

### 3. DESIGN APPROACH

This document describes the design approach for the Gestura product. Gestura uses a gesture-based system to control smart home devices. The system is divided into five subsystems. The six subsystems are Gesture Recognition, User Interface, Power Supply, Sensor Module, Data Communications, and 3D Modeling. The product has the following constraints: Economic, Safety, Hardware, and Accuracy. The main requirements for the design include communication and accuracy.

#### 3.1. Design Options

For Gestura, the main problem was how to get the data for gesture recognition. Two main design choices were being considered. The first design option used a wristband that the user would wear to track the movement of the hand. The other approach involved using cameras and AI to detect gestures.

##### 3.1.1. Design Option 1

The first design option considered was using a wristband with a gyroscope and an accelerometer that collects data to send to a microcontroller over Wi-Fi which would interpret the data and signal an output to the smart home device. The advantages of this approach are that it is cheaper to make and can provide more accurate data, but the disadvantages are that wearing a wristband could be uncomfortable or inconvenient for some users to use. This approach was done in the IEEE study by Alemuda and Lin [1].

##### 3.1.2. Design Option 2

The option that the team chose was to use computer vision to recognize gestures. Two advantages are that it is user-friendly, and there is no extra technology that the user must wear. On the other hand, the disadvantages are that it is very costly and extracting valuable data is much more difficult. The design team decided to tackle the second approach because the comfort and accessibility to the user in this approach outweigh the benefit of the low cost in the first approach.

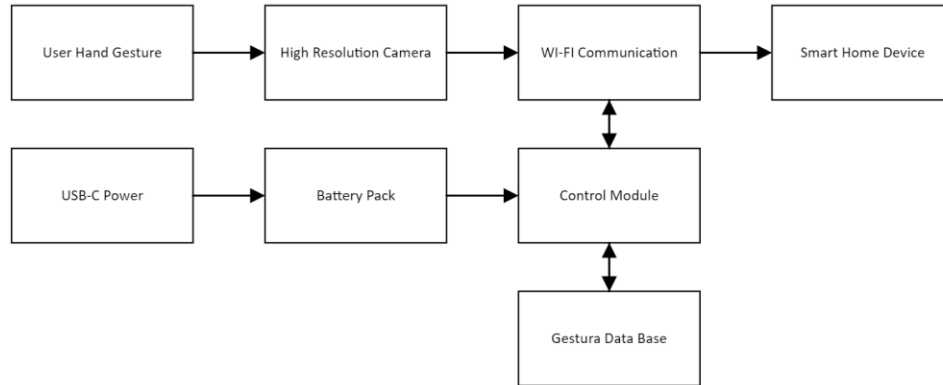
#### 3.2. System Overview

Gestura is a system that uses a gesture to create a command to send to a smart home device. Figure 3-1 shows how Gestura works at a glance.



**Fig. 3-1: Gestura System at a Glance (Level 0)**

Going into more depth, the system works through wireless communications via Wi-Fi. All communications go through the control module. Figure 3-2 explains an in-depth view of the system.



**Fig. 3-2: Gestura System Diagram (Level 1)**

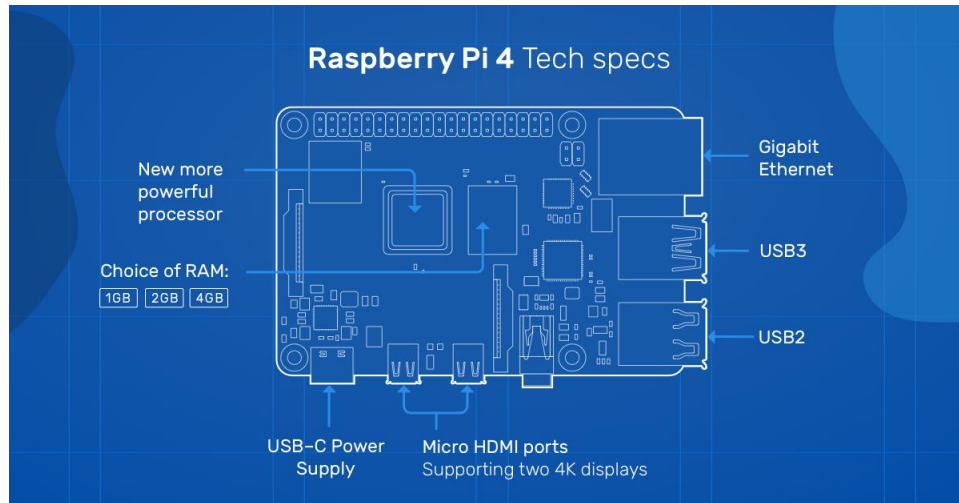
The gesture is captured through a camera and sent to the control module. The control module analyzes the video to detect gestures. The detected gesture outputs a corresponding command to control a smart home device. This is explained more in the subsystem descriptions below.

### 3.2.1. Microprocessor

	Raspberry Pi 4 Model B[3]	Allen Bradley PLC[2]
Advantages	Cost-Efficient (\$90)	Industrial Grade
	Easy Connectivity	
	Can use multiple programming languages	
Disadvantages	Less Reliable	Can only use one programming language
		Less Economical (\$255)

**Table 3-1: Advantages vs. Disadvantages**

For this project, the team considered two different options for the microprocessor. The first option was the Raspberry Pi 4 Model B, while the second option was an Allen Bradley PLC (Programmable Logic Controller). The Raspberry Pi offers multiple advantages compared to the PLC. First, the Raspberry Pi is more cost effective. The Raspberry Pi is \$165 cheaper than the PLC. Secondly, the Raspberry Pi can use multiple different programming languages, while the PLC can only use one. Additionally, for the PLC to be the team's main control module, it would need extra products to be able to connect to the team's different subsystems. The Raspberry Pi, however, already has on-demand WIFI and Bluetooth connectivity. Given that the team's project focused on residential use, the team found the PLC to be an overkill solution, and the Raspberry Pi met the project's specific needs more effectively. **Fig. 3-3** is a picture of the team's chosen microprocessor.



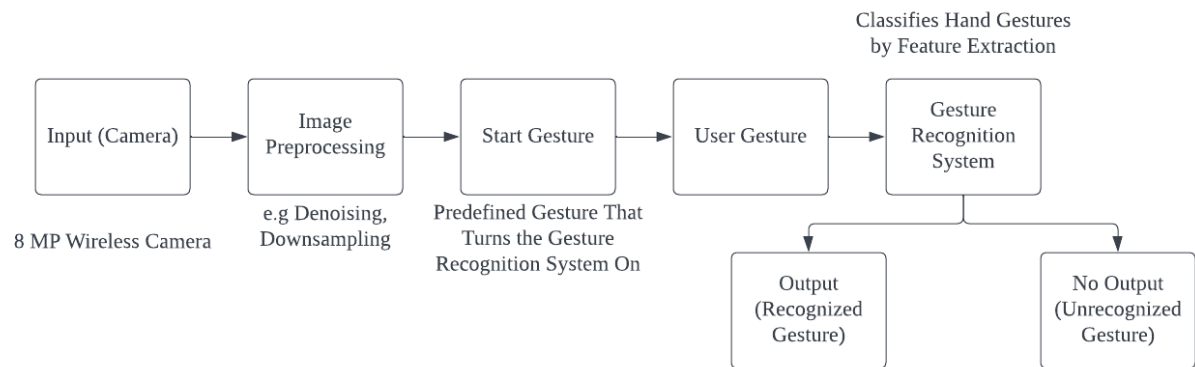
**Fig. 3-3: Raspberry Pi 4 Model B [3]**

### 3.3. Subsystems

Gestura has various subsystems designed to work together to allow the user to use gestures to control their smart home devices. The prototype has 6 subsystems. Subsystem 1 is the gesture recognition subsystem. It will detect and analyze the gestures the user is using. Subsystem 2 is the user interface subsystem. The interface will be on a tablet to allow the user to interact, connect smart home devices, and display user feedback. Subsystem 3 is the power supply subsystem. The power supply will power the LCD screen and Raspberry Pi for the tablet. Subsystem 4 is the sensor module subsystem. The sensor module will be the subsystem detecting and sending the information of the gestures to the Raspberry Pi. Subsystem 5 is the data communications subsystem. The data communication subsystem is how information is being passed between the control module, the smart home devices, and the cameras. The final subsystem is subsystem 6 which is the 3D design subsystem. The 3D design is designed to hold the Raspberry Pi, a fan, LCD screen, power supply, and the internal wiring for the tablet.

#### 3.3.1. Gesture Recognition Subsystem

The gesture recognition subsystem is responsible for detecting gestures by using computer vision and analyzing the gestures using a machine learning algorithm to assign them to a certain function. The popular programming languages for computer vision and machine learning are C++ and Python [4]. The design team has decided to use python because of its extensive library [5], and the experience that the design team has with the programming language. **Fig. 3-4** below shows a general flowchart of the gesture recognition subsystem.

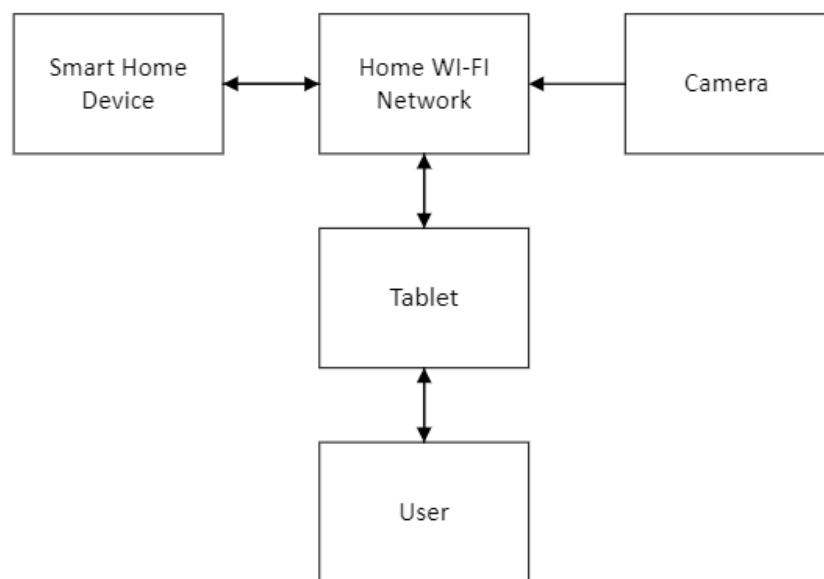


**Fig. 3-4: Gesture Recognition Flowchart**

An 8 MP camera was chosen because an increase in the quality of the video input would result in an increase in the accuracy of the output in vision-based AI. According to Cheng, “Vision-based AI requires outstanding video input in order to be effective” [6]. Image preprocessing such as denoising and down sampling is used to reduce the file size which will increase the efficiency and speed of gesture recognition. The “start gesture” is a gesture stored on the device that makes Gestura look for user gestures once the gesture is performed. This is done to prevent unintentional activation of the device. User gestures are gestures that, once the confidence threshold for the gestures has been reached, activate the smart home device that they are programmed to affect. Training data is stored on the device with predefined gestures. The team is exploring adding user-programmable gestures in the following semester.

### 3.3.2. User Interface Subsystem

The user interface is on a tablet powered by the Raspberry PI. The touchscreen allows the user to connect to a home Wi-Fi network or new smart home devices. It also outputs if a gesture is detected and the corresponding command. The figure below offers a visual depiction of the user interface.



**Fig. 3-5 User Interface Diagram**

The team decided to go with a 7-inch LCD Touch Screen for the touch screen. The team decided to go with a Hosyond LCD touch screen panel. The touch screen is compatible with the Raspberry PI and an economical and practical choice. Amazon offers several different touch screens with resolution quality and sizes based upon cost. The team decided that based upon resolution and cost, the Hosyond offers ample size and quality at a lower price compared to other options. The options are listed below in **Table 3-2**.

Product Description	Price
Hosyond 3.5 Inch 480x320 Touch Screen TFT LCD SPI [7]	\$17.99
Hosyond 7 Inch IPS LCD Touch Screen Display Panel 1024×600 [8]	\$45.99
InnoView Portable Monitor 4K Touchscreen - 14 Inch [9]	\$299.99

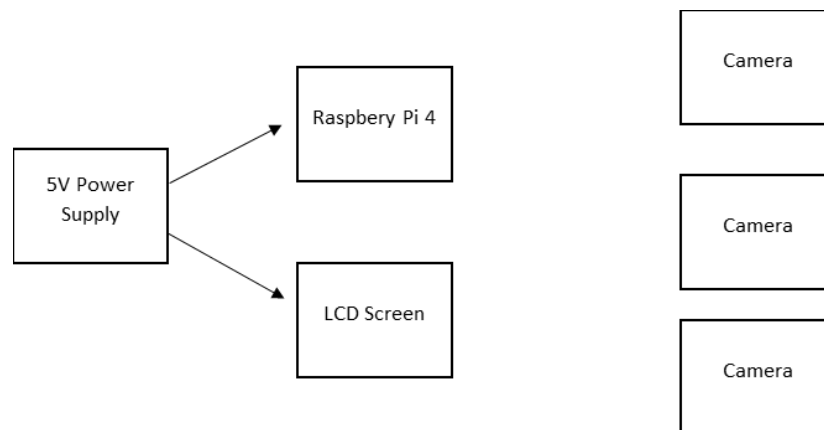
**Table 3-2 Touch Screen Options**

These options define a clear choice for the team, however, there are many options to choose from. With the product price and user experience, the Hosyond 7-inch display is the best option.

### 3.3.3. Power Supply Subsystem

The Raspberry Pi and the LCD Touch screen are powered by a 5V rechargeable power supply. The power supply uses four 18650 batteries in series to achieve the longevity of the device. The device has 7000 mAh, and the power supply can be recharged by a USB type C charger. The cameras have a separate power supply.

**Fig. 3-6** below shows a general diagram of the power supply subsystem.

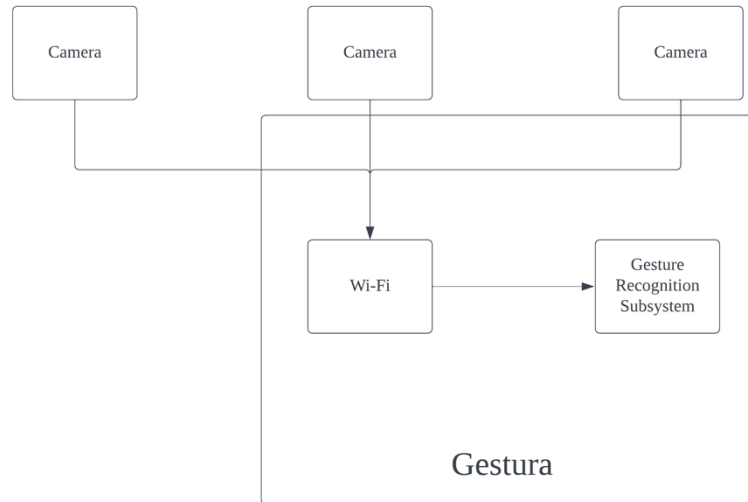


**Fig 3-6: Power Supply Diagram**

The voltage required by both the screen and the controller is 5V. The team chose to use a rechargeable power supply to eliminate the hassle of additional cost of batteries. With the camera constantly monitoring for gestures, a good battery life is needed for the device. The team chose four 18650 batteries in series to achieve 7000mAh. With 7000mAh, the device can run for 2-3 days with moderate use before needing charge.

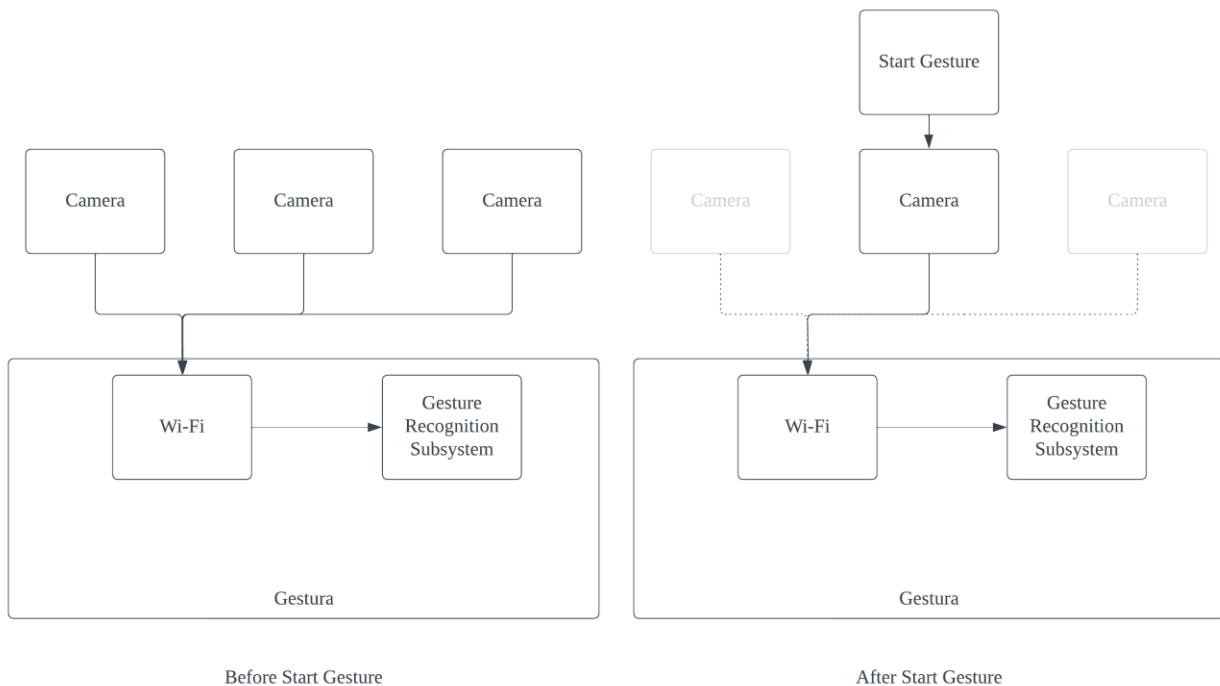
### 3.3.3. Sensor Module Subsystem

The sensor module subsystem will be responsible for taking the input from the cameras of the system. This will be done by using Wi-Fi to retrieve footage from the cameras, which is sent to the gesture recognition subsystem. **Fig. 3-7** below shows the flow chart from the input to the output of the subsystem.



**Fig. 3-7: Sensor Module Flowchart**

The code is in Python using the cv2 library. This is used to take input each frame of the recording of the camera. This can be used for multiple cameras. Each is sent to the gesture recognition subsystem to look for a start gesture. Once the start gesture is performed on a certain camera, Gestura takes input from only that camera. **Fig. 3-8** demonstrates this interaction.

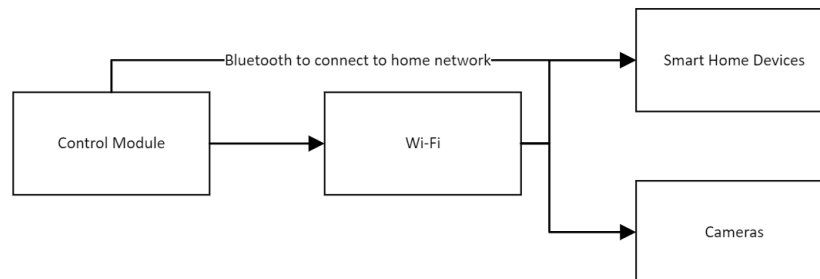


**Fig. 3-8: Start Gesture Interaction**

This implementation speeds up the response time of Gestura by efficiently using processing power where it is important.

### 3.3.4. Data Communications Subsystem

The system uses Wi-Fi and Bluetooth to communicate. It uses Bluetooth to connect to new smart home devices and uses Wi-Fi to communicate with the cameras, and smart home devices. Wi-Fi is also used to connect to a home network. **Fig. 3-9** provides a general overview of the data communications subsystem.

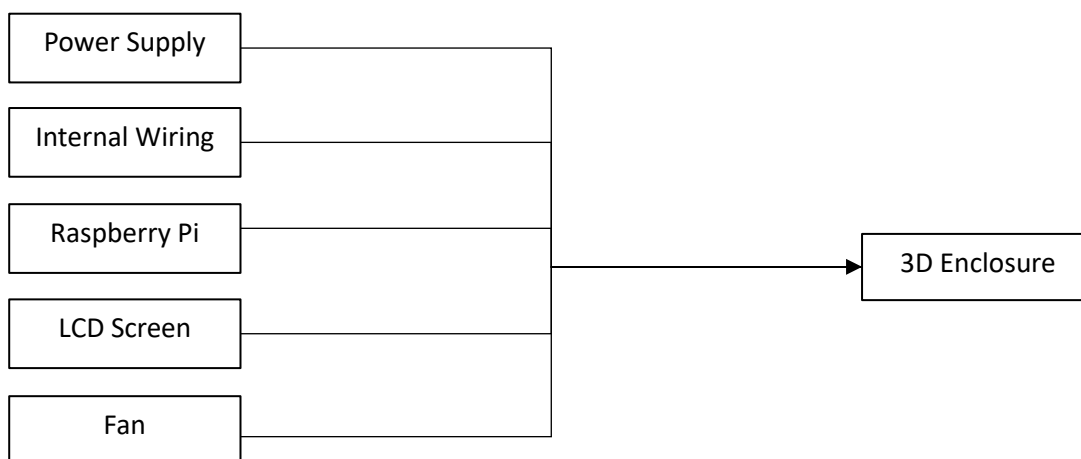


**Fig. 3-9: Data Communications Diagram**

The cameras, and smart home devices use the home network to communicate with the control module. The gesture recognition system is on the control module which translates the signals into usable commands.

### 3.3.5. 3D Design Subsystem

The 3D design subsystem holds the Raspberry Pi, fan, LCD screen, power supply, and internal wiring in an enclosure for the tablet. The ports of the Raspberry Pi are also accounted for in the design by leaving open holes to easily access the ports and being able to plug wiring to it. **Fig. 3-10** displays all components that are part of the 3D design subsystem.

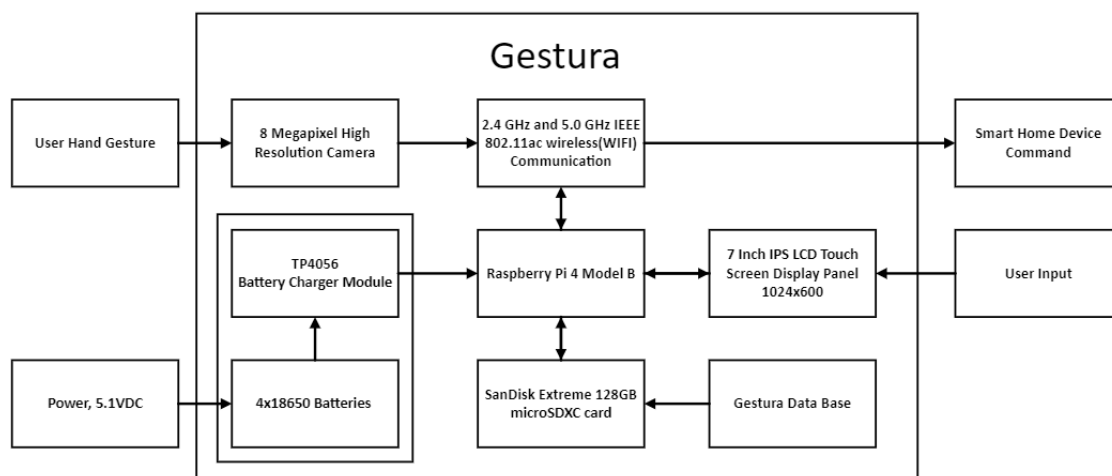


**Fig. 3-10: 3D Design Diagram**

## 3.4. Level 2 Prototype Design

For Gestura, the team is planning to encase the control module, cables, battery pack, battery charger module, etc. in a 3D printed case. The case has a touchscreen monitor on it to act like a tablet. The team has intentions on using ABS plastic filament to 3D the case. ABS plastic is a sturdy enough material for the case to uphold moderate wear and tear.

### 3.4.1. Level 2 Diagram



**Fig. 3-11: Diagram for Gestura (Level 2)**

Each subsystem provides a vital role in Gestura's functionality. The choices made here provide a foundation of how Gestura works. Our next step is to validate these concepts with rigorous testing, addressing any problems as they arise.

### References:

- [1] F. Alemuda and F. J. Lin, "Gesture-Based Control in a Smart Home Environment," 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and



IEEE Smart Data (SmartData), Exeter, UK, 2017, pp. 784-791, doi: 10.1109/iThings-GreenCom-CPSCoM-SmartData.2017.120.

- [2] “PLC hardware - Allen Bradley 2080-LC30-24QBB series B, surplus in ...,” PLC Hardware, <https://www.plchardware.com/Products/RA-2080-LC30-24QBB-B-NSS.aspx> (accessed Oct. 24, 2023).
- [3] Raspberry Pi, “Raspberry pi 4 model B specifications,” Raspberry Pi, <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/> (accessed Oct. 23, 2023).
- [4] Y. Noema, “Top 3 programming languages for implementing a computer vision system,” Medium, <https://medium.com/imagescv/my-top-3-programming-languages-for-implementing-a-computer-vision-system-2aa0a3ad0a2a> (accessed Oct. 20, 2023).
- [5] M. R. Munawar, “C++ vs. python: Choosing the best language for computer vision development,” VisoByte, <https://www.visobyte.com/2023/07/cpp-vs-python-choosing-the-language-for-computer-vision-development.html> (accessed Oct. 20, 2023).
- [6] D. (Wei-Y. Cheng, “Ai with high IQ (image quality).,” Ambarella, <https://www.ambarella.com/blog/ai-with-high-iq/> (accessed Oct. 22, 2023).
- [7] “Amazon.com: Hosyond 3.5 inch 480X320 touch screen TFT LCD SPI display ...,” Amazon, <https://www.amazon.com/Hosyond-480x320-Screen-Display-Raspberry/dp/B0BJDTL9J3> (accessed Oct. 22, 2023).
- [8] “Hosyond 7 Inch IPS LCD Touch Screen Display Panel 1024×600 Capacitive Screen ...,” Amazon, <https://www.amazon.com/Hosyond-Display-1024%C3%97600-Capacitive-Raspberry/dp/B09XKC53NH> (accessed Oct. 22, 2023).
- [9] “Amazon.com: InnoView Portable Monitor 4K Touchscreen - 14 Inch Auto-Rotating FreeSync ...,” Amazon, <https://www.amazon.com/InnoView-Portable-Monitor-Touchscreen-Auto-Rotating/dp/B08R9TSCL9> (accessed Oct. 22, 2023).