International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 111–136

p-ISSN 2655-8564, e-ISSN 2685-9432

Hardware Architecture and Implementation of an AI Pet Robot

Elton Lemos[1,*], Abhishek Ghoshal[1], Aditya Aspat[1]

[1]*Department of Computer Engineering,*
Xavier Institute of Engineering, Mahim, Mumbai, Maharashtra, India
[*]*Corresponding Author: eltonlemos2411@gmail.com*

(Received 03-08-2020; Revised 17-08-2020; Accepted 18-08-2020)

Abstract

The concept of AI companionship is gaining popularity in recent times. Our project is an attempt to create a robotic companion that can act just like a pet. We created a Pet Robot that can perform various functions like recognise people, recognise objects, recognise emotions, play games, follow a person around, listen to voice commands, dance and so on. This paper focuses on the hardware aspect of the robot and the working of the Core program (Kernel). It further discusses the algorithm that was implemented in this architecture and compares the final version with other variations. Our work was able to create a fully functioning AI pet robot by using just cheap commercially available development boards, motors and a Desktop PC.

Keywords: pet, robot, hardware, kernel

1 Introduction

For the longest time, man has tried to seek companionship in the form of pets. There are a lot of benefits of having a pet like entertainment, provide emotional support, protect you and your belongings, etc. However it owning a pet also has its responsibilities. Our project tries to create such a pet that one does not necessarily have

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

to deal with these responsibilities but still get the benefits. The AI pet robots available in the market are no doubt innovative but also expensive. Not to mention that they also are not perfect. Keeping the pros and cons of these pet robot we tried to create our own version of an AI pet robot. In this paper we explain the hardware components we used and how they interact with one another to form a functioning robot that is capable of performing complex AI and CV algorithms. We discuss the various problems that we faced due to the limitations of the AI algorithms and the Hardware that we used and how we overcame them. The crux of the paper is the system architecture that we used and the working of the Kernel of the robot. Throughout this paper all the AI algorithms mentioned are cited from this [1] paper.

2 Existing System

When we started our research there were already a few robots available in the market. These were quite expensive and designed by big companies like Sony. As a result there were not any papers or designs available related to these robots. Sony's Aibo was our main inspiration to create a pet robot. We wanted to create a similar pet but without the exorbitant cost. By researching on these pet robots we tried to understand what features made the robot more "Pet" like and how we could incorporate them in our robot. All of these Pets run on centralised systems. Here the main decision making hardware and computing hardware are the same. This requires a custom designed computer boards which increases the production cost. The board needs to be designed keeping in mind that it has to fit inside the robot, not over-heat and be powerful enough to perform complex AI programs. These are very tight constraints and you need to give up on one to gain the other. This in turn limits the performance of the Robot.

2.1 Aibo

Sony's attempt at an AI pet is Aibo, a realistic dog based AI pet with a slew of features like facial recognition, having personalities, emotion detection, automatic battery recharging and many more. The Aibo robot will do all the basic tricks like "sit", "handshake" and others like a normal pet. In additions to these it can also take pictures and

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

wirelessly connect to the internet. It is designed solely to be a companion and will not do simple tasks like telling the morning report or watching for intruders. However these are very expensive examples selling for upwards of $2,900. Key take away from Aibo is that a pet needs to be cute. Sure some pets have their benefits (like dogs guard the house) but pets are kept to provide company and entertainment.

2.2 Pibo

Pibo is a cute bi-pedal AI pet robot designed by a Korean Developer Circulus. This robot has a few AI assistant features like alarms, weather report, message notifications, etc. Other than that the robot can take photos are directly upload them to the users face-book profile. It has other playful features like stories and jokes. Pibo can also play music while dancing to it. Pibo can also be reprogrammed by the users giving users control over how this robot functions. This also brings more power to the community as they can design personalised programs for Pibo to execute. Pibo sells currently for $840. Pibo is more of an AI assistant than a pet. It can perform all of the functions that an AI assistant can whilst walking around the house. This is not really a bad thing as it can fill the void of the “benefit” of having a pet point mentioned earlier. As mentioned above Pibo is a biped robot however after viewing its demo videos we felt that it has very slow move-ment because of this reason. Seeing that walking motion does not really bring any pet like attributes we decided to omit it and implement wheels instead.

2.3 Lovots

There are simpler versions of plush smart toys that are capable of being therapeutic for the elderly and specially those people suffering from anxiety and dementia. There maybe cases where senior citizens need a therapy but cannot care for one on their own. These are like stuffed animal covered electronic appliance that has sensors that allow you to interact with it. This sort of “empathy” interaction is something that is upcoming in artificial intelligence. One of the recent products in these fields are called Lovots (Love + Robot). These have been created in Japan by Groove X and are designed to provide humans with love or at least the perception of love. According to their website,

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

Lovot will react to your mood and do all it can to fill you with joy and re-energize you. However, these Lovots also have exorbitant price tags costing upwards of $3,100. Lovots show us that understanding and showing emotions are an important feature of a Pet. Development of this feature would make the robot very relate able. Infact perfection of this feature is an important step towards creating the ideal AI pet.

3 Hardware Components

In this section we have mentioned and explained all the hardware components that we have used in this Project. We tried to keep the explanation as crisp as possible such that to only have details that are relevant to the project and possible future scope. The hardware that we have used in this project were affected by budget and availability constraints. The are many other solutions for the same hardware that can be explored.

3.1 Raspberry Pi 4

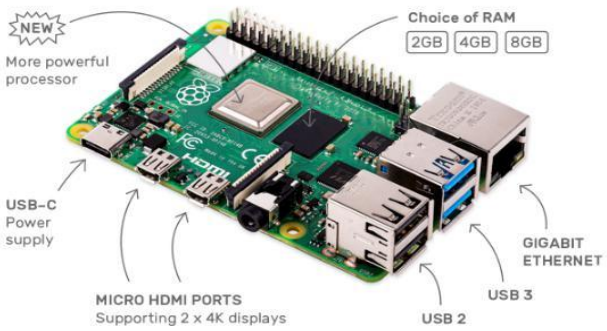


Figure 1. Raspberry Pi 4 [3]

Raspberry Pi are one of the worlds top most developer friendly computer boards available in the market. It is an economic solution to a Desktop Pc as it is fairly cheap and compact. Raspberry Pi boards use the linux based operating system named Raspbian. The latest addition to the Raspberry Pi family is the Raspberry Pi 4 model B. According to the Datasheet [2] the Raspberry Pi 4 has a Quad core 64-bit ARM-Cortex A72 running at 1.5GHz, H.265 (HEVC) hardware decode (up to 4Kp60), VideoCore VI 3D Graphics and Supports dual HDMI display output up to 4Kp60. This Raspberry Pi comes with 1, 2 and 4 Gigabyte LPDDR4 RAM options. For the Interfacing the Raspberry Pi has 802.11b/g/n/ac Wireless LAN, Bluetooth 5.0 with BLE, 1x SD Card, 4x USB ports, 1x Raspberry Pi camera port, 1x Gigabit Ethernet port and 40 pin GPIO

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

header supporting various interface options. We can see the Raspberry Pi 4 board in Figure 1. The role of the Raspberry Pi (RPi) in our project is to provide vision, communication and decision making. Therefore our Robot depends on the Raspberry Pi ability to provide a high performance CPU, RAM, USB interface and Camera interface.

3.2 Raspberry Pi NOIR Camera V2

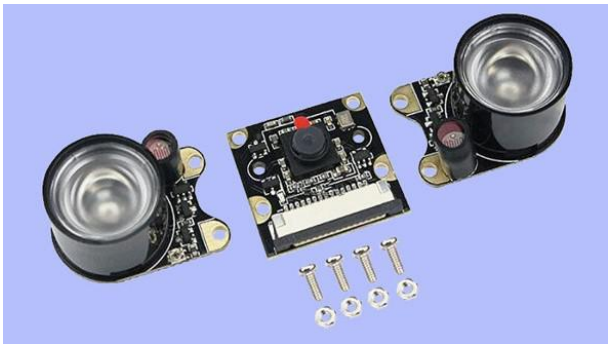


Figure 2. Raspberry Pi NOIR Camera V2 with IR torches [5]

Inorder to provide vision to the robot we need to have a camera installed on the robot. We have a wide range of option in todays market for cameras however we needed some-thing that was efficient and cost effective. Due to RPi's hardware, we can not only used USB cameras but also Cameras designed specially for the Rpi using the CSi interface. The RPi NOIR Camera V2 [4] is one such specially designed camera for the RPi. It has a high quality 8 megapixel Sony IMX219 image sensor and is capable of 3280×2464 pixel static images, and also supports 1080p30, 720p60 and 640×480p60/90 video. We can see the Raspberry Pi NOIR Pi camera Figure 2. The NOIR camera does not have an IR filter meaning it can also see IR light. This property when coupled with an IR Torch can let the Robot see even in the Dark which is also one of the features of our project. In addition to that the RPi Cam Interface (explained in the next subsection) which is an integral part of our Robot works only with CSi interfaced cameras. Hence the Raspberry Pi NOIR Camera V2 was our best option.

International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 111–136

p-ISSN 2655-8564, e-ISSN 2685-9432

3.3 Arduino UNO



Figure 3. Arduino UNO [6]

Arduino UNO is an open-source Hardware [4]. Its a development board that is great for starer level developers who want to experiment with coding and robotics. It is a micro-controller board based on the ATmega328P. Its operating voltage is 5V and recommended input voltage is 6-12V. It has a Vin pin, 3 ground pins, a 3.3V pin, 5V pin and an Aref pin. It has 14 Digital pins of which 6 can provide PWM output and another 6 Analog pins. It is a small lightweight board of the dimensions $68.6\,mm \times 53.4\,mm$ and weighs $25g$. Figure 3 is Arduino UNO board displayed on the official Arduino website.The board also has a USB port that can be used to program it as well as power it. We use the Arduino IDE software installed on a Desktop computer to program the UNO board. The arduino is cheaper board and also supports PWM and analog signals unlike the Raspberry Pi. Hence we will be using this as our motor controller.

International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 111–136

p-ISSN 2655-8564, e-ISSN 2685-9432

3.4 Arduino Motor Shield



Figure 4. Arduino Motor Shield

The Arduino Motor Shield is designed specifically for the Arduino UNO to upgrade it to be able to control motors without the need of other external circuitry. There are multiple versions of the motor shield available in the market made by different companies. We chose the version that had 4 motor ports with 2 servo ports as per the requirement of the current design. The motor shield has 2 L293D ICs to control the motors. It also has circuitry to take external power from a 12V battery which it uses to power the motors that it will control. We can see the Arduino Motor Shield in Figure 4. This power can also be used to power the UNO as the Motor Shield can convert the voltage to 5V and give it to the UNO. It must be noted that if we use a shield, other pins will not be available for use conventionally. If we need to access these pins we will need to solder the wiring to the back of the board. Some Motor Shields do address this problem by providing extra pins that can connect to the UNO's unused pins (pins that are not used by the Motor shield).

3.5 Display

Inorder to give visual feedback to the user we used a 0.96 inch oled display. The display is connected to the RPi via the SDA and SCL pins of the I2C protocol. We followed the setup instructions as per an online article [7] for it. It has a detailed explanation of how to use the libraries installed in the setup and also has a sample code that we can use to design our own code. Figure 5 shows us the Oled Display for Arduino boards.

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432



Figure 5. 0.96 inch Oled Display [7]

3.6 Servo Motors and Motors

For movement of the camera we will be using 2 MG90s. It is a PWM controlled servo which runs at operating voltages of 3.3V to 6V. For the bots wheels we will be using 3 standard 12V DC motors. We experimentally found out that 5V DC motors are not able to move the robot so we advice using motors in the Voltage range of 6-24V.

3.7 Power Bank 5V

Inorder to use the RPi and Arduino UNO we need a 5V reliable power source and for that we used a 2.5A 5V 2x port Power Bank that is available in the market. The power bank can be also used to power other parts that are connected to the Arduino like buttons, servos, lights, etc. The pins of an Arduino UNO can only provide an output current of $200\ mA$ safely. Therefore we should use the powerbank instead to power the other parts to prevent the Arduino from burning up. The plus side to this is that the Arduino and the parts will share the same Vin and Ground making the circuitry much easier. It is much safer to use a Power Bank rather than batteries to power the Raspberry Pi as the Power Bank comes with the circuitry that protects the device that it is powering from mishaps.

3.8 Battery 12V

Wheel Motors that can move the bot considering the weight had to be High Torque in the Voltage range of 7-24V. Using a 5V motor or just providing the above mentioned motors with 5V would not be suitable for the robot. Therefore we need to add a 12V

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

battery to the Circuit especially for the motors. This also assists to solve the burning up problem of the Arduino mentioned in the Power Bank subsection.

3.9 Sound Card

Raspberry Pi does not have an inbuilt sound card for input and therefore we have to add a external USB sound card to attach a mic to it. It can also be used to improve the quality of output sound if better sound card is used than the one inbuilt in RPi.

3.10 Speaker

We used the Speaker of a broken google home. It has a decent sound reach as well as was small enough to fit on the robot. Inorder to amplify the output signal to the Speaker we used a PAM8403S amplifier chip. An alternate solution could also be to use a bluetooth speaker.

3.11 Microphone

The microphone needs to be small and compact. Generic lapel microphones can record clear audio and are quite small so we used this for the project. Any other good quality microphone can be used like a bluetooth headset mic or USB mics.

3.12 Buttons and Joysticks

As of now we have not implemented the “Hot word” response like that of Google or Amazon’s Alexa. So the inorder to get the attention of the robot and give it commands, we need to push a button to start the voice recognition. We decided to replace that button with a joystick for a hand so instead we have to shake the hand of the robot to get its attention.

International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 111–136

p-ISSN 2655-8564, e-ISSN 2685-9432

3.13 Chasis and Wheels



Figure 6. Robot Base

We used the Wheel base of a broken Remote Controlled Toy Robot as seen in Figure 6. The base of the Toy Robot had 3 wheels with motors and 8x AA battery slot. the wheels on the robot were omnidirectional wheels so just 3 wheels were enough for it to move. The motors on the toy were old and needed to be replaced. The battery slot was not big enough to fit the power bank or our battery pack. However if reshaped a bit we could fit an arduino with the motor shield in it securely. The torso of the robot is just the powerbank to which the RPi and battery are stuck with double tape. The head of the robot is the camera and the mechanism to make it move. It is a 2 servo DIY mechanism that can be found in any robotics shop that lets the camera move in 2 axis 180 degrees each.

4 Wiring Connections

We divided Wiring into two aspects: Data connections and Electric connections. The data connections will show which component communicates with another components. The Electrical connections shows which power supply provides current to which component. Unless specifically mentioned which cable we have used to connect the components it can be assumed that we used standard DIY development cables.

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

4.1 Data Wiring

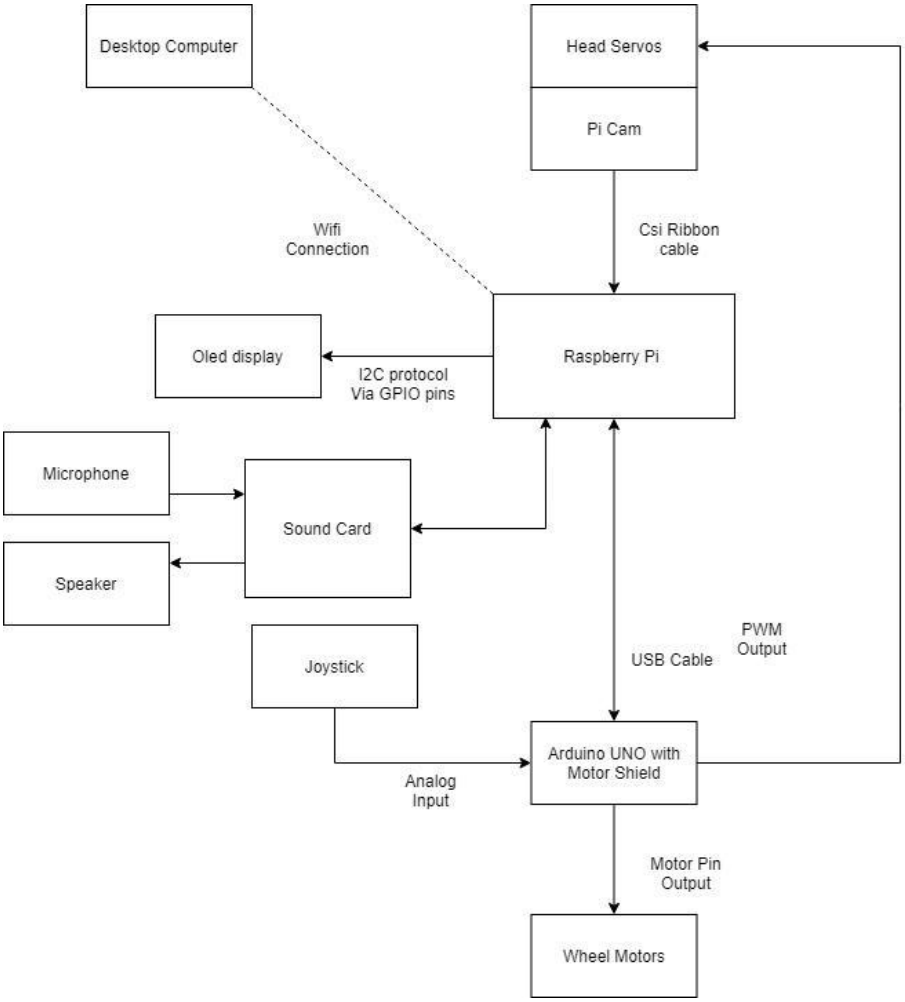


Figure 7. Data Connections

The Figure 7 only contains components of the robot that need logical connections. The tail of the arrows show that the data is output from there and the head of the arrow shows where the data is gonna be input. Except for the connections between the Raspberry Pi, Desktop PC and Arduino, all connections are one way connections as they are either sensors or motors. The Raspberry Pi and the PC are connected wirelessly using a wifi network. The Raspberry Pi and the Arduino are connected Via a USB Cable and communicate with each other Via the Serial Communication. We experimentally found out that the baud rate of 115200 was the most optimal for our implementation. It fit into

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

the sweet spot where there was quicker communication without putting stress on the Serial ports. The microphone is connected via a Sound Card. The Speaker component can be connected to the Raspberry Pi via the Sound card's port or Bluetooth.

4.2 Electric Wiring

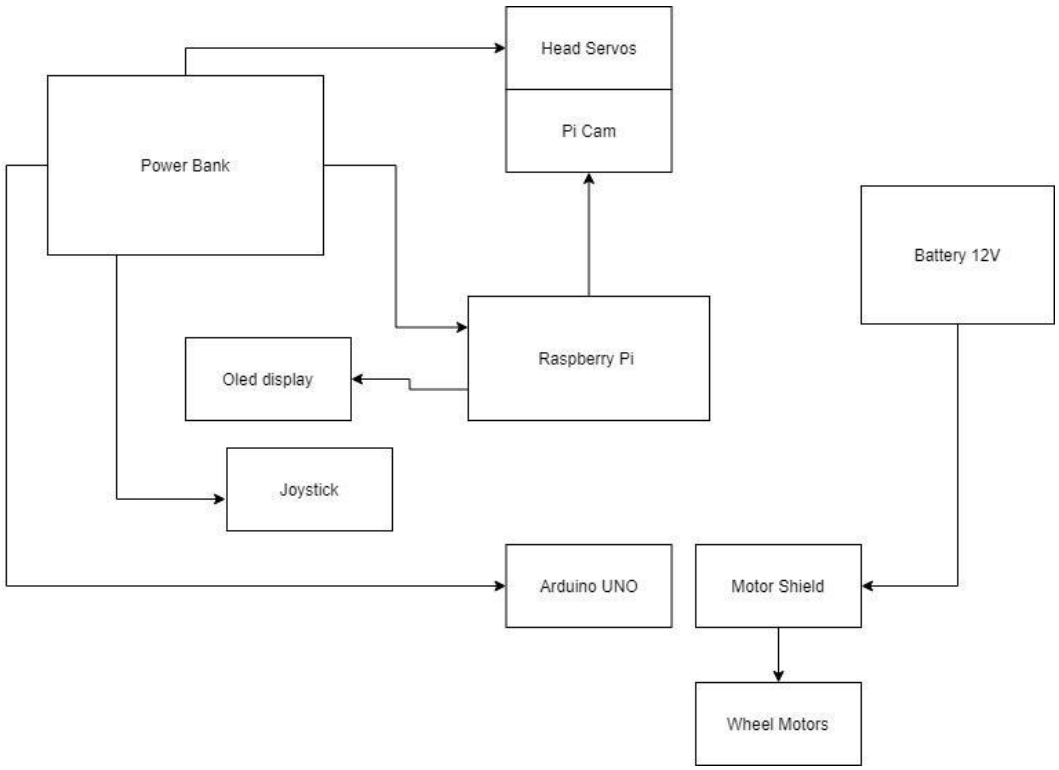


Figure 8. Power Connections

There are 2 main sources of power for the robot ie. The Power Bank and the Battery pack. The Power Bank is 5V while the Battery Pack is 12V. As most of the components work at 5V the power bank powers most of the robots circuitry as shown in shown in Figure 8. If we choose a power bank with more number of ports it becomes easier to design the wiring. Care must be taken that the Power bank must have enough Amperage to be able to be able to handle the requirements of all the parts of the robot. Else we would need another Power Bank or a more efficient power source. The Battery pack is used solely to power the Arduino's motor shield and hence the motors. We need to take care that the Motor Shield is not supplying power to the UNO otherwise due to some reason the voltage regulator on the UNO gets fried, disabling the ability of the UNO to take power from the barrel adapter port or the USB. Due to our lack of expertise in this

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

field we could not figure out why this is happening but we have reason to believe that it is caused because of the current from the USB port and the Vin to the UNO provided by the Motor Shield.

5 Software Components

In this section we have explained all the softwares that are needed to use the hardware that we used for the robot.

5.1 Raspberry OS

Raspberry Pi OS (previously called Raspbian) is the Foundation's official supported operating system [8]. We used the version Buster for our implementation. Raspbian Buster has further two types: Headed and Headless. The Headed version means the interface has a GUI which is displayed via its HDMI while Headless means it has a command line interface and its display ports are disabled. Having a Gui also means that it will have a lot of bloatware and dedicate its resources to provide graphics to the user. Since we need high performance from the RPi we chose to go with the Headless version. One can go to the official Raspberry Pi website to download it and follow the instructions to install it [8] on your Raspberry Pi 4.

5.2 PuTTY

Since we are using the headless version, its display ports are disabled. Therefore if we want to interact with the RPi we need to access it remotely. This can be achieved via the SSH protocol by using PuTTY. For a tutorial on how to install and use PuTTY and use it to connect to you RPi you can view this course [9].

5.3 OpenCV

OpenCV is a programming library mainly aimed at Computer Vision functions. If you want to follow our final implementation then you have to install it on the PC like a standard python library. Else if you want to use the Centralised architecture (explained in the further sections) you will need to install OpenCV on your RPi. This is a special OpenCV library where the code is optimised for the hardware architecture of the RPi for

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

maximum performance. It is highly advised to use the headless version of Raspbian if one wants to use OpenCV as it uses too many resources. There are multiple articles online that can help you install OpenCV for RPi like the one we followed [10]. We advise to find an updated article yourself because During our attempts to install OpenCV and other libraries we experienced difficulties because of miss match of the versions of Raspbian, OpenCV and the steps that we needed to execute.

5.4 RPi Cam Web Interface

RPi Cam Web Interface is a Web interface designed specially for the RPi cameras. It sets up a local webpage which lets us control the RPi's camera. The webpage is set up at the Raspberry Pi's IP address and can be easily accessed via a mobile or a computer. This Web Page is one of the means of communication for the RPi and the PC in our implementation. It only supports the CSi Raspberry Pi camera and not the USB cameras. We can learn more about it and install it from its website [11].

6 System Architecture

When we first designed the robot it was meant to run just purely on the Raspberry Pi. However we soon realised after our implementation with just one feature ie. Face detection and tracking, was too much for the RPi to handle. And that was the simplest of the Computer Vision algorithms of the robot. It became clear to us that if we wanted the robot to work reliably then we needed either better onsite hardware or process the information elsewhere. Due to constraints explained below we had to go with the latter which changed the system architecture entirely. We had to introduce a PC to process the data. In the current implementation the PC and the RPi have to be in the same network to run. Future Scope could be to implement this on a cloud computer.

6.1 Centralised Architecture

In this architecture there is only one computing unit: Raspberry Pi. We were able to implement only couple of features of the Robot before the Raspberry Pi started to throttle. For this architecture, the Raspberry pi was the heart of the system. The video stream from the Camera was processed by the Raspberry Pi and even the motors were

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

controlled by it. We first implemented a face tracking algorithm where the Raspberry Pi would detect a face using Haar cascade and then control the motors fixed to the camera such that it followed the face. At the start there were not any problems and it seemed like the Raspberry Pi could handle it. However at around the 7th minute the Raspberry Pi began to heat up and throttle which reduced the throughput of frames. It was around the 12th minute that the Raspberry Pi could only process one frame per second. This was unacceptable as it also introduced a considerable amount of latency to the robot. Inorder to implement our project we needed a better computer or an alternate approach. It was also during this phase that we realised some other mistakes. We realised that the Raspberry Pi is susceptible to reverse/induced current from the motors resulting in unresponsive pins. So it was imperative that we introduce a motor controller to do that task. The other problem was that the Raspberry pi only has digital pins. Most of the sensors feed analog data and thus would require a micro-controller in the middle to translate the information. This problem was rectified in the Decentralised version by introducing an Arduino UNO.

6.2 Problems of Centralised Architecture

While creating our Robot we designed two versions one with centralised architecture the other with a decentralised architecture. The following were the evident problems of the Centralised system.

6.2.1 Power

The seamless interaction of our pet with humans was the evaluation criteria for our project. For this our CV and AI programs had to work seamlessly and that is only possible if the computer can handle that kind of load.

6.2.2 Size

As we were creating a cute pet robot it is obvious that the robot cannot be large. That means the computer, motors and mechanisms have to all fit in a small robot. In order to run complex CV and AI algorithms the processor needs to be powerful enough but mostly powerful computers are larger and require a good cooling system which further adds to the size.

6.2.3 Cost

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

In order to run AI programs the Robot needs to have a powerful computer. And that comes at a cost. Not to mention that the size of the computer needs to be small and light. That constraint again adds to the cost. And when we cannot find such computers, development of our own board is the only option which is very expensive.

6.2.4 Compatibility

The Computer board also need to be compatible with the other parts of the robot like the motors, camera and other sensors. So we need a computer that is also good at running AI algorithms but also has the architecture for robotics.

6.3 Decentralised Architecture

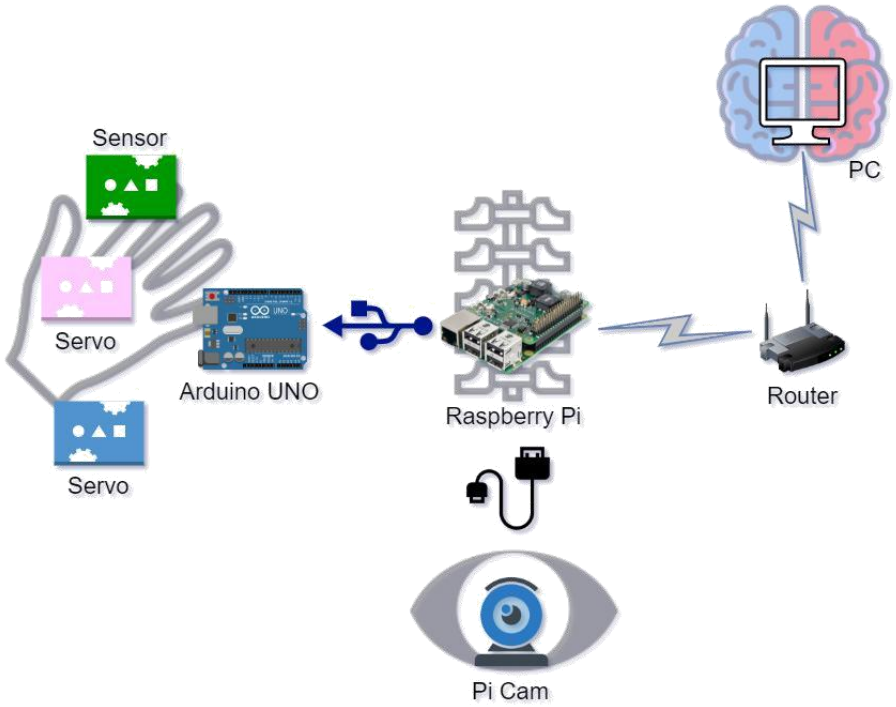


Figure 9. Decentralised Architecture

Inorder to overcome the drawbacks of the Centralised architecture we came up with this decentralised architecture. In this architecture the Computer Vision data is not processed at the RPi but at a Desktop Computer. Thus all the information is not processed at just one place, but rather multiple places relieving the load on the RPi. While technically this does increase the cost of our implementation, a PC or a laptop is already

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

available with everyone. We used the following design for our decentralised architecture explained with the aid of Figure 9.

6.3.1 Brain (PC)

In our architecture the PC is considered as the Brain of the Robot. It is responsible for performing the complex computations and execution of the CV and AI algorithms of the Robot. The Video stream sent by the Raspberry Pi (Spine) is processed by the PC and then the output is sent back to the Raspberry Pi. The output was designed to be simple instructions that the Raspberry Pi had to execute. This released a significant burden on the on-board Processor of the Robot (in this case Raspberry Pi) which made it possible to have a light weight computer to be used in the Robot. For our final implementation we used a Computer with the following specifications:

CPU: Ryzen 2700x

GPU: Nvidia GTX 10

Ram: 16 GB Dual Channel

Router: 300 Mbps

6.3.2 Spine (Raspberry Pi)

Spine or the Raspberry Pi which was the heart of the centralised structure acts as the Kernel of the System. Its primary duty is to coordinate the Arduino UNO, Pi Cam and PC. It has a microphone to take voice commands from the user. The Raspberry Pi is also responsible for determining the current mode of the robot. The Arduino and PC have to Switch modes as per the request of the Raspberry Pi. The reason why we call the Raspberry Pi as the Spine of the System is because it acts as a bridge between the PC and Arduino. The PC sends instructions to the Raspberry Pi. The Raspberry Pi then evaluates the instruction. If the instruction was for the Arduino, then the Raspberry Pi forwards the instruction to the Arduino. Along with that, the Raspberry Pi also acts as a bridge between the camera and the PC.

6.3.3 Eyes (NoIR Camera)

To provide the Robot with vision we used a NoIR Pi camera V2. This Camera when paired with IR torches provide vision to the Robot even when there is no light. This also helps to get a proper illuminated image for the PC to process as the IR torches act like a flash light. This feature is a great help to our Face Recognition Program because the IR

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

light is invisible to the human eyes and hence does not irritate the user even when the camera is pointed directly to the face of the User.

6.3.4 Limbs (Arduino UNO)

To control the Motor movements of the robot we chose to use an Arduino UNO with a motor shield. The Arduino receives instructions from the Raspberry Pi via the Serial bus. It then manipulates the motors as per the given instructions. It also processes data received from the sensors connected to it and only forwards data to the Raspberry Pi when an Event occurs (An Event is when the data from the sensors satisfy a particular condition). This reduces the strain on the Serial bus thus even the Raspberry Pi. It also allows the Arduino to control the motors without consulting the Raspberry Pi or PC in emergency situations. eg. When the distance sensor alerts the Arduino that the robot is close to an object, the arduino can stop the motors at that very moment rather than wait for the data to reach the Raspberry Pi and come back.

7 Internal Communication

This section explains how each processing component communicated with each other. We divided the explanation component wise so one can focus on all the communication requirements between those components. The methods mentioned in this sections are from our implementation of the robot. There are many other methods in which we can carry out the same objective and a detailed study of which method is most efficient could help improve the robot.

7.1 Raspberry Pi and PC

The RPi and the PC are the places where most of the data is communicated and processed. The 2 main types of data that needs to be communicated between them is a video stream and instructions in the form of characters.

7.1.1 Socket

We used a Python Socket program for instructions. The PC was the Server and the Raspberry Pi was the client. The Socket program was two way communication where the PC sends instructions to the Raspberry Pi while the Raspberry Pi uses it to alert the PC of mode changes. The Instructions sent by the PC can be either for the Raspberry Pi

Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

or for the Arduino. It is the Raspberry pi's responsibility to forward that command to the Arduino. The Instruction is just a character (due to a small instruction set. The more functionalities are added to the bot, the more number of instruction codes will be required). The RPi receives the character and uses it to check which instruction the PC is telling the RPi to do. At the same time the RPi can send characters to the PC to tell the PC which program it has to run.

7.1.2 Socket for Images

In the early stages we tried to send frames of the videos to the PC via the Socket program. However this was very inefficient and would often lead to asynchronization. The Raspberry Pi would send frames faster than the PC could process the frames. These frames would pile up in the buffer. The PC would process frames sequentially and thus any lag that occurred would become permanent. As the lag would start to pile, it would completely go out of sync and process frames about 5 seconds later than it was supposed to. Eventually we gave up on this concept and focused on the Rpi Cam Interface.

7.1.3 Rpi Cam Interface

We used the Rpi Cam Interface to set up a local web server where we hosted the video stream from the Pi camera. So the PC could access the video feed via the Url whenever it wanted. This gave the PC the freedom to read frames whenever it was free. It was not forced to read stale frames from the buffer as mentioned in the previous algorithm. This removed the piling up of lag problem that we faced before. However it restricted our System to run on the same network as we were hosting a local web server.

7.2 Raspberry Pi and Arduino UNO

The Arduino Uno and Raspberry Pi are boards that are on the Robot itself. Hence it is possible to have a wired connection between them. To achieve this we used Serial Communication via USB. The Raspberry Pi sends commands and also forwards commands from the PC to the Arduino via this interface. The Arduino on the other hand uses this interface to send sensor values to the Raspberry Pi.

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

8 Implementation

In this section we will explain the step by step working of the hardware of the Robot. However before that it is important to understand what Instruction set and Mode are with respect to our project. The rest of the terms are explained in detail in the earlier sections.

8.1 Instruction Set

Our Pet Robot has many features like face tracking, face detection, follow around, mini games, etc. All of these algorithms do not run simultaneously. Each of them have some kind of instructions that needs to be passed around. There are sometimes errors in received messages due to connection issues. eg if the Arduino is trying to send “a” the Raspberry Pi might read it as “b” maybe because the cable moved a little bit. In such situations if there are individual codes for individual instructions then there is a possibility that an interpretation of the instruction might belong to some other algorithm. This would lead to the robot performing random functions which might not be in the scope of the current algorithm to stop. One such situation occurred when the forward wheel movement instruction from the Follow algorithm was executed by the Arduino while the robot was just supposed to look for faces as per the Face tracking algorithm. Not only did the it execute a wrong instruction but also there was no way of stopping it as the counter to it ie. The stop instruction, can only be executed by the instruction sent by the Follow algorithm. Therefore we separated the instructions as per the algorithms that are supposed to be executing currently. We called them modes. Thus only the instruction which belonged to the current mode were executed while the rest were ignored. This also gave us the opportunity to reduce the size of the instruction set where we could have one character (code) be interpreted as different instructions depending on the current mode. The decision of which algorithm is to be executed is decided by the Mode the bot is in. And that is decided by the RPi as mentioned earlier. If the bot is in Face tracking mode, then the PC will run the Face Tracking algorithm and give instructions in the Face tracking instruction set. Similarly the RPi and Arduino will perform their algorithms and interpret instructions from the Face tracking instruction set

Volume 2, Issue 2, pages 111–136

p-ISSN 2655-8564, e-ISSN 2685-9432

8.2 Working

1. The PC starts the Socket Server.
2. The RPi starts a Socket client and tries to connect to the PC.
3. On Establishment of connection the RPi starts a Web Server via the RPi Cam Web Interface.
4. The Raspberry Pi starts the Face detection and Tracking mode.
5. The PC takes frames from the Web Server and runs the CV algorithm depending on the Robots mode.
6. The PC then sends Instruction to the RPi via the Socket.
7. On receiving the Instructions the RPi interprets it and executes it.
8. If the Instruction is for the Arduino, the RPi forwards the Instruction to the Arduino Via Serial Communication.
9. The Arduino then Interprets the Instruction and executes it.
10. If the Arduino's sensors send important information to the Arduino, the Arduino will interpret it.
11. The Arduino might take action. OR
12. The Arduino will alert the RPi of the Sensors information.
13. The Rpi will then Interpret the information and take action.
14. If the information was a handshake (movement of the joystick up and down) then the RPi will listen to the User for commands.
15. The Rpi will then interpret the commands, decide which mode the robot needs to be in and inform the PC and Arduino of the mode change.
16. On receiving the mode change the PC and Arduino will acknowledge it and change its mode.

Steps 5 onward run on loop till the robot is turned off. It should be noted that the RPi Cam Web Interface runs as a different user on the the RPi and does not follow or interfere with the flow execution of the main code of the robot

International Journal of Applied Sciences and Smart Technologies
Volume 2, Issue 2, pages 111–136
p-ISSN 2655-8564, e-ISSN 2685-9432

9 Evaluation

In our efforts to improve the performance of the robot we tested the changes made to our architecture against the Face Tracking Algorithm. We also used PCs with different specifications. As a standard we used video stream of 800x600p resolution as the input to our algorithm. The criteria for evaluation was Fps, Avg Latency and Latency after 7 mins. We also capped the frames per second at 60fps. We have the following PC specifications for the test.

Raspberry Pi 4: Quad-core ARM Cortex-A72 Processor, 4 GB RAM

PC A: Intel i7 CPU, AMD Radeon R5 M335 GPU, 16 GB RAM

PC B: Intel i7 CPU, Nvidia 970M GPU, 16 GB RAM

PC C: AMD Ryzen 2700x CPU, Nvidia 1070 GPU, 16 GB RAM

10 Results

As per Figure 10 we can see that the more powerful the Computer is, the lesser is the latency. We can also see from Figure10 that when the PC takes frames from the Raspberry Pi, it does not have a lag build up as opposed to when the Raspberry Pi pushes images to the PC. This can be credited to the fact that the PC only gets fresh frames as it reads frames only when it has finished processing the previous frame. As per Figure 11 and Figure 12 we can see that without using the Rpi Cam interface the video would have a large amount of lag thus making it impractical for the Robot.

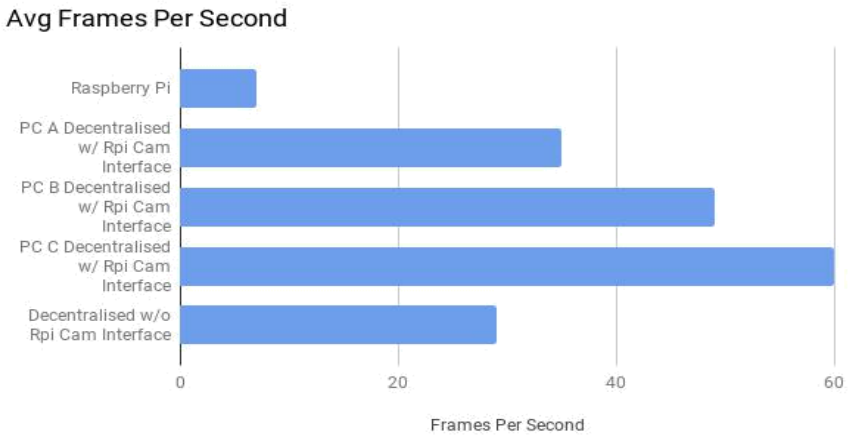


Figure 10. Frames Per Second

International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 111–136

p-ISSN 2655-8564, e-ISSN 2685-9432

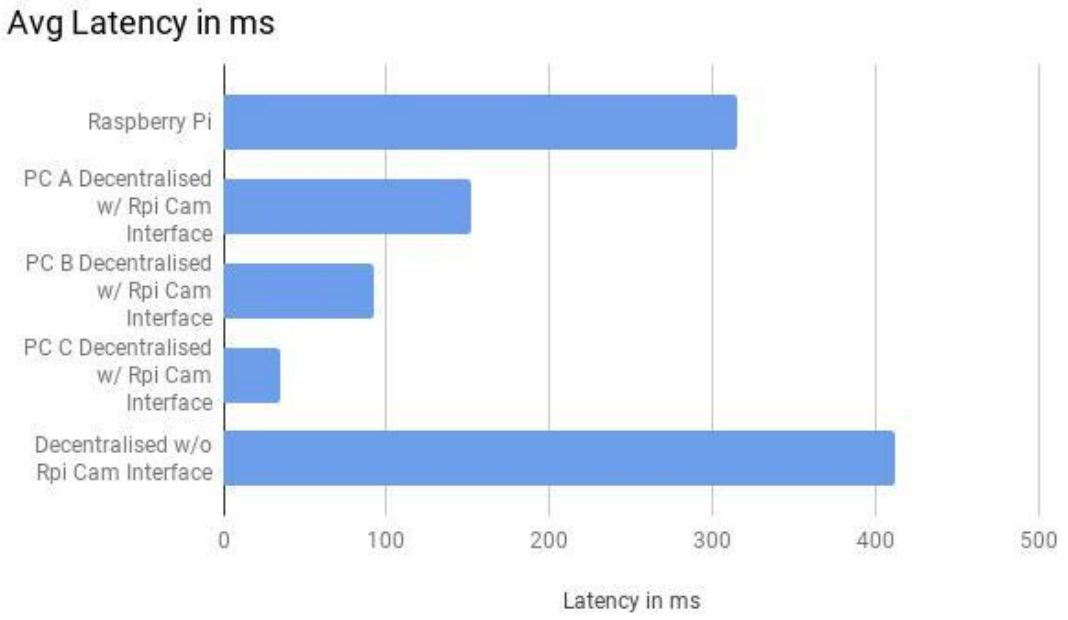


Figure 11. Latency in ms

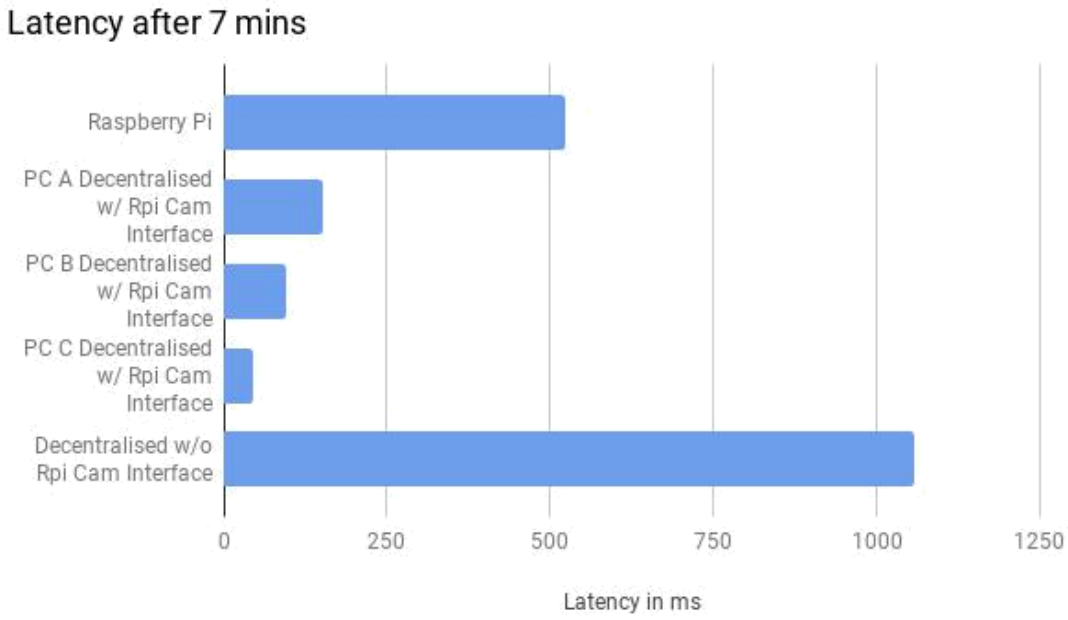


Figure 12. Latency in ms (after 7 mins)

Volume 2, Issue 2, pages 111–136

p-ISSN 2655-8564, e-ISSN 2685-9432

11 Conclusion

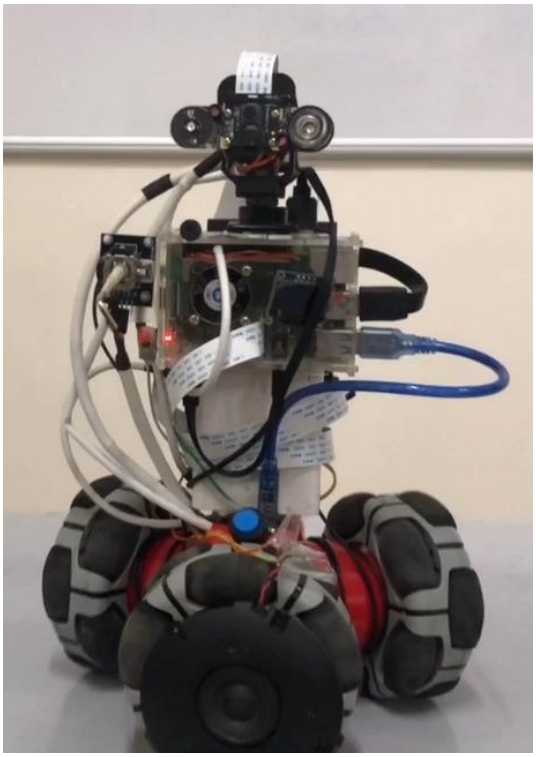


Figure 13. AI Pet Robot

We were successfully able to design the hardware of the robot that can run complex AI and CV algorithms. The Pet Robot is shown in Figure 13. The Robot was small enough that it could be carried in our hands. The architecture that we propose can be easily scaled to make one PC handle multiple Robots. This, coupled with the concept of cloud computing would significantly reduce the cost in the case of mass production as the PC was the most expensive component.

References

[1] A. Ghoshal, E. Lemos, A. Aspat, "OpenCV Image Processing for AI Pet Robot." Unpublished, 2020.

[2] Raspberry Pi Foundation, "Raspberry Pi 4 Model B Data Sheet." June 2019.

[3] Raspberry Pi Foundation, https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

Volume 2, Issue 2, pages 111–136

p-ISSN 2655-8564, e-ISSN 2685-9432

[4] Raspberry Pi Foundation, "Raspberry Pi NOIR Camera Model V2 Data Sheet." 2016.

[5] T.K. Hareendran, "Night vision camera adapter." Electro Schematics, https://www.electroschematics.com/night-vision-camera/

[6] Raspberry Pi Foundation, "Raspberry Pi NOIR Camera Model V2 Data Sheet." 2016.

[7] Matt, "Using an I2C OLED Display Module with the Raspberry Pi." Raspberrypi-spy, https://www.raspberrypi-spy.co.uk/2018/04/i2c-oled-display-module-with-raspberry-pi/, April 2018.

[8] Raspberry Pi Foundation, https://www.raspberrypi.org/downloads/raspberry-pi-os/

[9] P. Dalmaris, "Raspberry Pi: Full Stack." Section3: Getting started.

[10] S. Malik, "Install OpenCV 4 on Raspberry Pi." Learn OpenCV, https://www.learnopencv.com/install-opencv-4-on-raspberry-pi/, November 2018.

[11] Silvanmelchior, "RPi Cam Web Interface." *eLinux*, https://elinux.org/index.php?title=RPi-Cam-Web-Interface&action=edit, 2013.

International Journal of Applied Sciences and Smart Technologies

Volume 2, Issue 2, pages 111–136

p-ISSN 2655-8564, e-ISSN 2685-9432

This page intentionally left blank