display Up to Date	Successful
Exit	• Exit System	• Exit System	Successful

Table 5.1 shows the testing of functions of the system with the expected output and the actual output. If all things worked out properly, then the expected output would be the same as the actual output.

Table 5.2: Testing Of Translating The Prescriptions

No. of Testing	Name of Medicine	Expected Output	Actual Output
01	**Amoxicillin**	**Successful**	**Successful**
02	**Amoxicillin**	**Successful**	**Successful**
03	**Amoxicillin**	**Successful**	**Successful**
04	**Amoxicillin**	**Successful**	**Failed**
05	**Amoxicillin**	**Successful**	**Successful**
06	**Cephalexin**	**Successful**	**Successful**
07	**Cephalexin**	**Successful**	**Successful**
08	**Cephalexin**	**Successful**	**Successful**
09	**Cephalexin**	**Successful**	**Successful**
10	**Cephalexin**	**Successful**	**Successful**
11	**Ciprofloxacin**	**Successful**	**Successful**
12	**Ciprofloxacin**	**Successful**	**Successful**

13	**Ciprofloxacin**	**Successful**	**Successful**
14	**Ciprofloxacin**	**Successful**	**Successful**
15	**Ciprofloxacin**	**Successful**	**Successful**
16	**Clindamycin**	**Successful**	**Failed**
17	**Clindamycin**	**Successful**	**Successful**
18	**Clindamycin**	**Successful**	**Successful**
19	**Clindamycin**	**Successful**	**Successful**
20	**Clindamycin**	**Successful**	**Successful**
21	**Doxycycline**	**Successful**	**Failed**
22	**Doxycycline**	**Successful**	**Successful**
23	**Doxycycline**	**Successful**	**Successful**
24	**Doxycycline**	**Successful**	**Successful**
25	**Doxycycline**	**Successful**	**Successful**

Table 5.2 shows the testing done to the system. Out of 25 tests for detection, 3 detections were not successful in the detection of the medicine. This means that there are times that the algorithm of the system was not able to detect the medicine used for the testing.

$$\textbf{Accuracy} = 100\% - \left(\frac{|Expected\ Output - Actual\ Output|}{Expected\ Output} \times 100\right)\%$$

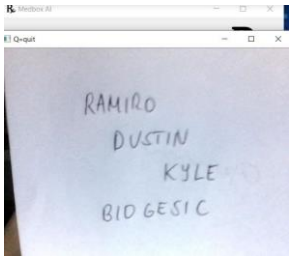
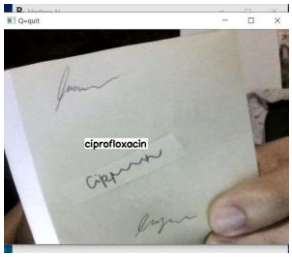
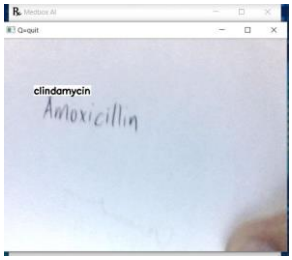
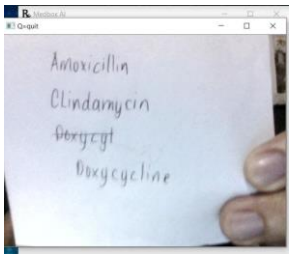
(equation 7)

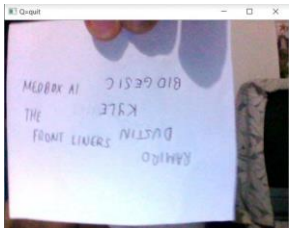
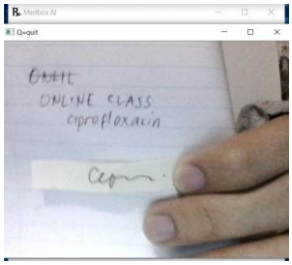
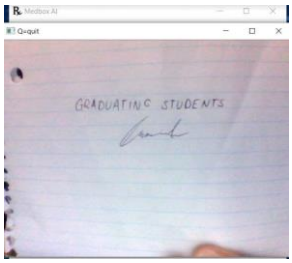
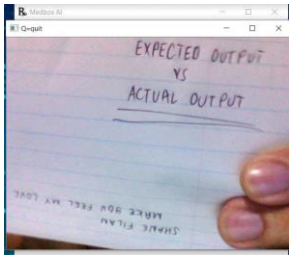
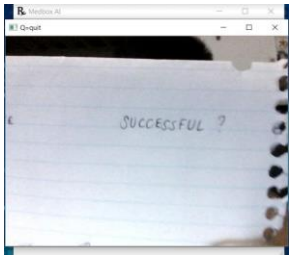
$$\textbf{Accuracy of the System} = 100 - \left(\frac{|25-23|}{25} \times 100\right)\%$$

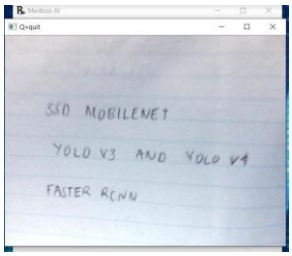
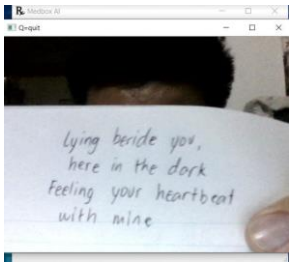
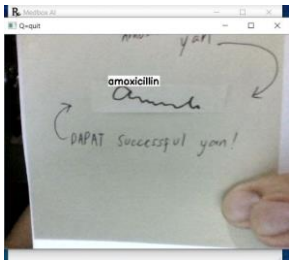
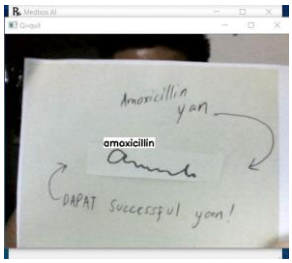
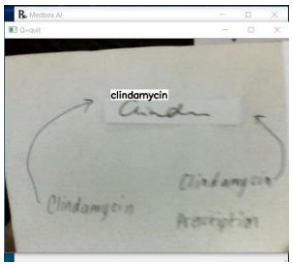
Accuracy of the system = 100 - 8%

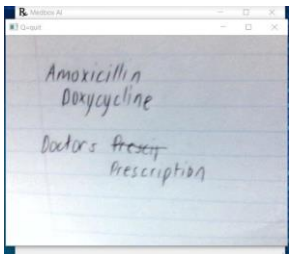
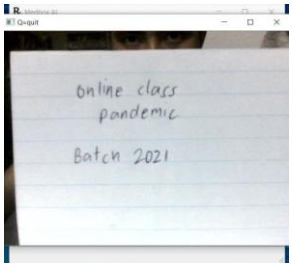
Accuracy of the system = 92%

Table 5.3: Testing Of Consistency Of The System

No. of Testing	Test Input	Expected Output	Actual Output
01		Successful	Successful
02		Successful	Failed
03		Successful	Successful
04		Successful	Failed
05		Successful	Successful

06		**Successful**	**Successful**
07		**Successful**	**Failed**
08		**Successful**	**Successful**
09		**Successful**	**Successful**
10		**Successful**	**Successful**
11		**Successful**	**Successful**

12		**Successful**	**Successful**
13		**Successful**	**Successful**
14		**Successful**	**Successful**
15		**Successful**	**Successful**
16		**Successful**	**Successful**
17		**Successful**	**Successful**

18		**Successful**	**Successful**
19		**Successful**	**Successful**
20		**Successful**	**Successful**

This table shows the consistency of the system. The input in this test is a mix of the correct input and the incorrect input. If the system were able to display the output with the correct input and an incorrect input then without an output, then the test for that is a success. But if the system were able to output something with an incorrect input, then that means the system failed that test.

$$\textbf{Accuracy} = 100\% - \left(\frac{|Expected\ Output - Actual\ Output|}{Expected\ Output} \times 100\right)\%$$

(equation 7)

$$\textbf{Accuracy of the System} = 100 - \left(\frac{|20-17|}{20} \times 100\right)\%$$

Accuracy of the system = 100 - 15%

Accuracy of the system = 85%

PROTOTYPE

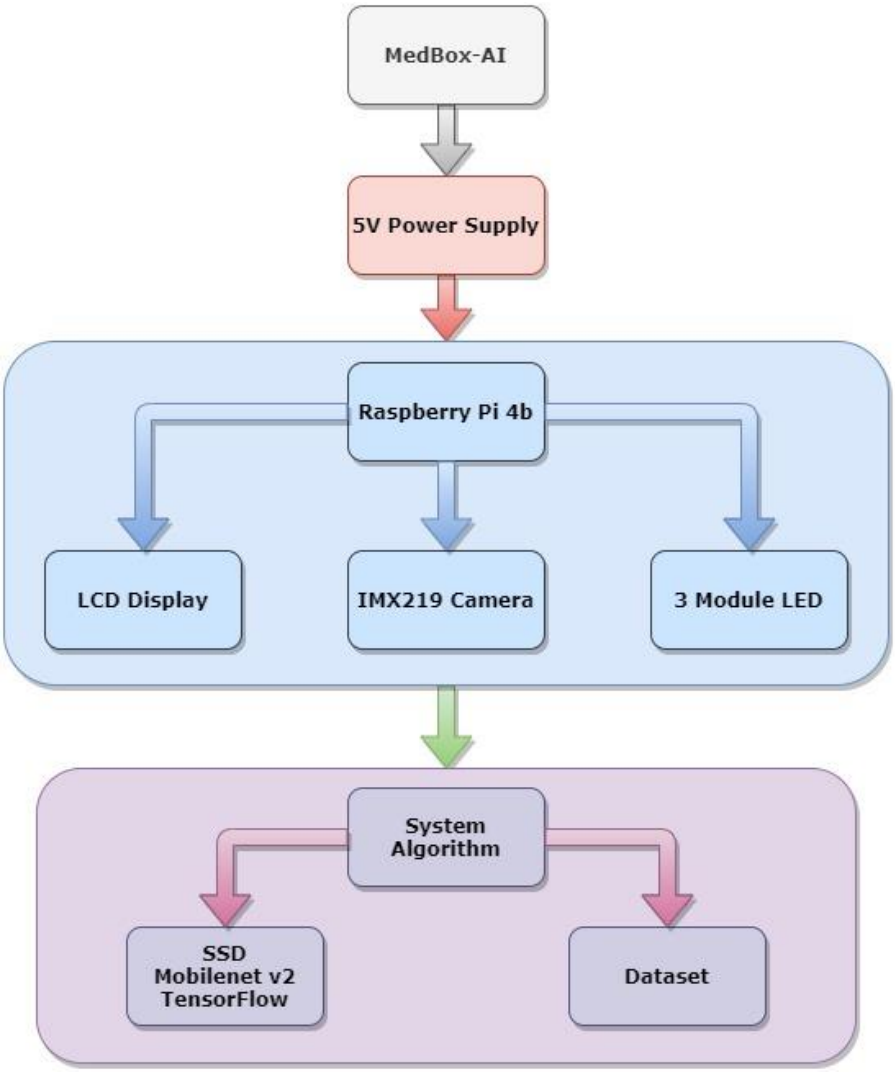


Figure 5.16 Final Design Diagram

Figure 5.16 represents the final system design. Based on the trade-offs and constraints, this particular system design was assessed as the best design. The whole system was composed of three categories: Power Management, Device Components, and System Algorithm.

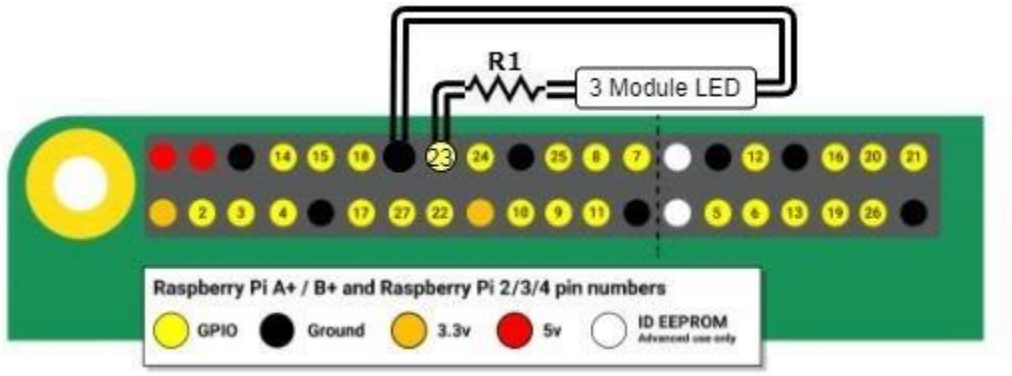


Figure 5.17 Schematic Diagram

Figure 5.17 shows the connections between the system's different components into the central processor, Raspberry Pi 4. The pin used for the power source is the GPIO-23 pin 16 was used for the LED's connection.

Components

Figure 5.18 Raspberry Pi 4 Model B 8 Gb.

The Raspberry Pi 4 Model B is the latest version of the low-cost Raspberry Pi computer. It has different RAM sizes, 4 GB and 8 GB. It is like a mini-computer that can run some applications. [25]

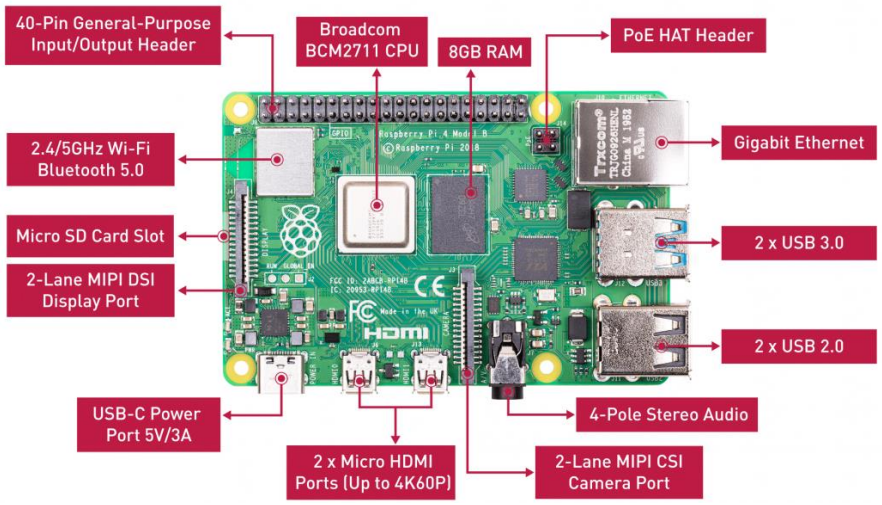


Figure 5.19 Raspberry Pi 4 Model B Parts

Table 5.4 Raspberry Pi 4 Model B Specification

SPECIFICATION	
Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	8GB
Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless
	LAN, Bluetooth 5.0, BLE
	Gigabit Ethernet
	2 × USB 3.0 ports
	2 × USB 2.0 ports
GPIO	Standard 40-pin GPIO header
Video & Sound	2 × micro HDMI ports (up to 4Kp60 supported)

	2-lane MIPI DSI display port
	2-lane MIPI CSI camera port
	4-pole stereo audio and composite video port
Multimedia	H.265 (4Kp60 decode)
	H.264 (1080p60 decode, 1080p30 encode)
	OpenGL ES, 3.0 graphics
SD card support	Micro SD card slot for loading operating system and data storage
Input power	5V DC via USB-C connector (minimum 3A)
	5V DC via GPIO header (minimum 3A)
	Power over Ethernet (PoE)–enabled(requires separate PoE HAT)
Environment	Operating temperature 0–50ºC

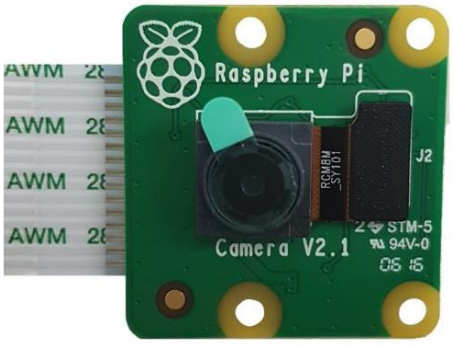


Figure 5.20 Raspberry Pi Camera Module V2 8 Megapixels

This camera module would be used for the image capturing device of the winning design options is the Sony IMX219. The IMX219 is a camera designed by the Raspberry Pi family, and unlike the usual raspberry pi camera module that only has 5 megapixels resolution, this camera has 8 megapixels resolution. It can also process 3280 x 2464 pixel static images and supports 1080p@30fps, 720p@60fps, and 640x480p@90fps video. It attaches to Pi by the dedicated standard CSI interface. It is supplementary for Raspberry Pi's official camera to fulfill the demands for different lens mounts, the field of view (FOV), depth of the field (DOF), and the motorized IR, cut filter for both daylight and night vision. [27]

Figure 5.21 Sd Card

Figure 5.21 is a Class 10 UHS 1 microSD card that is one of the best options if you are looking for fast application performance. The 32 GB microSD card features a maximum transfer speed of up to 98 MBps. This SD card would act as the Rom of both of the microprocessors.

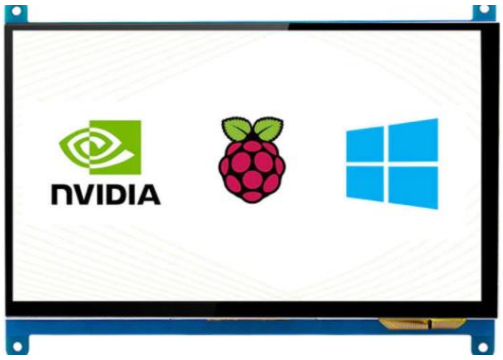


Figure 5.22 Touchscreen Lcd

Figure 5.22 is a 7-inch touchscreen LCD that serves as the display and the user's primary way to interact with the platform. This LCD would utilize both HDMI and USB for the display and touch interface. It has a display resolution of 800 x 480 and can be set up to 1024 x 600. It has a product dimension of 165mm x 125mm x 15mm.

Figure 5.23 Led Strip

Figure 5.23 is the LED Module that would serve as the lighting for the platform's tray, serving as a guide. The led strip has a wattage of 0.72 watts, has an input voltage of DC 12v, and has a 77 x 12 x 6 mm dimension.

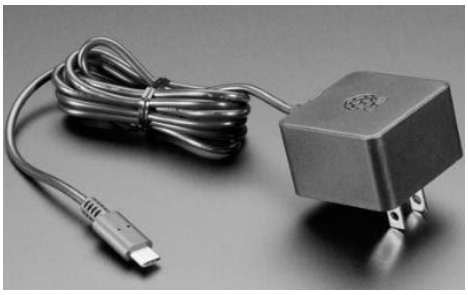


Figure 5.24 Raspi Power Adapter

Figure 5.24 is the Raspberry Pi Power Adapter. It would be used to power the raspberry pi with its micro USB port used for power connection. It has an input capacity of AC 100 – 240V, 50 / 60Hz, and an output capacity of DC 12V, Max 5A, 60W. It can supply with all amperage less than 5A. If the device draws out the only 3A, then the power adapter would only supply 3A, but note that if the device draw exceeded 5A, then the power adapter will be damaged soon.

Figure 5.25 Casing (Abs Plastic Filament)

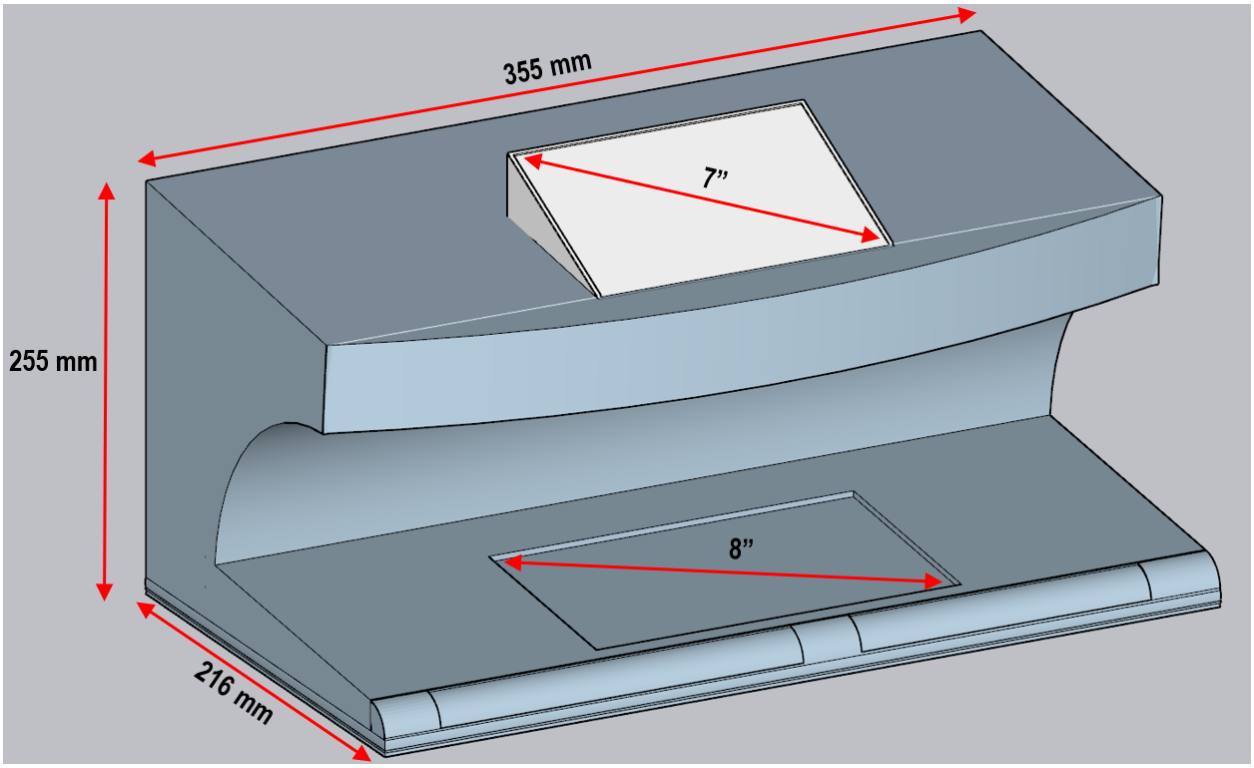
Figure 5.25 is the Casing (ABS plastic filament) that would be used for 3d printers. It would serve as the casing of the platform. This ABS plastic filament has an operating temperature of 210-240 °C and has a diameter of 1.75mm.

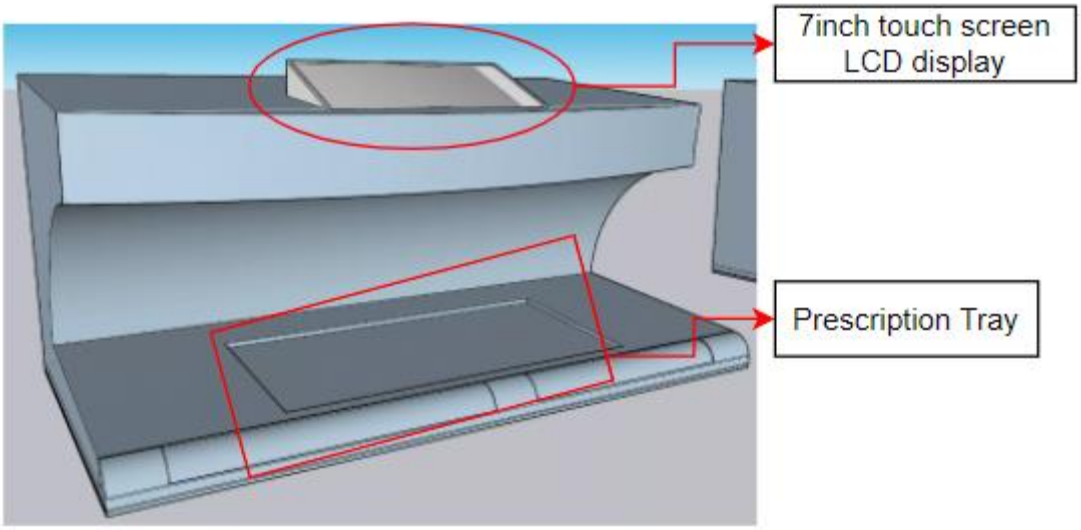
Table 5.5 Bill Of Materials

Bill of Materials		
Components	**Device**	**Cost**
Processor	Raspberry Pi 4	4,749.75 Php
Memory	SD card	269.00 Php
Display	LCD Touch Screen	3,000.00 Php
Power Supply	Raspberry Pi Power Supply	574.75 Php
Camera	IMX219	1,650.00 Php
Lighting	3 module led light	9.00 Php
Casing	ABS plastic filament	1,000.00 Php
Total Cost		**11,252.50 Php**

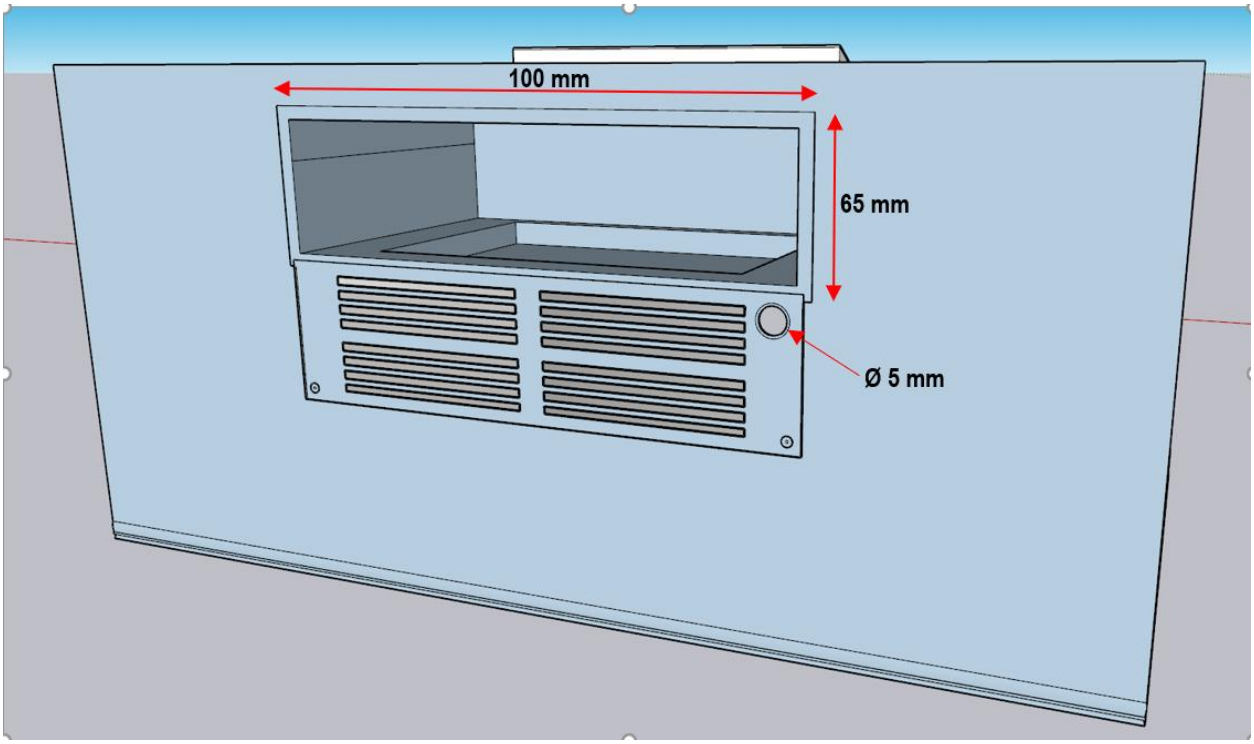
This table shows the components and cost when using Raspberry Pi 4 B as the processor. The total cost would be 11,252.50 PHP. Some of these prices may be subject to change.

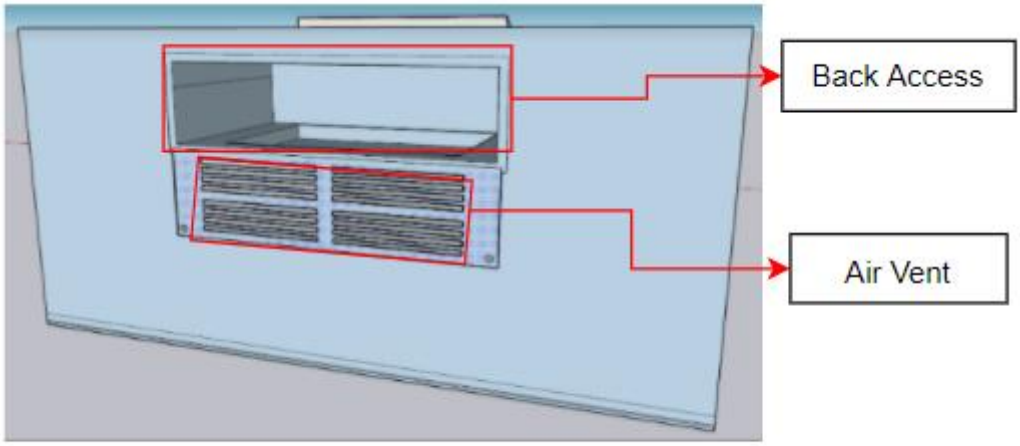
Prototype Layout





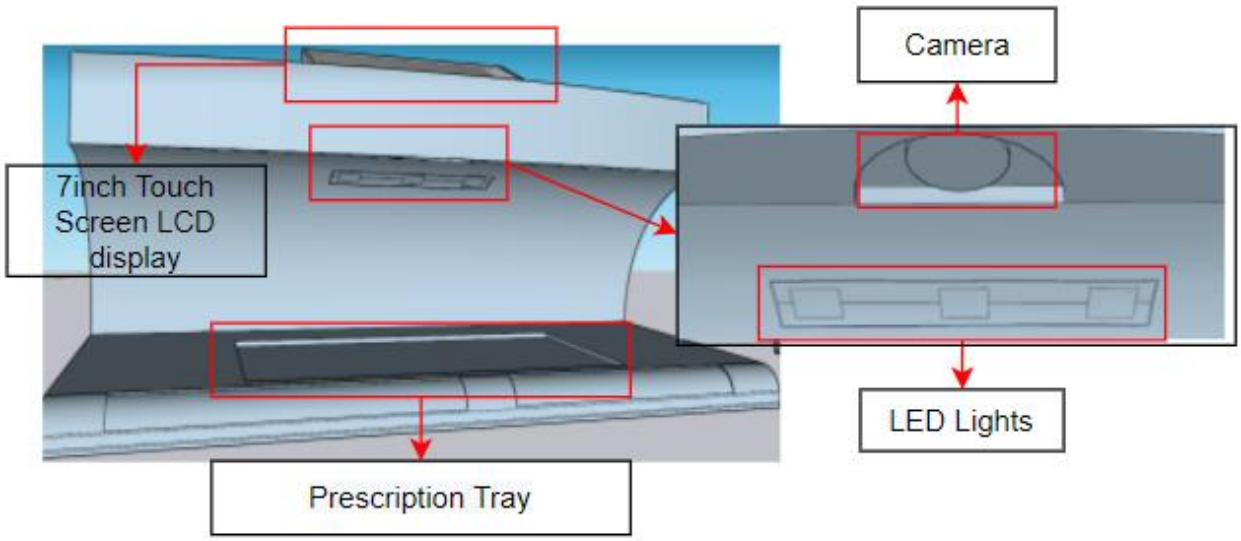
- ***LCD Display*** - This is a touch screen display where users can interact with the device.
- ***Prescription Tray*** - This is where the prescription will be placed.





Back View

- ***Back Access*** - Where you can access the Raspberry Pi 4 and its connection.
- ***Air Vent*** - This will let hot air coming to the processor go out.



- ***LCD Display*** - This is a touch screen display where users can interact with the machine.
- ***Prescription Tray*** - This is where the prescription will be placed.
- ***LED light*** - This will provide light to brighten up the image that the camera will capture.
- ***Camera*** - This will capture the prescription placed in the prescription tray and feed it to the algorithm.

Summary Of Findings:

Medbox AI was specifically designed to help pharmacists in translating the medicine names that are present on the prescription. The project started with 3 design options. The 3 design options include YOLOV4, SSD Mobilenet, and finally Faster RCNN. Each of these design options was tested 10 times and each of them has varying results. (*please refer to chapter 3 for a detailed explanation of each design option)*

And based on the trade-off analysis of the different design options, design option 1, the design option with the SSD Mobilenet algorithm, came out on top. So the proponents did further testing to the winning design. Additional 25 tests were done to check the consistency of the accuracy of the system. And the results of the test were promising with the average accuracy of the system being at 92%. After testing the consistency of the accuracy, the proponents tested the system's ability to detect errors. This means that if the input has no input, then the system must have no output. To test for this scenario, the proponents added 15 more tests dedicated to finding out if the system were able to have a correct output. And the result of that testing is that the system has 85% of having a correct output. And that the system has a 15% error where either the system has a correct input but wrong output or an incorrect input that has an output.

Assessment Of The Attainment Of The Project Objectives

- Project objective # 1:Provide a platform that helps the pharmacy to translate the sloppily written prescriptions. The proponents were able to develop a platform that would translate the sloppy handwriting present in the prescription. The proponents were able to train and use an algorithm that resulted in the translation of the sloppy handwriting that is present in the prescription correctly. [7]
- Project objective # 2: Decrease the turnaround time in verifying the context of the prescriptions. The proponents were able to accomplish this objective by using the algorithm with a low time consumption while not compromising the accuracy of the system.

- Project objective # 3: Decrease the probability of incorrect interpretation of the doctor's prescription by the pharmacist. The proponents were able to accomplish this objective because the system has a high percentage of having the correct output depending on the input.

Conclusion

Based on the testing done to the system, the system has an accuracy of 92% when it comes to the first testing. The first testing was focused more on having the correct input. It means that the input that was fed into the system was present from the 5 medicine names that were in the dataset. While the second test focused more on the incorrect input or input that was either not on the dataset or not related to the training of the system at all. In this regard, the system has an accuracy of 85% which means that if the system were fed incorrect or improper input, then the system would have no detection as an output. This was done because the current limitation of the platform would make it exposed to the inputs that were not meant to be translated. And because of these results, the proponents can say that they were able to achieve all of the project objectives and the client's requirements.

The proponents achieved the first project objective because they were able to create a system that can help the pharmacist translate the sloppily written prescriptions properly with high average accuracy. The second objective was achieved because the system was able to reduce the turnaround time for the translation of sloppily written prescriptions and that the system was far from the usual time it takes for the pharmacist to translate the prescriptions properly. And finally, the last project objective was achieved because the system has a high accuracy even though the input that was fed on the system was incorrect. It was able to know when to produce output depending on the input and that it has a low percentage of actually having a wrong output.

And when it comes to the client's requirements, the proponents were able to successfully achieve their requirements. Because the system was able to determine the medicine name on the prescription with the correct translation, display the prescribed medicine accurately, and has a very low time consumption when it comes to the manual translation done by the pharmacists. And finally, when it comes to the costs, the total costs of the components of the platform were at 11,252.50 PHP which is way below the limit set by the client which was at 15,000 PHP.

So the proponents can say that the project objectives and the client's requirements were successfully achieved using design option 1 as the final system.

RECOMMENDATION

The proponents provided a list of recommendations that would further help improve the design project.

- The additional sample prescription for the dataset. It is because the dataset of the platform is limited to 5 antibiotic names only. A further increase in the size of the dataset would greatly expand the platform's scope from medicine names to full pledge written prescription translation.
- The testing should be done on an actual Raspberry Pi 4 model b to test the system’s performance accurately.
- Designing the Medbox AI would further improve the image quality, which may help improve the system's accuracy.
- Find a way to connect the platform to an inventory system of the pharmacy.
- Further, improve the system by adding the feature to help identify if the prescription is still valid and legitimate.

APPENDICES

APPENDIX A. TESTING

A. Testing of SSD mobilnetV2 Algorithm

This figure shows the testing done for each of the medicines using the SSD Mobilenet algorithm.

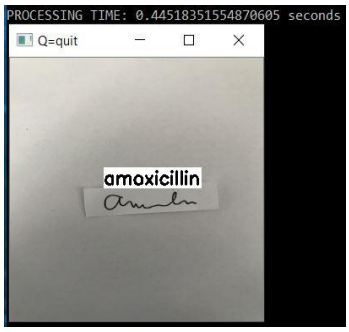
B. Testing YOLOv4 Darknet Algorithm

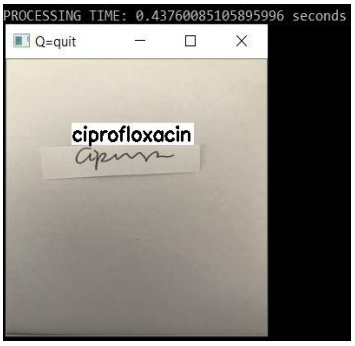
This figure shows the testing done for each of the medicines using YOLOv4 algorithm.

C. Testing Faster RCNN Inception v2 Algorithm

This figure shows the result of the testing of each medicine using Faster - RCNN Algorithm.

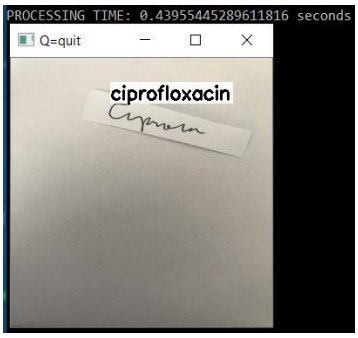
D. Testing of Final Design Option

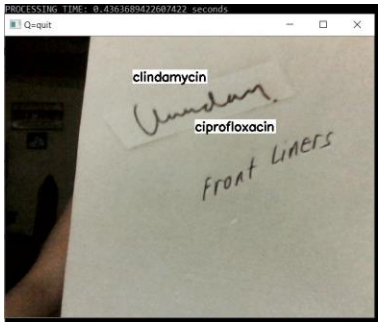





PROCESSING TIME: 0.4526674747467041 seconds
Q=quit
ciprofloxacin


PROCESSING TIME: 0.4516727924346924 seconds
Q=quit
ciprofloxacin

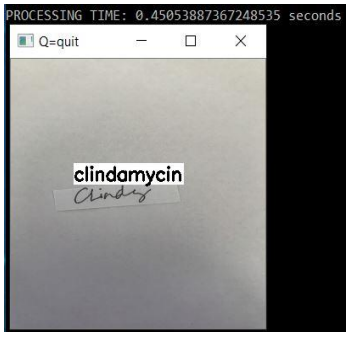

PROCESSING TIME: 0.43955445289611816 seconds
Q=quit
ciprofloxacin

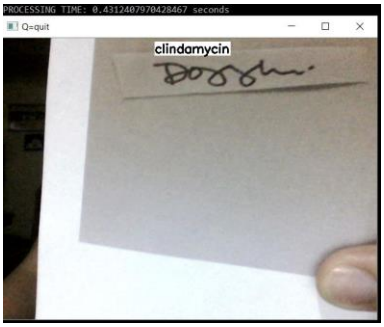

Q=quit
clindamycin
ciprofloxacin
Front Liners


Q=quit
clindamycin

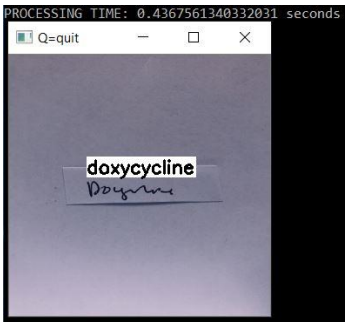

Q=quit
clindamycin

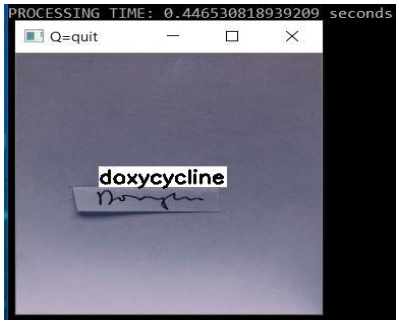

PROCESSING TIME: 0.44596433639526367 seconds
Q=quit
clindamycin

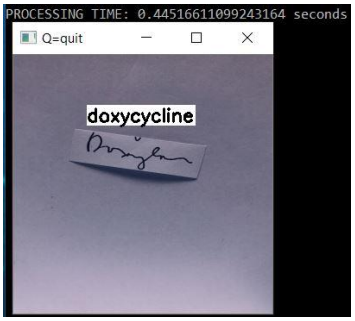

PROCESSING TIME: 0.45053887367248535 seconds
Q=quit
clindamycin

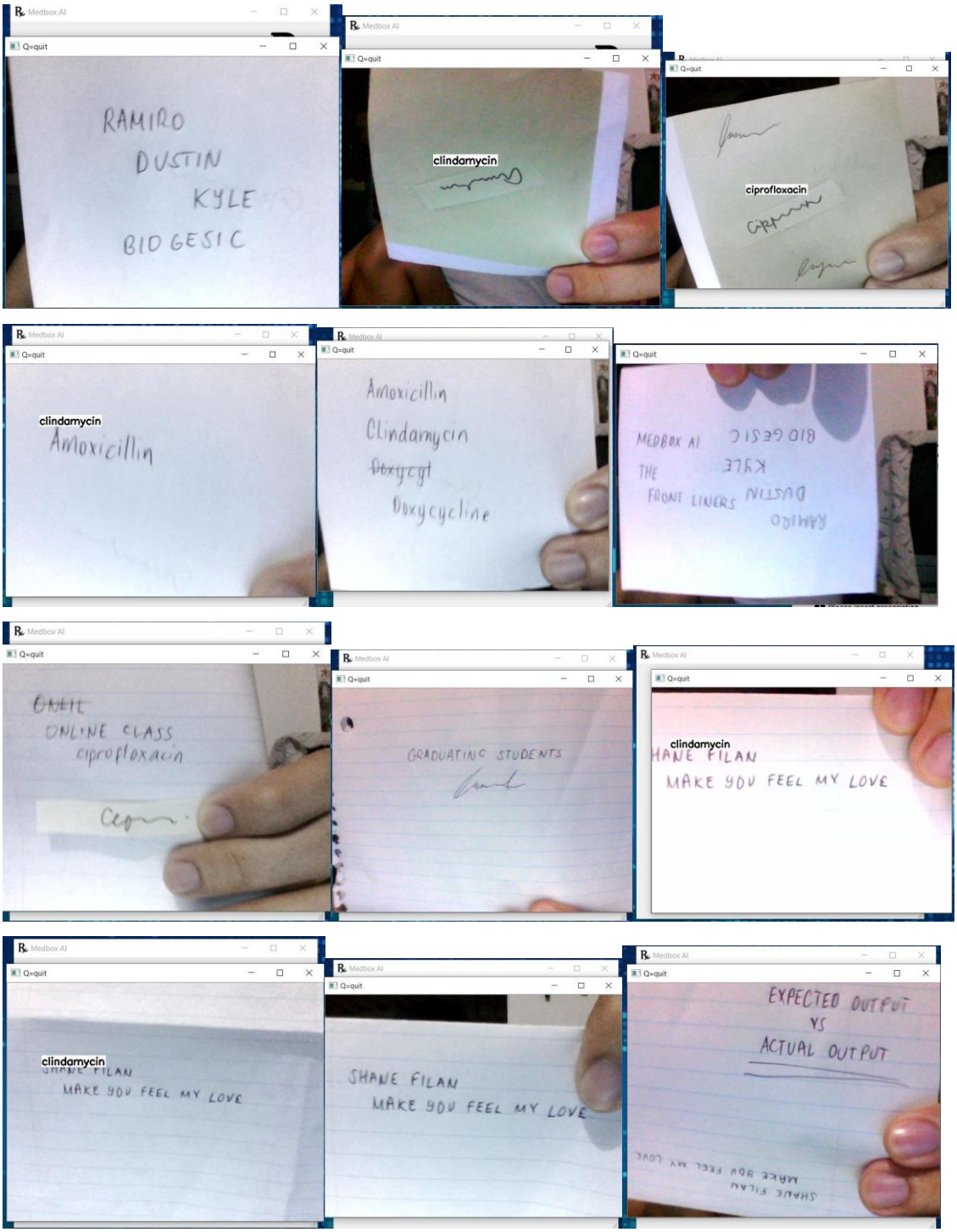

Q=quit
clindamycin

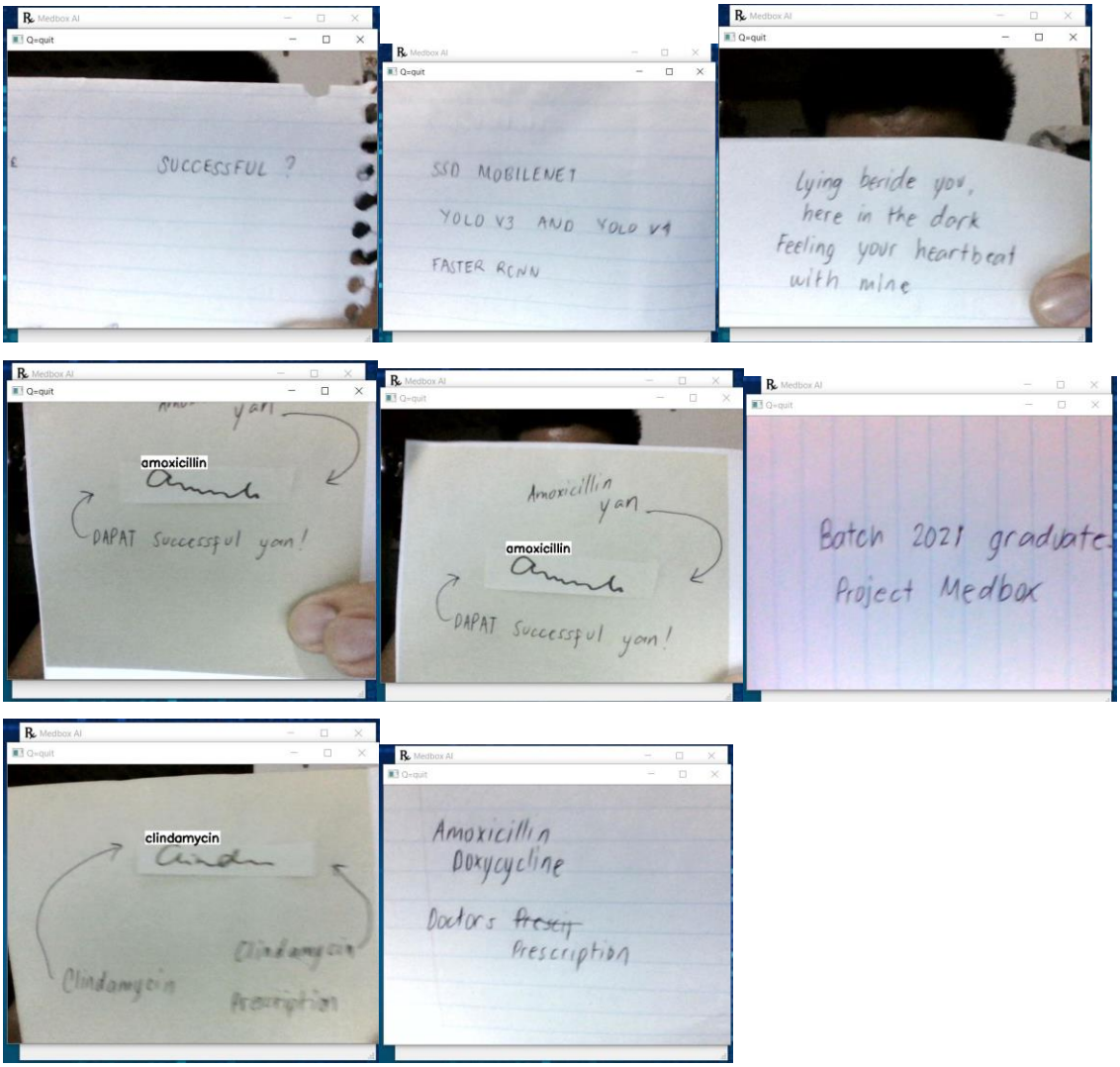

doxycycline


PROCESSING TIME: 0.4367561340332031 seconds
Q=quit
doxycycline


PROCESSING TIME: 0.446530818939209 seconds
Q=quit
doxycycline


PROCESSING TIME: 0.44516611099243164 seconds
Q=quit
doxycycline


Medbox AI
Q=quit
RAMIRO
DUSTIN
KYLE
BIOGESIC
clindamycin
ciprofloxacin
Amoxicillin
Clindamycin
Doxycycline
ONLINE CLASS
ciprofloxacin
GRADUATING STUDENTS
clindamycin
SHANE FILAN
MAKE YOU FEEL MY LOVE
EXPECTED OUTPUT
VS
ACTUAL OUTPUT



This figure shows the result of the testing of each medicine using the Final design.

APPENDIX B. GANTT CHART

	Activities	TIMESCALE (Weeks)											
		1	2	3	4	5	6	7	8	9	10	11	12
1	**Project Planning**												
1.1	Project Goal												
1.2	Research												
1.3	Identifying Potential Costumer												
1.4	Interview												
1.5	Online Survey												
1.6	Plan Review												
1.7	Requirement Analysis												
2	**Project Design**												
2.1	Design Options												
2.2	Graphic User Interface												
3	**Implementation**												
3.1	Front End Development												
3.2	Back End Development												
3.3	Finalizing												
4	**System Testing**												
4.1	Testing												
4.2	Launching												

This chart shows the timescale of the progress of the project.

APPENDIX C. 5 WHYS

- Why did the medical error happen?

  - because the patient received a wrong medicine

- Why did the patient receive the wrong medicine?

  - because the pharmacist dispenses the wrong medicine

- Why did the pharmacist dispense the wrong medicine?

  - because the pharmacist got confused releasing the medication pills

- Why did the pharmacist get confused, releasing the medication pills?

  - because of the medical prescription the pharmacist received.

- Why the medical prescription the pharmacist received?

  - because the prescription of the patient is unreadable

MedBox-AI: An AI-Powered Medicine Scanner using Deep Learning-Based Translation of Doctor's Prescription

User's Manual

General Description

MedBox-AI: An AI-Powered Medicine Scanner using Deep Learning-Based Translation of Doctor's Prescription is a device that can detect and translate the generic name of medicine prescribed by a doctor in the prescription.

Accessories

- MedBox-AI Device
- 12V AC Adapter
- User's Manual

Guide Step

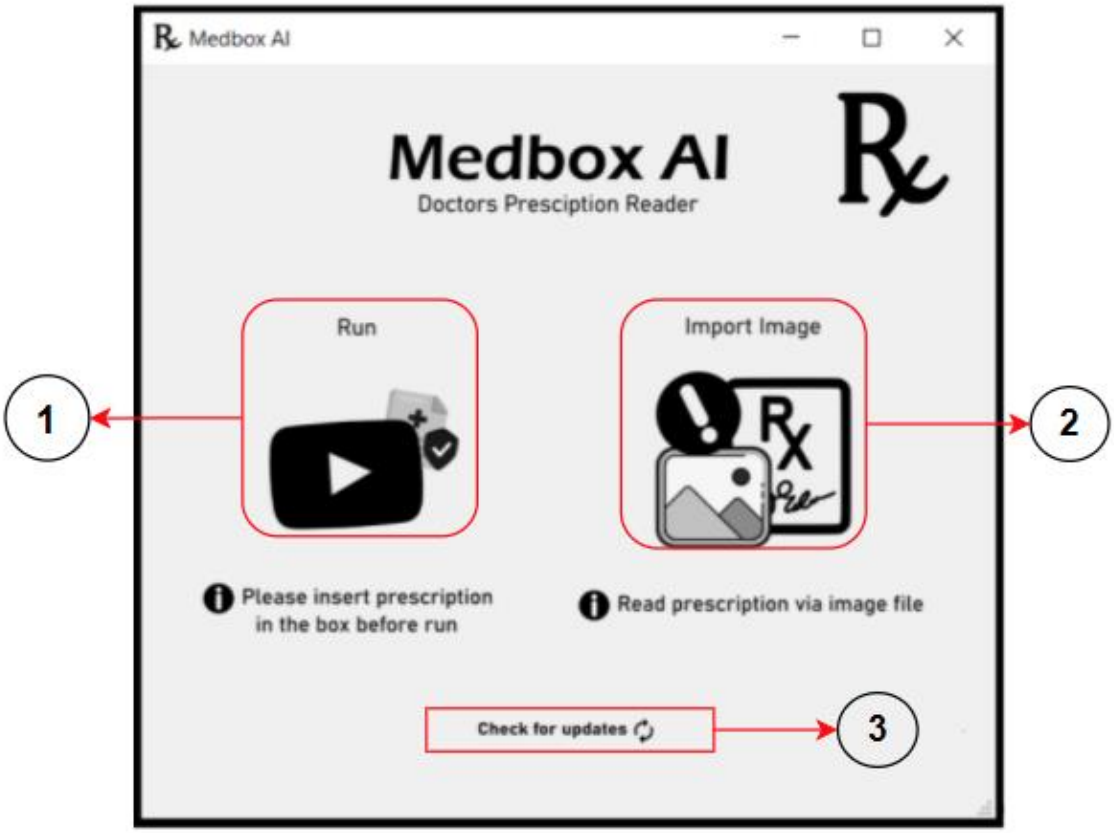
Step 1: Plug MedBox-AI Device

Plug the device to power on and wait to boot up.

Step 2: Open MedBox-AI Application

Open the application installed on the device.

GUI & Functions



No.	GUI	Function
1	Run Button	To display output of the inputted prescription
2	Import Image Button	To display output of the uploaded image of the prescription
3	Update Button	To update the version of the tflite file and graphical user interface

Step 3: Input Prescription or Upload Prescription Image

Select in the GUI application either run or import image.

Step 4: Wait to display output

Results will be displayed on the device screen once you select a button in the GUI application.

Step 5: Check for updates

Click the Update Button to check some updated versions of the weights file of the GUI.

APPENDIX E. SOURCE CODE

```
import os
import argparse
import cv2
import numpy as np
import sys
import glob
import http.client
import importlib.util
import time
import git
import urllib
import requests
from urllib.request import urlopen
from threading import Thread
import threading
from PyQt5.QtWidgets import QInputDialog, QFileDialog, QWidget
from PyQt5 import QtCore, QtGui, QtWidgets
global str
import pandas as pd
from tkinter import *
from tkinter import messagebox


class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(614, 544)
        icon = QtGui.QIcon()
        icon.addPixmap(QtGui.QPixmap("medbox/art/prescript.ico"),          QtGui.QIcon.Normal,
QtGui.QIcon.Off)
        MainWindow.setWindowIcon(icon)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
```

```
        self.centralwidget.setObjectName("centralwidget")
        self.lbl_Title = QtWidgets.QLabel(self.centralwidget)
        self.lbl_Title.setGeometry(QtCore.QRect(10, 10, 581, 141))
        self.lbl_Title.setFrameShape(QtWidgets.QFrame.NoFrame)
        self.lbl_Title.setText("")
        self.lbl_Title.setPixmap(QtGui.QPixmap("medbox/art/medbox.ico"))
        self.lbl_Title.setScaledContents(False)
        self.lbl_Title.setAlignment(QtCore.Qt.AlignCenter)
        self.lbl_Title.setObjectName("lbl_Title")
        self.lbl_Run = QtWidgets.QLabel(self.centralwidget)
        self.lbl_Run.setGeometry(QtCore.QRect(50, 180, 211, 31))
        self.lbl_Run.setFrameShape(QtWidgets.QFrame.NoFrame)
        self.lbl_Run.setText("")
        self.lbl_Run.setPixmap(QtGui.QPixmap("medbox/art/run.ico"))
        self.lbl_Run.setAlignment(QtCore.Qt.AlignCenter)
        self.lbl_Run.setObjectName("lbl_Run")
        self.btn_Run = QtWidgets.QPushButton(self.centralwidget)
        self.btn_Run.setGeometry(QtCore.QRect(80, 220, 161, 131))
        self.btn_Run.setText("")
        icon1 = QtGui.QIcon()
        icon1.addPixmap(QtGui.QPixmap("medbox/art/play.ico"), QtGui.QIcon.Normal,
QtGui.QIcon.Off)
        self.btn_Run.setIcon(icon1)
        self.btn_Run.setIconSize(QtCore.QSize(150, 150))
        self.btn_Run.setFlat(True)
        self.btn_Run.setObjectName("btn_Run")
        self.lbl_ImpImg = QtWidgets.QLabel(self.centralwidget)
        self.lbl_ImpImg.setGeometry(QtCore.QRect(360, 170, 161, 41))
        self.lbl_ImpImg.setFrameShape(QtWidgets.QFrame.NoFrame)
        self.lbl_ImpImg.setText("")
        self.lbl_ImpImg.setPixmap(QtGui.QPixmap("medbox/art/import.ico"))
        self.lbl_ImpImg.setAlignment(QtCore.Qt.AlignCenter)
```

```
        self.lbl_ImpImg.setObjectName("lbl_ImpImg")
        self.btn_Image = QtWidgets.QPushButton(self.centralwidget)
        self.btn_Image.setGeometry(QtCore.QRect(360, 220, 161, 131))
        self.btn_Image.setText("")
        icon2 = QtGui.QIcon()
        icon2.addPixmap(QtGui.QPixmap("medbox/art/image.ico"), QtGui.QIcon.Normal, QtGui.QIcon.Off)
        self.btn_Image.setIcon(icon2)
        self.btn_Image.setIconSize(QtCore.QSize(150, 150))
        self.btn_Image.setFlat(True)
        self.btn_Image.setObjectName("btn_Image")
        self.img_Rx = QtWidgets.QLabel(self.centralwidget)
        self.img_Rx.setGeometry(QtCore.QRect(500, 20, 81, 91))
        self.img_Rx.setText("")
        self.img_Rx.setPixmap(QtGui.QPixmap("medbox/art/prescript.ico"))
        self.img_Rx.setScaledContents(True)
        self.img_Rx.setObjectName("img_Rx")
        self.lbl_aboutImg = QtWidgets.QLabel(self.centralwidget)
        self.lbl_aboutImg.setGeometry(QtCore.QRect(310, 370, 261, 41))
        self.lbl_aboutImg.setText("")
        self.lbl_aboutImg.setPixmap(QtGui.QPixmap("medbox/art/about_img.ico"))
        self.lbl_aboutImg.setObjectName("lbl_aboutImg")
        self.lbl_aboutRun1 = QtWidgets.QLabel(self.centralwidget)
        self.lbl_aboutRun1.setGeometry(QtCore.QRect(40, 370, 221, 31))
        self.lbl_aboutRun1.setText("")
        self.lbl_aboutRun1.setPixmap(QtGui.QPixmap("medbox/art/about_run1.ico"))
        self.lbl_aboutRun1.setObjectName("lbl_aboutRun1")
        self.lbl_aboutRun2 = QtWidgets.QLabel(self.centralwidget)
        self.lbl_aboutRun2.setGeometry(QtCore.QRect(50, 390, 191, 31))
        self.lbl_aboutRun2.setText("")
        self.lbl_aboutRun2.setPixmap(QtGui.QPixmap("medbox/art/about_run2.ico"))
        self.lbl_aboutRun2.setObjectName("lbl_aboutRun2")
```

```
        self.lbl_Check = QtWidgets.QLabel(self.centralwidget)
        self.lbl_Check.setGeometry(QtCore.QRect(240, 470, 111, 21))
        self.lbl_Check.setText("")
        self.lbl_Check.setPixmap(QtGui.QPixmap("medbox/art/check.ico"))
        self.lbl_Check.setScaledContents(True)
        self.lbl_Check.setAlignment(QtCore.Qt.AlignCenter)
        self.lbl_Check.setObjectName("lbl_Check")
        self.btn_Update = QtWidgets.QPushButton(self.centralwidget)
        self.btn_Update.setGeometry(QtCore.QRect(350, 470, 21, 21))
        self.btn_Update.setText("")
        icon3 = QtGui.QIcon()
        icon3.addPixmap(QtGui.QPixmap("medbox/art/update.ico"), QtGui.QIcon.Normal,
QtGui.QIcon.Off)
        self.btn_Update.setIcon(icon3)
        self.btn_Update.setShortcut("")
        self.btn_Update.setFlat(True)
        self.btn_Update.setObjectName("btn_Update")
        self.btn_Webcam = QtWidgets.QPushButton(self.centralwidget)
        self.btn_Webcam.setGeometry(QtCore.QRect(580, 470, 21, 21))
        self.btn_Webcam.setText("")
        icon4 = QtGui.QIcon()
        icon4.addPixmap(QtGui.QPixmap("medbox/art/webcam.ico"), QtGui.QIcon.Normal,
QtGui.QIcon.Off)
        self.btn_Webcam.setIcon(icon4)
        self.btn_Webcam.setFlat(True)
        self.btn_Webcam.setObjectName("btn_Webcam")
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 614, 26))
        self.menubar.setObjectName("menubar")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
```

```
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "Medbox AI"))

        self.btn_Image.clicked.connect(self.pushButton_Image)
        self.btn_Run.clicked.connect(self.pushButton_Run)
        self.btn_Update.clicked.connect(self.pushButton_Update)

#Update
    def pushButton_Update(self):
        # PLEASE WAIT
        window=Tk()
        lbl=Label(window, text='Please wait', fg='black', font=("Helvetica", 12))
        lbl.place(x=65, y=35)
        window.title('Medbox AI')
        window.geometry("400x100+10+10")
        window.after(2500, lambda: window.destroy())
        window.mainloop()

        def is_internet():
            try:
                urlopen('https://www.google.com', timeout=1)
                return True
            except urllib.error.URLError as Error:
                return False
```

```
    if is_internet():
        print('may net')
        url = 'https://github.com/mcabie3098/Medbox_AI'
        r = requests.get(url)
        print(r)
        page = r.status_code

        if page==200:
            # NO UPDATES YET
            window = Tk()
            window.eval('tk::PlaceWindow %s center' % window.winfo_toplevel())
            window.withdraw()
            messagebox.showinfo("Medbox AI", "No Updates Available")
            window.deiconify()
            window.destroy()
            window.quit()

        elif page==404:
            # UPDATE IS HERE!
            # NOTE: YOU HAVE TO RENAME GITHUB FOLDER EVERY TIME YOU UPDATE eg. Medbox_AIv2
            window = Tk()
            window.eval('tk::PlaceWindow %s center' % window.winfo_toplevel())
            window.withdraw()
            if messagebox.askyesno('Medbox AI', 'Update is now available! \n\nDo you want to update now?', icon='info') == True:
                # DOWNLOADING
                window.deiconify()
                window.destroy()
                window.quit()
                window=Tk()
```

```
            lbl=Label(window, text='Downloading...', fg='black', font=("Helvetica", 12))
            lbl.place(x=65, y=35)
            window.title('Medbox AI')
            window.geometry("400x100+10+10")
            window.after(2500, lambda: window.destroy())
            window.mainloop()
            # YOU ARE NOW UP TO DATE
            git.Git("").clone("https://github.com/mcabie3098/Medbox_AI_V2.git")
            window = Tk()
            window.eval('tk::PlaceWindow %s center' % window.winfo_toplevel())
            window.withdraw()
            messagebox.showinfo("Medbox AI V2", "You are now up to date! \nThe file is in Medbox_AI_V2 folder")
            window.deiconify()
            window.destroy()
            window.quit()
            sys.exit()

        else:
            print('no')
        window.deiconify()
        window.destroy()
        window.quit()

    else:
        #NO INTERNET
        window = Tk()
        window.eval('tk::PlaceWindow %s center' % window.winfo_toplevel())
        window.withdraw()
        messagebox.showinfo("Medbox AI", "No Internet Connection")
        window.deiconify()
        window.destroy()
```

```
        window.quit()


#Run
    def pushButton_Run(self):
        videoCaptureObject = cv2.VideoCapture(0, cv2.CAP_DSHOW)
        result = True
        while(result):
            ret,frame = videoCaptureObject.read()
            cv2.imwrite("image.jpg",frame)
            result = False
        cv2.destroyAllWindows()
        self.open_dialog_box3()


    def open_dialog_box3(self):
        img_elec = 'image.jpg'


        parser = argparse.ArgumentParser()
        parser.add_argument('--modeldir', help='Folder the .tflite file is located in',
                            default='medbox')
        parser.add_argument('--graph', help='Name of the .tflite file, if different than detect.tflite',
                            default='detect.tflite')
        parser.add_argument('--labels', help='Name of the labelmap file, if different than labelmap.txt',
                            default='labelmap.txt')
        parser.add_argument('--threshold', help='Minimum confidence threshold for displaying detected objects',
                            default=0.5)
        parser.add_argument('--image', help='Name of the single image to perform detection on. To run detection on multiple images, use --imagedir',
                            default=None)
        parser.add_argument('--imagedir', help='Name of the folder containing images to perform detection on. Folder must contain only images.',
                            default=None)
```

```
    parser.add_argument('--edgetpu', help='Use Coral Edge TPU Accelerator to speed up detection',
                        action='store_true')

    args = parser.parse_args()

    MODEL_NAME = args.modeldir
    GRAPH_NAME = args.graph
    LABELMAP_NAME = args.labels
    min_conf_threshold = float(args.threshold)
    use_TPU = args.edgetpu

    # Parse input image name and directory.
    IM_NAME = args.image
    IM_DIR = args.imagedir

    # If both an image AND a folder are specified, throw an error
    if (IM_NAME and IM_DIR):
        print('Error! Please only use the --image argument or the --imagedir argument, not both. Issue "python TFLite_detection_image.py -h" for help.')
        sys.exit()

    # If neither an image or a folder are specified, default to using 'test1.jpg' for image name
    if (not IM_NAME and not IM_DIR):
        IM_NAME = img_elec

    pkg = importlib.util.find_spec('tensorflow')
    if pkg is None:
        from tflite_runtime.interpreter import Interpreter
        if use_TPU:
            from tflite_runtime.interpreter import load_delegate
    else:
```

```
        from tensorflow.lite.python.interpreter import Interpreter
        if use_TPU:
            from tensorflow.lite.python.interpreter import load_delegate

    # If using Edge TPU, assign filename for Edge TPU model
    if use_TPU:
        # If user has specified the name of the .tflite file, use that name, otherwise use default 'edgetpu.tflite'
        if (GRAPH_NAME == 'detect.tflite'):
            GRAPH_NAME = 'edgetpu.tflite'



    # Get path to current working directory
    CWD_PATH = os.getcwd()

    # Define path to images and grab all image filenames
    if IM_DIR:
        PATH_TO_IMAGES = os.path.join(CWD_PATH,IM_DIR)
        images = glob.glob(PATH_TO_IMAGES + '/*')

    elif IM_NAME:
        PATH_TO_IMAGES = os.path.join(CWD_PATH,IM_NAME)
        images = glob.glob(PATH_TO_IMAGES)

    # Path to .tflite file, which contains the model that is used for object detection
    PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,GRAPH_NAME)

    # Path to label map file
    PATH_TO_LABELS = os.path.join(CWD_PATH,MODEL_NAME,LABELMAP_NAME)

    # Load the label map
    with open(PATH_TO_LABELS, 'r') as f:
```

```
    labels = [line.strip() for line in f.readlines()]

if labels[0] == '???':
    del(labels[0])

# Load the Tensorflow Lite model.
# If using Edge TPU, use special load_delegate argument
if use_TPU:
    interpreter = Interpreter(model_path=PATH_TO_CKPT,
                              experimental_delegates=[load_delegate('libedgetpu.so.1.0')])
    print(PATH_TO_CKPT)
else:
    interpreter = Interpreter(model_path=PATH_TO_CKPT)

interpreter.allocate_tensors()

# Get model details
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
height = input_details[0]['shape'][1]
width = input_details[0]['shape'][2]

floating_model = (input_details[0]['dtype'] == np.float32)

input_mean = 127.5
input_std = 127.5

start_time = time.time()
# Loop over every image and perform detection
for image_path in images:
```

```
        # Load image and resize to expected shape [1xHxWx3]
        image = cv2.imread(image_path)
        image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        imH, imW, _ = image.shape
        image_resized = cv2.resize(image_rgb, (width, height))
        input_data = np.expand_dims(image_resized, axis=0)

        # Normalize pixel values if using a floating model (i.e. if model is non-quantized)
        if floating_model:
            input_data = (np.float32(input_data) - input_mean) / input_std

        # Perform the actual detection by running the model with the image as input
        interpreter.set_tensor(input_details[0]['index'],input_data)
        interpreter.invoke()

        # Retrieve detection results
        boxes = interpreter.get_tensor(output_details[0]['index'])[0] # Bounding box coordinates of detected objects
        classes = interpreter.get_tensor(output_details[1]['index'])[0] # Class index of detected objects
        scores = interpreter.get_tensor(output_details[2]['index'])[0] # Confidence of detected objects
        #num = interpreter.get_tensor(output_details[3]['index'])[0]  # Total number of detected objects (inaccurate and not needed)

        # Loop over all detections and draw detection box if confidence is above minimum threshold
        for i in range(len(scores)):
            if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):

                # Get bounding box coordinates and draw box
                # Interpreter can return coordinates that are outside of image dimensions, need to force them to be within image using max() and min()
                ymin = int(max(1,(boxes[i][0] * imH)))
```

```
                xmin = int(max(1,(boxes[i][1] * imW)))
                ymax = int(min(imH,(boxes[i][2] * imH)))
                xmax = int(min(imW,(boxes[i][3] * imW)))

                # Draw label
                object_name = labels[int(classes[i])] # Look up object name from "labels" array using class index
                label = '%s' % (object_name) # Example: 'person: 72%'
                labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2) # Get font size
                label_ymin = max(ymin, labelSize[1] + 10) # Make sure not to draw label too close to top of window
                cv2.rectangle(image, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED) # Draw white box to put label text in
                cv2.putText(image, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2) # Draw label text

            # All the results have been drawn on the image, now display the image
            cv2.imshow('Q=quit', image)
            print("PROCESSING TIME: %s seconds" % (time.time() - start_time))

            # Press any key to continue to next image, or press 'q' to quit
            if cv2.waitKey(0) == ord('q'):
                break

        # Clean up
        os.remove("image.jpg")
        cv2.destroyAllWindows()

#Image
    def pushButton_Image(self):
        self.open_dialog_box()
    def open_dialog_box(self):
```

```
img = QFileDialog.getOpenFileName()
img_elec = img[0]


# Define and parse input arguments
parser = argparse.ArgumentParser()
parser.add_argument('--modeldir', help='Folder the .tflite file is located in',
                    default='medbox')
parser.add_argument('--graph', help='Name of the .tflite file, if different than detect.tflite',
                    default='detect.tflite')
parser.add_argument('--labels', help='Name of the labelmap file, if different than labelmap.txt',
                    default='labelmap.txt')
parser.add_argument('--threshold', help='Minimum confidence threshold for displaying detected objects',
                    default=0.5)
parser.add_argument('--image', help='Name of the single image to perform detection on. To run detection on multiple images, use --imagedir',
                    default=None)
parser.add_argument('--imagedir', help='Name of the folder containing images to perform detection on. Folder must contain only images.',
                    default=None)
parser.add_argument('--edgetpu', help='Use Coral Edge TPU Accelerator to speed up detection',
                    action='store_true')


args = parser.parse_args()


MODEL_NAME = args.modeldir
GRAPH_NAME = args.graph
LABELMAP_NAME = args.labels
min_conf_threshold = float(args.threshold)
use_TPU = args.edgetpu
```

```
    # Parse input image name and directory.
    IM_NAME = args.image
    IM_DIR = args.imagedir

    # If both an image AND a folder are specified, throw an error
    if (IM_NAME and IM_DIR):
        print('Error! Please only use the --image argument or the --imagedir argument, not both. Issue "python TFLite_detection_image.py -h" for help.')
        sys.exit()

    # If neither an image or a folder are specified, default to using 'test1.jpg' for image name
    if (not IM_NAME and not IM_DIR):
        IM_NAME = img_elec

    # Import TensorFlow libraries
    # If tensorflow is not installed, import interpreter from tflite_runtime, else import from regular tensorflow
    # If using Coral Edge TPU, import the load_delegate library
    pkg = importlib.util.find_spec('tensorflow')
    if pkg is None:
        from tflite_runtime.interpreter import Interpreter
        if use_TPU:
            from tflite_runtime.interpreter import load_delegate
    else:
        from tensorflow.lite.python.interpreter import Interpreter
        if use_TPU:
            from tensorflow.lite.python.interpreter import load_delegate

    # If using Edge TPU, assign filename for Edge TPU model
    if use_TPU:
        # If user has specified the name of the .tflite file, use that name, otherwise use default 'edgetpu.tflite'
```

```
    if (GRAPH_NAME == 'detect.tflite'):
        GRAPH_NAME = 'edgetpu.tflite'


# Get path to current working directory
CWD_PATH = os.getcwd()

# Define path to images and grab all image filenames
if IM_DIR:
    PATH_TO_IMAGES = os.path.join(CWD_PATH,IM_DIR)
    images = glob.glob(PATH_TO_IMAGES + '/*')

elif IM_NAME:
    PATH_TO_IMAGES = os.path.join(CWD_PATH,IM_NAME)
    images = glob.glob(PATH_TO_IMAGES)

# Path to .tflite file, which contains the model that is used for object detection
PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,GRAPH_NAME)

# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH,MODEL_NAME,LABELMAP_NAME)

# Load the label map
with open(PATH_TO_LABELS, 'r') as f:
    labels = [line.strip() for line in f.readlines()]

# Have to do a weird fix for label map if using the COCO "starter model" from
# https://www.tensorflow.org/lite/models/object_detection/overview
# First label is '???', which has to be removed.
if labels[0] == '???':
    del(labels[0])
```

```
# Load the Tensorflow Lite model.
# If using Edge TPU, use special load_delegate argument
if use_TPU:
    interpreter = Interpreter(model_path=PATH_TO_CKPT,
                              experimental_delegates=[load_delegate('libedgetpu.so.1.0')])
    print(PATH_TO_CKPT)
else:
    interpreter = Interpreter(model_path=PATH_TO_CKPT)

interpreter.allocate_tensors()

# Get model details
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
height = input_details[0]['shape'][1]
width = input_details[0]['shape'][2]

floating_model = (input_details[0]['dtype'] == np.float32)

input_mean = 127.5
input_std = 127.5

start_time = time.time()
# Loop over every image and perform detection
for image_path in images:

    # Load image and resize to expected shape [1xHxWx3]
    image = cv2.imread(image_path)
    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    imH, imW, _ = image.shape
```

```
        image_resized = cv2.resize(image_rgb, (width, height))
        input_data = np.expand_dims(image_resized, axis=0)

        # Normalize pixel values if using a floating model (i.e. if model is non-quantized)
        if floating_model:
            input_data = (np.float32(input_data) - input_mean) / input_std

        # Perform the actual detection by running the model with the image as input
        interpreter.set_tensor(input_details[0]['index'],input_data)
        interpreter.invoke()

        # Retrieve detection results
        boxes = interpreter.get_tensor(output_details[0]['index'])[0] # Bounding box coordinates of detected objects
        classes = interpreter.get_tensor(output_details[1]['index'])[0] # Class index of detected objects
        scores = interpreter.get_tensor(output_details[2]['index'])[0] # Confidence of detected objects
        #num = interpreter.get_tensor(output_details[3]['index'])[0]  # Total number of detected objects (inaccurate and not needed)

        # Loop over all detections and draw detection box if confidence is above minimum threshold
        for i in range(len(scores)):
            if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):

                # Get bounding box coordinates and draw box
                # Interpreter can return coordinates that are outside of image dimensions, need to force them to be within image using max() and min()
                ymin = int(max(1,(boxes[i][0] * imH)))
                xmin = int(max(1,(boxes[i][1] * imW)))
                ymax = int(min(imH,(boxes[i][2] * imH)))
                xmax = int(min(imW,(boxes[i][3] * imW)))
```

```
                # Draw label
                object_name = labels[int(classes[i])] # Look up object name from "labels" array using class index
                label = '%s' % (object_name) # Example: 'person: 72%'
                labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2) # Get font size
                label_ymin = max(ymin, labelSize[1] + 10) # Make sure not to draw label too close to top of window
                cv2.rectangle(image, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED) # Draw white box to put label text in
                cv2.putText(image, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2) # Draw label text

            # All the results have been drawn on the image, now display the image
            cv2.imshow('detected', image)
            print("PROCESSING TIME: %s seconds" % (time.time() - start_time))

            # Press any key to continue to next image, or press 'q' to quit
            if cv2.waitKey(0) == ord('q'):
                break

        # Clean up
        cv2.destroyAllWindows()

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

APPENDIX F. REFERENCES

1. Venosa, A., *Medical Daily*, in *History Of The Pharmacy: How Prescription Drugs Began And Transformed Into What We Know Today*. 2016.
2. Kathleen Holloway, T.G., *Drug, and Therapeutics Committees - A Practical Guide*. 2003, Arlington, Virginia, US: Management Sciences for Health.
3. Cerio, N.A.L.B.M.R.M.S.T.A.A.P., *Assessment of the legibility of the handwriting in medical prescriptions of doctors from public and private hospitals*. 2015, Quezon City 1101, Philippines: Elsevier B.V.
4. *This is Why Many Doctors Have Sloppy Handwriting*, in *The Healthy*. 2020.
5. Russon, A.R., *Shorthand*, in *Encyclopædia Britannica*. 2017.
6. F Javier Rodríguez-Vera, M., Y Marín, MD, A Sánchez, MD, C Borrachero, MD, and E Pujol, MD, *Illegible handwriting in medical records.* Journal of the Royal Society of Medicine, 2002: p. 95(11): 545–546.
7. Vallantin, L., *Why you should not trust only in accuracy to measure machine learning performance*, in *Lima Vallantin*. 2018.
8. McDermaid, D., *What is Design to Cost? An Overview with Examples*, in *aPriori*. 2019.
9. Charatan, F., *Family compensated for death*. 1999, BMJ.
10. Caplan, J., *Cause of Death: Sloppy Doctors*. 2007 Jan 15.
11. *Errors associated with outpatient computerized prescribing systems*. 2011, J Am Med Inform Association. p. 18(6):767–73.
12. *What is Python?*, in *opensource.com*.
13. *Object Detection*, in *Mathworks*.
14. *Image Quality Factors*, in *IMATEST*. 2004-2020.
15. Yegulalp, S., *What is TensorFlow? The machine learning library explained*, in *InfoWorld*. 2019.
16. Alexey Bochkovskiy, C.-Y.W., Hong-Yuan Mark Liao, *YOLOv4: Optimal Speed and Accuracy of Object Detection.* 2020.
17. Alake, R., *Deep Learning: Understanding The Inception Module*, in *Towardsdatascience*. 2020.

18. Howard, M.S.a.A., *MobileNetV2: The Next Generation of On-Device Computer Vision Networks*. 2018, Google Research.
19. Hui, J., *SSD object detection: Single Shot MultiBox Detector for real-time processing*, in *Jonathan Hui*. 2018.
20. Tyagi, N., *Introduction To YOLOv4*, in *Analytic Steps*. 2020.
21. Kumar, A., *Faster RCNN — Object Detection*, in *Abhishek Kumar*. 2020.
22. Wardynski, D., *What is a Microprocessor and How Does it Work?*, in *Brainspire*. 2019.
23. Pokhrel, S., *Image Data Labelling and Annotation — Everything you need to know*, in *Toward Data Science*. 2020.
24. Brownlee, J., *How to Configure Image Data Augmentation in Keras*, in *Machine Learning Mastery: Make Developers Awesome at Machine Learning*. 2019.
25. *Raspberry Pi 4*, in *RaspBerryPi*.
26. *Jetson Nano*, in *Nvidia. Developer*.
27. *8MP Camera Module with SONY IMX219 sensor for Raspberry Pi*, in *SunFounder*.

APPENDIX G. CURRICULUM VITAE

JEVANELLE AMICAN

COMPUTER ENGINEERING
Technological Institute of the Philippines (TIP) MLA
Address: Blk. 13 Lot 77 Masinop St. Purok 4 Lupang Arenda Taytay Rizal
Tel. No.: None
Email Address: mjamican@tip.edu.ph
Cellular No.: 09554812135

EDUCATION AND EXPERIENCE

- Technological Institute of the Philippines, Manila
  Bachelor of Science in Computer Engineering
- Experts Academy for Applied Information Technology Inc. Makati City
  Intern

DESIGN PROJECT COMPLETED/ RESEARCH

Floripix

This application includes different features such as, Snaps wherein the captured images of the flowers from the user are stored, Offline Flower Directory that a user can access even with the absence of the internet, and Bookmarks where the user can immediately get back on a certain flower in the directory. The developers consider user-device interaction and the fact that these algorithms must execute in a resource-constrained environment. For each problem, the developers perform both theoretical and empirical analyses in an attempt to optimize performance and usability. This project makes contributions related to the successful implementation of image processing and artificial intelligence techniques.

MedBox-AI: An AI-Powered Medicine Scanner using Deep Learning-Based Translation of Doctor's Prescription

MedBox-AI provides a smart way to recognize the recommendation of a doctor utilizing deep machine learning. This effort would support some people who are suffering while seeing the doctor's medications. The key role of this paper lies in the data used for training purposes. This device uses TensorFlow as a machine learning library and Custom Repository to match the partial string to the name of the drug. With MedBox, instances of misinterpretation of the names of pharmaceutical items may be reduced. This makes it easier for ordinary people to understand what a doctor prescribes on a prescription, and also to help pharmacists.

KNOWLEDGE, SKILLS AND ATTITUDE

Having graduated from TIP with its orientation towards outcome-based education, I have acquired and can demonstrate the following student acquire outcomes (knowledge, skills, and attitudes) necessary to the practice of the computing profession:

- *Analyze complex problems and identify and define the computing requirements appropriate for solution.*
- *Use modern techniques and tools of computing practice in complex activities.*
- *Understand professional, ethical, legal, security, and social issues and responsibilities relevant to professional computing.*

LEADERSHIP ACTIVITIES

Hubs and Cisco Router Student Society - Treasurer

SEMINARS AND TRAININGS ATTENDED

- AWS Siklab T.I.P Cloud Computing
  TIP-Manila Anniversary Hall
  July 20, 2019
- Experience the Possibilities of AI in the Industry-Day 2
  TIP-Manila Arlegui Seminar Room
  August 17, 2019
- Basic Android Application Development using B4A
  TIP-Manila Arlegui Rm.228
  September 14, 2019
- On-the-Job Training Security Program
  Expert Academy Makati City via Cisco Webex
  October 05, 2020, to November 26, 2020
- Security Awareness Level 1
  Expert Academy Makati City via Cisco Webex
  October 05 - 09, 2020
- Introduction to Malware Threats
  Trend Micro Secure Learning via Zoom
  October 22, 2020
- Introduction to Email Threats and Security Attacks
  Trend Micro Secure Learning via Zoom
  November 19, 2020
- IT Project Management: A Practical Approach
  TIP-Manila through Google Meet
  December 12, 2020
- Stay Connected: A Deep Dive to the Evolving World of Cloud Computing
  TIP-Manila through Google Meet
  December 17, 2020
- Stay Connected: A Deep Dive to the Evolving World of Cloud Computing
  TIP-Manila through Google Meet
  December 19, 2020

EXTRA AND CO-CURRICULAR ENGAGEMENT AND VOLUNTEER WORK

- Tech 101-NCR Student Bootcamp
  Participants

CERTIFICATIONS

- CCNA Routing and Switching: Introduction to Networks
- CCNA Routing and Switching: Routing and Switching Essentials
- CCNA Routing and Switching: Scaling Networks
- CCNA Routing and Switching: Connecting Networks Introduction to Cybersecurity
- Security Awareness Level 1
- Sophos Certified Technician - Sophos
- Sophos Certified Engineer - Sophos
- NSE 1 Network Security Associate - Fortinet
- NSE 2 Network Security Associate - Fortinet
- Splunk 7.x Fundamentals Part 1 (eLearning) - Splunk>

REFERENCES

Engr. Cherry Casuat
OJT Adviser
Technological Institute of the Philippines
cherrycasuat2019@gmail.com
(02)8733-9117

.

MICHAEL MAR CABIE

COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) MLA

Address: **213 Doña Aurora St., Atimonan, Quezon**

Email Address: mmmcabie@tip.edu.ph

Cellular No.: **+639389355510**

CAREER OBJECTIVE

Reliable individual willing to work in the position of a Computer Engineer, To develop and improve skills on English communication, troubleshooting, networking design, as well as my programming skills to develop algorithms for data collection and system performance.

DESIGN PROJECTS COMPLETED/ RESEARCHES

MedBox-AI: An AI-Powered Medicine Scanner using Deep Learning-Based Translation of Doctor's Prescription
(Current Design Project)
The project aims to develop a mechanism that will help pharmacists in performing their job effectively, specifically to develop a device that interprets the doctor's handwriting on the prescription, to create an algorithm that translates the handwriting of the doctors in determining the prescriptions prescribed by the doctor and to make a device that can help the pharmacist to easily understand the prescription.

Student Behavior System
This project is a python program and API object detection, that detects the behavior of the student, whether the student is Attentive and Non-attentive in class, that outputs a timestamp, object id, status and percentage of the status, and then generates in CSV file, this project helps teachers/professor to monitor his/her student who is really attentive in class.

KNOWLEDGE, SKILLS AND ATTITUDE

Having graduated from TIP with its orientation towards outcome-based education, I have acquired and can demonstrate the following student acquire outcomes (knowledge, skills and attitudes) necessary to the practice of the computing profession:

- *Analyze complex problems and identify and define the computing requirements appropriate for solution.*
- *Use modern techniques and tools of the computing practice in complex activities.*
- *Inderstand professional, ethical, legal, security and social issues and responsibilities relevant to professional computing.*

SEMINAR AND TRAINING ATTENDED

EXPERTS ACADEMY
Rada, Legazpi Village, Makati, Metro Manila
March 08, 2020

OTHER SKILLS

- *Programming (Python-Anaconda, Java-Android Studio, Assembly-Tasm)*
- *Web development (CodeIgniter, Microsoft Visual Studio and XAMPP)*
- *PC Troubleshooting*
- *Average communication skills both English and Tagalog*
- *Average practical use of computer with documentation tools of MS Office (Word, Excel and Powerpoint) and other tools such as (Adobe Photoshop, Android Studio and PyQt)*
- *Hardware hands-on: (Formatting Desktop or OS, Cleaning formatting disk partitions, setting up dual boot operating system such as Ubuntu, Windows 7 & 10 and Configuring & setting up Raspberry Pi)*

REFERENCES

Engr. Cherry D. Casuat
Faculty Member, Computer Engineering Department
Technological Institute of the Philippines - Manila
ccasuat.cpe@tip.edu.ph
09989516508

Engr. Alvin S. Alon
Faculty Member, Computer Engineering Department
Technological Institute of the Philippines - Manila
alvinsarragaalon@hotmail.com

Engr. Jennifer B. Enriquez
Chairperson, Computer Engineering Department
Technological Institute of the Philippines - Manila
(02) 733-9142

I hereby certify that the above information are true and correct to the best of my knowledge and belief.

Cabie, Michael Mar P.
Applicant

ANTONIO MIGUELLE CAPILLO

COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) MLA

Address: **144 Ipil St. Camella 5 Homes Pulanglupa Dos, Las Piñas City**

Tel. No.: **(8) 846-40-42**

Email Address: **miguelle619@gmail.com**

Cellular No.: **+639395493149**

DESIGN PROJECTS COMPLETED/ RESEARCHES

A Multiple Input Door Lock using Arduino
This project is a door lock system that would only unlock when a correct passcode was used in the keypad and the use of the correct RF id card and the buzzer would go off if enough incorrect inputs were used. It uses Arduino that is connected to a solenoid lock that would be in charge of opening or closing the door. It was presented through a miniature house with the solenoid lock placed behind the door.

Simulating CCNA Security Configurations on GNS
This project is a basic implementation of CCNA security policies and settings on a network using GNS3 as a platform. It was basically a lab activity when applied on the usual packet tracer but instead it would be applied to the GNS3. After the implementation, the group would check if the correct settings were set in the network. Then the results of the testing would then be recorded and compared to the expected output to see if the simulation was done correctly.

Hospital Management System
This group project was created in the hopes of improving the efficiency of the hospital's ability to gather the patient's information from the patients themselves. And at the same time interface of the project was user friendly that the medical staff that would be in charge of the system would be able to use it easily. Which would result in the staff would be able to provide correct services that every patient would need. This system primarily uses java as its programming language and MySQL for the database configuration.

KNOWLEDGE, SKILLS AND ATTITUDE

As a student from TIP with its orientation towards outcome-based education, I have acquired and can demonstrate the following student acquire outcomes (knowledge, skills and attitudes) necessary to the practice of the computing profession:

- *Analyze complex problems and identify and define the computing requirements appropriate for solution.*
- *Use modern techniques and tools of the computing practice in complex activities.*
- *Understand professional, ethical, legal, security and social issues and responsibilities relevant to professional computing.*

SEMINARS AND TRAININGS ATTENDED

- ICPEP.SE NCR Convention "Unfolding Reality Through Emerging Technologies"
  Technological Institute of the Philippines - Manila, P.Casal Campus, P.E Center and the P.E Center Annex.
  February 29, 2020
- IP Management and Technology Transfer
  Technological Institute of the Philippines - Manila, Arlegui Building, Teresita Quirino Hall
  December 17, 2019
- CCNA Routing and Switching: Introduction to Networks - Cisco
- CCNA Routing and Switching: Routing and Switching Essentials - Cisco
- CCNA Routing and Switching: Scaling Networks - Cisco
- Sophos Certified Technician - Sophos
- Sophos Certified Engineer - Sophos
- NSE 1 Network Security Associate - Fortinet
- NSE 2 Network Security Associate - Fortinet
- Splunk 7.x Fundamentals Part 1 (eLearning) - Splunk>
- Security Awareness Level 1

EXTRA AND CO-CURRICULAR ENGAGEMENTS AND VOLUNTEER WORKS

- Hubs and Cisco Routers Student Society
  Member
  July 7 2017 - 2018

OTHER SKILLS

- Proficient in English both written and verbal communication
- Interpersonal Skills
- Flexibility/ Adaptability
- Teamwork/ Collaboration
- Knowledgeable in MS Office Programs (Word, Powerpoint, Excel, Publisher, and Project)
- Basic knowledge in LabVIEW, Arduino IDE, and AutoCAD
- Proficient in programming (C++, C#)
- Familiar in programming (Python, Assembly, JAVA)
- Proficient in Network Management
- Familiar with Hardware Troubleshooting

REFERENCES

Dr. Alvin Alon
Assistant Professor,
Technological Institute of the Philippines, Manila
Aalon.cpe@tip.edu.ph
09493684564

GILBERT DUMLAO

COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) MLA

Address: **Suntrust Parkview Natividad Lopez Ermita, Manila**

Email Address: **gddumlao05@gmail.com**

Cellular No.: **+639471811380**

ON-THE-JOB TRAINING/ WORKING EXPERIENCE

- **IT Intern – (April 2019 - May 2020)**

[24]7.ai Philippines
106 Valero, Makati, 1227 Kalakhang Manila

- Support the IT team in maintaining hardware, software, and other systems. Assist with troubleshooting issues and provide technical support. Organize and maintain IT resources

- **Customer Service Representative – (Nov. 2018 – Mar. 2019)**

Inspiro Relia Inc.
España Blvd, Sampaloc, Quezon City, 1008 Metro Manila

- Customer Service Representative handling customer concern. Taking order of the customer through calls and handing their concern about their account or order. Making our customer satisfied.

KNOWLEDGE, SKILLS AND ATTITUDE

Having graduated from TIP with its orientation towards outcome-based education, I have acquired and can demonstrate the following student acquire outcomes (knowledge, skills and attitudes) necessary to the practice of the computing profession:

- Proficient in window-based applications such as Word, Excel and PowerPoint
- Use modern techniques and tools of the computing practice in complex activities
- Understand professional, ethical, legal, security and social issues and responsibilities relevant to professional computing
- Trained to work under pressure
- Good interpersonal communication skill
- Open to challenging opportunities and eager to learn more

CERTIFICATION

- **CCNA Routing and Switching: Introduction to Networks**
  Cisco Networking Academy (December 4, 2017)
- **CCNA Routing and Switching: Routing and Switching Essentials**
  Cisco Networking Academy (July 17, 2019)
- **Google Ads Display Certification**
  Google Digital Academy Skillshop (November 2020)
- **The Fundamental of Digital Marketing**
  Google Digital Garage (November 2020)

ALVIN PAULO JUNIO

COMPUTER ENGINEERING

Technological Institute of the Philippines (TIP) MLA

Address: **578 Angat Bulacan**

Tel. No.: **022532994**

Email Address: **apjunio18@gmail.com**

Cellular No.: **+639993621521**

CAREER OBJECTIVE

To establish a career in information technology where I can demonstrate the learning outcomes of the Information Technology program of the Technological Institute of the Philippines (TIP), a proram accredited by the US-based outcomes-oriented ABET (Accreditation Board for Engineering and Technology), Computing Accreditation Commision.

DESIGN PROJECT COMPLETED/ RESEARCH

Flood Detection using Raspberry Pi 4
This project is primarily focused on the detection of the flood by installing a flood detector device with a camera beside the bridge column. The camera is facing the three lines with different colors. If one of the colors was tempered by the river water, the device will send an alarm to the community that the water level in the river is high. This aims to alert the community and the authorities to be aware and be ready for the approaching flood.

KNOWLEDGE, SKILLS AND ATTITUDE

Having graduated from TIP with its orientation towards outcome-based education, I have acquired and can demonstrate the following student acquire outcomes (knowledge, skills, and attitudes) necessary to the practice of the computing profession:

- *Analyze complex problems and identify and define the computing requirements appropriate for solution.*
- *Use modern techniques and tools of the computing practice in complex activities.*
- *Understand professional, ethical, legal, security and social issues and responsibilities relevant to professional computing.*
- Capable in cabling and configuring small Networks.
- Proficient in using Microsoft Office program.
- Knowledgeable in Configuring Cisco Routers
- Hardware hands-on: Troubleshooting Computer Hardware and Softwares; Prototyping projects assemble and disassemble.

OTHER SKILLS

CCNA Routing and Switching: Introduction to Networks
CCNA Routing and Switching: Routing and Switching Essentials
CCNA Routing and Switching: Scaling Networks
CCNA Version 7: Enterprise Networking, Security, and Automation

REFERENCE

Engr. Alvin Alon
Faculty Member, Computer Engineering Department
Technological Institute of the Philippines - Manila
alvinsarragaalon@hotmail.com
02733-9142

VIOLET STREET HOBART VILLAGE 2
BRGY.UGONG VALENZUELA CITY
09208742456
dkaramiro@gmail.com

DUSTIN KYLE RAMIRO

SKILLS

- Use modern techniques and tools of the computing practice in complex activities.
- Analyze complex problems and identify and define the computing requirements appropriate for solution.
- Understand professional, ethical, legal, security, and social issues and responsibilities relevant to professional computing.

EXPERIENCE

Experts Academy for Applied IT Inc, Makati City – *Intern*

EDUCATION

Technological Institute of the Philippines, Manila – *Bachelor of Science in Computer Engineering*

S.Y 2017 - 2020, 5th Year Student

CERTIFICATIONS

- CCNA Routing and Switching: Introduction to Networks - Cisco
- CCNA Routing and Switching: Routing and Switching Essentials - Cisco
- CCNA Routing and Switching: Scaling Networks - Cisco
- CCNAv7: Enterprise Network, Security, and Automation - Cisco
- Sophos Certified Technician - Sophos
- Sophos Certified Engineer - Sophos
- NSE 1 Network Security Associate - Fortinet
- NSE 2 Network Security Associate - Fortinet
- CyberArk Certified Trustee - CyberArk
- Splunk 7.x Fundamentals Part 1 (eLearning) - Splunk>
- Security Awareness Level 1

REFERENCE

Jenikka Mariel Rotol

Settlement Analyst

Alorica Teleservices

jenikkamariel@gmail.com

09194412930