

# Design and Implementation of a Machine Learning Assisted Smart Wheelchair in an IoT Environment

**Lijo Jacob Varghese**

Christian College of Engineering and Technology

**Suma Sira Jacob**

Christian College of Engineering and Technology

**CHINNASAMY SUNDAR** (✉ [sundarc007@yahoo.com](mailto:sundarc007@yahoo.com))

Christian College of Engineering and Technology <https://orcid.org/0000-0002-1743-234X>

**Jacob Raglend I**

Vellore Institute of Technology: VIT University

---

## Research Article

**Keywords:** gesture control, smart wheelchair

**Posted Date:** June 3rd, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-490123/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

# Design and Implementation of a Machine Learning assisted Smart Wheelchair in an IoT environment

Lijo Jacob Varghese  
*Department of Electrical and  
Electronics Englineering  
Christian College of  
Engineering and Technology,  
Oddanchatram, India*  
[lijojacobvarghese@gmail.com](mailto:lijojacobvarghese@gmail.com)

Suma Sara Jacob  
*Department of Computer  
Science and Englineering  
Christian College of  
Engineering and  
Technology,  
Oddanchatram, India*  
[sumasarajacob@gmail.com](mailto:sumasarajacob@gmail.com)

C.Sundar  
*Department of Computer  
Science and Englineering  
Christian College of  
Engineering and Technology,  
Oddanchatram, India*  
[Sundarc007@yahoo.com](mailto:Sundarc007@yahoo.com).

I.Jacob Raglend.  
*School of Electrical  
Engineering (SELECT),  
Vellore Institute of  
Technology,  
Vellore, India*  
[jacobraglend@rediffmail.com](mailto:jacobraglend@rediffmail.com)

**Abstract**—Gesture controlled wheelchair is designed and intended to perform specific job functions combining software interfacing with the hardware along with the embedded electronics, through hand controlled gestures. In this paper, a novel multiple control option-based wheelchair is proposed to cater to the needs of handicapped patients and senior citizens. This work is aimed at designing, integrating, coding, interfacing and testing a fully motorized, gesture controlled wheelchair which involves the physical implementation of the hardware, the software interfacing and testing. This Smart wheelchair is furnished with sensors, interactive cameras controlled by a processor unit that can control the various tasks it is designed to perform.

The proposed Smart Wheelchair utilizes micro-electro-mechanical-system (MEMS) accelerometers to recognize various hand gestures performing a specific task. It is an accelerometer-based system which controls a robotic arm wirelessly using a small, low-cost, 3-axis degrees of freedom accelerometer through RF signals. The gestures are also trained by machine learning to identify the training patterns by choosing the appropriate mode selection. The hand gestures are thus based on motion sensor and pattern recognition. Apart from these hand gestures, the smart wheelchair can also be controlled by an android mobile app through a smartphone or by a hand held device that can navigate the smart wheelchair in the forward, backward, right, left directions and also perform a stop operation. The Smart Wheelchair also detects an obstruction present on the way of navigation and is operated in accordance with the command. An interactive camera equipped with a microphone and speaker is meant for communication for the elderly for sending and receiving voice commands remotely. The camera can be accessed through an IoT environment which can be communicated on remote IP addresses on smartphones and personal computers. There are two Arduino Uno Rev3 Controller modules programmed to control the gestures of the gyroscopic accelerometer and other control methods based on a mode selection switch held by the user. The proposed methodology is also realized by Proteus, a simulation tool to check its practical credibility. The proposed circuitry is physically realized as a Smart Wheelchair prototype implemented with the above multi control which could be modified as any physical structure platform having two wheels.

**Keywords**— gesture control, smart wheelchair

## I. INTRODUCTION

Studies show that 2 out of 10 adults globally have a disability and about 30.6 million people have difficulty walking. Physical disabilities are a result of temporary or permanent injury leaving the people either handicapped or paralyzed partially. About 8.5 percent of people worldwide (617 million) are aged 65 in which about 35% face difficulty in walking. In both these cases people are bound to be dependent on a wheel chair for their restricted movement which is impossible to be operated independently [1]. Using a wheelchair to be operated for their needs makes them a liability to others, as they are dependent on another person for their movement. Researches were made to design a smart wheel chair, a self-reliable entity for the disabled people as well as the elderly to be self-operated at any point of time. It becomes difficult for a person who is paralyzed to operate this wheelchair through the hard buttons or with the aid of a joystick.

A Human-Smart Wheelchair Interaction Model plays an essential role in designing a smart wheelchair. Utilizing a reinforcement learning technique [2] ensures efficient interaction between user and wheelchair where sensor feedbacks under various environment are used to control signals for the wheel motors in the navigation of the mobile robotic base. The human-machine interface could be established by means of active gestures through various means of non-verbal body actions. This includes variation in light, sound, movement in body parts that includes hand, head, and chin, even the movement caused due to inhalation and exhalation or even brain signals. The appropriate gesture could be measured through optical, acoustic, tactile and other motion sensors, cameras, infrared radiations, etc. which are then communicated for decision making for the movement control of the wheelchair. There are several disadvantages in the existing systems such as delayed responses and complicated hardware.

The paper is organized under the following sections. The literature review associated with the Smart Wheelchair is discussed under section II, the overview of the proposed methodology of the smart wheelchair is summarized under Section III. Section IV presents briefly about the implementation details of the wheelchair, while the machine learning aspects of the gesture control is summarized under Section V. Results and Hardware Prototype are presented

under Section VI and finally, the evolved conclusion of the work along with the future scope.

## II. LITERATURE REVIEW

People inheriting mobility disabilities or diseases rely on electric-powered wheel chairs for their mobility. A manual wheelchair operated by an electric motor is termed an electric-powered wheelchair. This is primarily used by the disabled who find it difficult in utilizing a manual wheel chair which involves a lot of muscle work in steering this, which is not always possible for the elderly and the disabled who do not have the required visual insight and motor skill leading to dependent mobility. A few literature as to the different types of human machine interface and the types of navigation system are taken for discussion.

To overcome these difficulties of operating a power wheel chairs, it led researchers to utilize technologies for the innovative development of mobile robots in creating a smart wheelchair [3], [4]. A Smart Wheelchair (SW) essentially is made up of a basic power wheel chair base in addition to a computing control with a combination of the needful, or a seat mounted on a mobile robot [5].

Since some people with disabilities cannot use a traditional joystick to navigate their power wheel chairs, they use alternative control systems like head joysticks [6], where the movement of the head is sensed as different controls of navigation. The next in the list is the chin joysticks [7] in which the steering control of the wheel chair is based on the movement of chin. Tongue control [8] is based on a device placed intra-orally consisting of several inductive sensory coils, connected externally to a radio chip controlled by a microcontroller and powered by a battery. The motion control is based on the change in inductance of the coil when the disabled user touches the sensor through his ferromagnetic pierced tongue. Sip and Puff [9] is a type of navigation control which utilizes inhalation and exhalation of the user on a tube or a wand to send motion control commands. Researchers also have evolved with a Face/gaze control [10], where intelligent controls are adopted on the basis of the gaze or tracking the face direction which is focused continuously for navigation of the wheelchair. Direction control also utilizes electrical impulses of the brain which is interpreted through the electrodes attached to the scalp, which is termed as the Electroencephalography or the thought control [11]. Voice assisted navigation control [12] using voice commands as inputs could be employed as control methodology for Smart Wheelchairs. This is based on recognition algorithms which already trains voice commands and is stored in the control database, which gets executed in accordance to the appropriate commands. Human Machine Interface also could be done by a motion control system based on hand gesture [13]. This system employs accelerometer that could effectively translate hand gestures to motion commands controlling the navigation of the Smart Wheelchair.

There are three methods based on assistance levels for navigation which are shared control, semi-autonomous and autonomous control. The assistance of the navigation increases as it is automated in autonomous controls. In a semi-autonomous control, the system receives control from user for short distance destinations whereas the user is given the liberty of motion to plan the route in a shared control mode. In an improvised version of the latter, an adaptive

shared control system [14] proposes a system, where the involvement of the user determines the control assistance which analyses the users' capability to continue with the appropriate action in line with its safety measures.

Semi-autonomous systems are adapted for users who are visually challenged. It coordinates a control, based on virtual reality for the needful instructions and control commands for navigations. The user specifies the final target location, while the wheel chair is steered to the specific destination with the complete navigation control up to 1 kilometer, influenced by the system, with the necessary mapping and position monitoring arrangements is termed as an autonomous system. As autonomous and semi-autonomous systems employ many smart sensors [15] like infrared (IR) or Light Detection and Ranging (LIDAR), Sound Navigation and Ranging (SONAR) for avoiding obstacles and other position monitoring. It requires a controller with good computing power.

This paper is based on a low cost, machine learning assisted, hand gesture controlled smart wheel chair which is designed to be operated under the shared control mode for navigation suitable for the elderly and the disabled.

## III. PROPOSED METHODOLOGY

The concept of independent operation by means of a smart wheelchair could be realized by enabling the commands of a wheelchair using the hand gestures of the user for navigation in the specified directions which could be accomplished in two ways. Firstly, the gesture could be fed to the transmitter by adopting a robotic arm, utilizing a micro-electromechanical system (MEMS) accelerometers which is a miniscule device which has both mechanical and electronic components. This carries out the positional changes in acceleration by the different gestures for performing a specific task. It is basically an accelerometer-based system which controls a robotic arm wirelessly using a, small and low-cost, 3-axis degrees of freedoms (DOF's) accelerometer through RF signals. The variation in the levels of voltages based on the acceleration, which is an analog signal, is fed to the microcontroller based Arduino controller which consists of Analog to Digital Converters (ADC) ports converting them to an Encoder that converts the parallel data into the required serial data to be transmitted by the Radio Frequency (RF) transmitter module. The RF receiver module in receipt of the serial data from the transmitter, which is again converted to the needful parallel data through a Decoder. The data is transferred to the Arduino module in the receiver end. The Arduino module processes the data and provides the command information to the motor driver that controls the navigation of the Smart Wheelchair. Secondly, as a backup to the input feed, utilizing a Convolution Neural Network based machine learning, the same gestures are pre-trained and fed to a controller. Based on the training, these gesture patterns are recognized and the appropriate movement control is initiated. The main disadvantage of an accelerometer is that only a few precise gestures could be utilized with reasonable accuracy. On the contrary, in machine learning, any number of custom gestures could be taken, and with training these gestures, the system could recognize any of them at a faster rate with reasonable error tolerance and a swifter response.

Apart from the gesture controls, the wheelchair in this system can be controlled by using a smart phone for

navigation control commands that include forward, backward, stop, turn left and turn right by means of an Android Application. A provision is also included for the person to interact with a keyboard interface for taking out these navigational commands. These commands are responded accordingly by the robotic wheelchair when the commands are issued.

The system is equipped with an interactive camera module, which is basically a CCTV camera with a 360-degree field of view consisting of a microphone and a speaker. The camera could be tilted to any angle for viewing by the user with the aid of an Android application custom designed by the manufacturer. These gesture controls could be used by the disabled users to steer the wheelchair to the needful location, right from their current location. The user could also employ this wheelchair to navigate to a different location and could access the view of the same with the aid of the attached camera.

The wheelchair could be navigated to interact with other persons even without the user to reach out to the other person. The receiver is also equipped with a Wi-Fi module which enables the access of the camera and ensures communication. The camera could be accessed through an IoT environment which could be communicated on remote IP addresses on smartphones and Personal Computers. This enables the disabled person to communicate with other persons to listen from the others through this camera module made accessible to the user. There are two Arduino Uno Rev3 Controller modules which is programmed based on the control gestures of the gyroscopic accelerometer or other control methods based on a mode selection switch held by the user for steering the wheelchair for various types of the assigned movement. The system is also empowered by an ultrasonic based obstacle monitoring system indicating the sight of obstacle at about a distance of a feet. Liquid Crystal Display (LCD) display units are provided both at the transmitter as well as the in the receiver module to inform the user regarding the mode selection as well as the command of the gesture as to what is the appropriate motion control to the Smart Wheelchair.

The disabled user would be able to control the motion of the smart wheel chair for about 700 meters through the gesture commands and could be navigated accordingly. Thus, the system provides the needful navigation and an interactive control independent of others, while the disabled or the elderly also could be monitored under the watchful eye in an IoT environment by the immediate caretaker which could be accessed on the remote IP addresses through smartphones and Personal Computers.

#### IV. IMPLEMENTATION DETAILS OF THE WHEELCHAIR

The Smart Wheelchair control proposed in this paper is based on hand gesture inputs of the user for navigation in the desired directions. Here, a hand gesture input by the user results in the motion control command for the wheelchair. The implementation details pertaining to each of the modules involved is discussed in the following section.

##### A. MEMS accelerometer module

The implementation of the same could be done by using a gesture sensor called as MEMS accelerometer which could be operated in 3 degrees of freedom. The MEMS sensors,

when switched on, measures the static accelerations like gravity forces in all the three perpendicular directions of X, Y and Z [16] that enables them to be used as tilt sensors. A low cost, monolithic IC, ADXL335 is used as the MEMS accelerometer in the proposed system for the acceleration measurement which measures the acceleration in all the 3 directions. The output signals of this module are analog voltages that are proportional to the acceleration. This module measures acceleration in g unit within a range of  $\pm 3g$  in the x, y and z axis and 'g' refers to the acceleration, only due to cause of gravity or the g force. The maximum voltage level at 0g is 1.65V and sensitivity scale factor of 330mV/g. Therefore, the range of the analog voltage output will vary between 0V to 3.3V.

The deflection of the silicon structure suspended by several polysilicon springs is measured by means a differential capacitor which gets unbalanced due the deflection of the silicon structure caused by the acceleration through various hand gestures. These changes produced by the differential capacitor, results in producing an output square wave whose amplitude is proportional to acceleration. The direction of the acceleration could be detected by utilizing phase-sensitive demodulation techniques to the signal thus obtained. The MEMS accelerometer is mounted on a glove, to be worn by the user, which mimic a Robotic Arm for giving the hand gestures. Five hand gestures namely Upward, Downward, Horizontal, Left and Right are meant to represent five navigation commands to the wheel chair are chosen as shown in Fig.1.

##### B. Arduino Module

The analog signal has to be converted to digital signals before communicating it to the Radio Frequency Communication System [17] which requires an ADC. This could be achieved by feeding these signals to an Arduino controller which houses ADC ports. Arduino Uno Rev3 controller is used in the proposed system. Arduino Uno, which is based on microcontroller ATmega328P which could be powered by an USB cable or by a 9-volt dc supply, and accepts voltages ranging between 7 and 20 volts. This has 14 digital input or output pins, out of which 6 pins could be utilized as PWM outputs and 6 analog inputs. In an Arduino Uno, Pins A0-A5 are capable of reading analog voltages and converting them digitally. The ADC operated on an Arduino has 10-bit resolution, which is capable of representing the analog voltage inputs by 1,024 digital levels in the output. Coding can be done with the Arduino in Integrated Development Environment (IDE) that uses a text editor for writing programs. The Arduino language is basically a set of C/C++ functions that could be called from the programs.

The analog outputs of the accelerometer are fed to the analog ports A0, A1 and A2 of the Arduino which takes up the 3-dimensional (XYZ) Co-ordinate space. As the palm is moved to indicate a particular gesture, the voltage changes accordingly in the X, Y & Z sensing units of the accelerometer as the physical changes in the palm movement causes appropriate change in voltage.

The Arduino is programmed such that the accelerometer output voltage values are converted to a digital value in a range between 0-1024, (as the ADC has a 10-bit resolution) of the voltage reading corresponding to the sensing of the position of the palm movement in the XYZ Co-ordinate space.

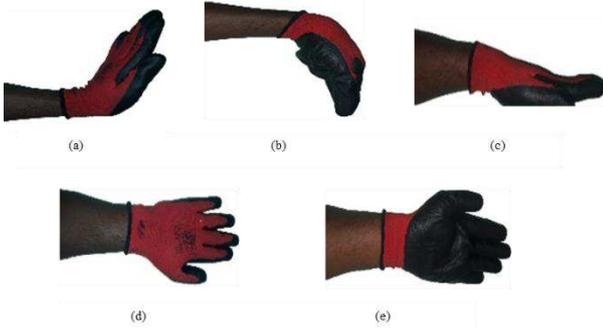


Fig. 1. Hand Gestures for (a) backward, (b) forward, (c) stop, (d) left and (e) right

The values of acceleration in g unit along X, Y and Z axis could be calculated by,

$$A_{out} = \frac{(((ADC \text{ value} * V_{ref}) / 1024) - \text{Voltage level at } 0 \text{ g})}{\text{Sensitivity Scale Factor}} \quad (1)$$

For getting the value of acceleration in g unit along X axis

$$A_{Xout} = \frac{((X \text{ axis ADC value} * V_{ref}) / 1024) - 1.65}{0.33} \quad (2)$$

Similarly, the values of acceleration along Y axis and Z axis,  $A_{Yout}$  and  $A_{Zout}$  also could be found out. The function `analogRead()` could be utilized as, `int x = analogRead(memsPin[0]);` which means that variable x has a digital voltage value in range 0 -1024 that would change as the user moves his palm along the X axis. Similarly, the values along the Y and Z axis also could be retrieved as a digital value.

The measurement range of the accelerometer varies between  $\pm 3$  g. Therefore the values of acceleration in g along the X, Y & Z Co-ordinate space in terms of analog voltage corresponding to the position of the palm along different axis will result in a range anywhere between an analog voltage of 0V to 3.3V which when converted digitally varies between 0 to 1024 (where 0 Volts refers to a value 0 and 3.3Volts refers to a value 1024). So, as the palm is moved the voltage changes accordingly in the X, Y and Z sensing units of the accelerometer. Based on the different hand gestures, the values of the digital output in along each axis of a coordinate space could be coded as follows:

```
const char cmd_name[8][16] = {
  "Forward ", //1
  "Reverse ", //2
  "Left  ", //3
  "Right ", //4
  "Stop  ", //5 };
const int ref[8][3] = {
  270, 330, 340, //1
  390, 330, 375, //2
  330, 400, 340, //3
```

```
340, 266, 350, //4
```

```
320, 350, 408, //5};
```

```
if ((abs(ref[i][0] - x) < tolerance) && (abs(ref[i][1] - y) <
tolerance) && (abs(ref[i][2] - z) < tolerance))
```

As the Arduino code is running in a loop, whenever the pre-set values of X, Y & Z axis of the five values of the gesture preloaded are reached within the value of the tolerance, the Arduino at the receiver end will detect the particular gesture and communicates it to the motor driver. But there would be a slight accuracy issue. So, as the values of X, Y & Z approach these absolute values, the gesture is identified as it does not require exact values.

To produce these five motion control commands, the coding for the selected gesture is done by taking these digital values in terms of angular position of the palm movement which are chosen as follow. To choose a backward movement of the chair, an upward movement of the palm, creating an approximate angle between  $+20^\circ$  to  $+90^\circ$  with Y axis is chosen as a gesture. Similarly, to move the chair forward, a similar gesture in the reverse, i.e., the palm movement creating an approximate angle between  $-20^\circ$  to  $-90^\circ$  with Y axis is chosen as a gesture. When the palm is held parallel with the horizontal, it brings the wheelchair to a stop position.

To enable the motion of the wheel chair in the left direction causing an anti-clockwise rotation of the wheel motor, a gesture is chosen where the palm is positioned to produce an approximate angle between  $-20^\circ$  to  $-90^\circ$  with X axis, while a hand gesture with the palm position at an approximate angle between  $+20^\circ$  to  $+90^\circ$  with X axis, produces a clockwise rotation of the wheel motor, leading the wheelchair to the right. Table I demonstrates the relationship between the gestures, palm position and the appropriate motion command to the wheelchair as per the gesture.

### C. Radio Frequency Communication System

The gesture sensors produce analog output voltages proportional to the acceleration of the gesture. The changes in the positional acceleration is measured for different gestures and compared and the specific motion command is given to the drive motor. These voltages obtained from the accelerometer could be communicated to the wheelchair to process these commands through a Radio Frequency communication system. This system consists of RF transmitter module and RF receiver module constituting of an encoder and a decoder respectively.

The RF communication module of the proposed system is implemented as follows: The digital output from the ADC of the Arduino is fed to the encoder before its transmission by the RF transmitter. In this system RF transmitter IC TWS-434 along with its encoder chip HT-12E forms the transmitter circuit while the receiver module is constituted by receiver IC RWS-434 along with its decoder chip IC HT-12D. The transmitter has a supply voltage of 3-12 V dc and an output of up to 32mW. Both the transmitter and the receiver operate at 433.92 MHz and the communication between could be established within a range of 200 feet indoor, enabling transmission even through most of the walls and more than 500 feet outdoors. The receiver works with a supply voltage of 5 V dc and works with a sensitivity of 3uV.

TABLE I. GESTURE AND COMMAND INTERPRETATION

The encoder and decoder chips HT-12E and HT-12D work in conjunction with TWS-434 and RWS-434 respectively produce the desired output. Fig.2 shows the basic blocks involved in the RF communication module along with other controls through the Arduino. The encoder used, HT-12E is a 212 series encoder that could transmit 12 bits of parallel data serially. The 12-bit parallel data is converted in to a serial data output by dividing it into a 4-bit data and an 8-bit address that could be transmitted with the aid of a RF transmitter for the final relay. The address bits provide an 8-bit security code for the 4-bit data to be transmitted and enable multiple receivers to receive the data through a single transmission. The antenna connected via, the receiver IC, RWS-434, receives the data transmitted by the above transmitter module and sends it to IC HT-12D for the necessary decoding of the serial data. The decoder converts the received serial data through RF receiver to 12-bit parallel data. These 12-bit parallel data is decoded by dividing it into 8-bit address and the required 4-bit data, which is produced eventually at the respective outputs for driving the connected loads. Thus, the transmitter module complements the receiver module in its operation to exchange 4-bits of digital data for controlling the external loads. The chip RWS-434's antenna receives the data transmitted by the above transmitter module and sends it to the IC HT-12D for the necessary decoding of the 4-bit data which ultimately is decoded and produced at the respective outputs for driving the connected loads.

#### D. Power supply Module

All the devices are to be powered by a dc voltage supply which is either a 12V or 5V. An AC voltage of 230V is stepped down to 15V by using a step-down transformer followed by a bridge rectifier converting it from ac to dc, then utilizing a Filter capacitor for filtering the ac ripples. A regulated power of 5V or 12 V by feeding this filter output to a fixed voltage regulator ICs LM7805 and LM7812 respectively. The 12V dc supply is given to the Arduino and all other devices are given 5V each. Batteries of voltage rating 12V dc is given as power supply for Arduino and an exclusive 6V for powering the camera module.

#### E. LCD Display Module

The LCD module used in this system is JHD162A display controller, which is a monochrome LCD module that could be used with Arduino Uno to display output values and messages to the user. This is provided both at the Transmitter module as well as in the Receiver module to display messages regarding the mode selection and the status of the gestures and about the obstacle sensed during navigation. The JHD162A module based on the HD44780 driver from Hitachi, has 16 pins which could be operated in 4-bit mode (using only 4 data lines) or 8-bit mode (using all 8 data lines). This is basically a 16x2 LCD Module that can be used

Gesture	Position of the Palm	Command
Upward	Palm creates ( $\cong$ ) $+20^\circ$ to $+90^\circ$ with Y axis	Chair moves backward
Downward	Palm creates ( $\cong$ ) $-20^\circ$ to $-90^\circ$ with Y axis	Chair moves forward
Horizontal	Palm is parallel with horizontal	Chair stops
Left	Palm creates ( $\cong$ ) $-20^\circ$ to $-90^\circ$ with X axis	Chair moves left
Right	Palm creates ( $\cong$ ) $+20^\circ$ to $+90^\circ$ with X axis	Chair moves right

to display 32 ASCII characters in 2 lines (16 characters in 1 line).

#### F. Camera Module

The primary Camera is an interactive camera that could help the user to navigate the wheel chair through different hand gestures for various movement options. The camera is an Infrared enabled camera that could work even at the night time or at very low light intensity. Thus, it ensures an unperturbed usage for the user irrespective of day or night. The other important aspect of the camera is the interactive phase, supporting two-way audio for communication as the module includes a built in Microphone and speaker enabling several people for communication at the same time. This could be utilized by the user to navigate the chair indoors also to check the various prevailing conditions and also to answer the doorbell without requiring the user to move. The user can make use of a mobile application both for controlling the camera which is capable of rotating 360 degrees and also for two-way audio communication. Fig.3 illustrates the user interface when the user accesses the camera module for camera rotating adjustments and for audio communication.

A BUSH PLUS Camera is a Wi-Fi enabled camera integrated with remote camera configuration management, remote viewing, and remote playback. The camera has a 2-megapixel resolution, with clear imaging features for monitoring. Though the camera has a video recording feature that requires a hard drive but this is not utilized in this system as it is employed only for viewing purpose. It requires charging for operation and provided with a 6V battery exclusively for meeting out this purpose. The camera can also support capturing images in night vision. The module could be accessed for remote monitoring through open source surveillance software application iSpy for IOS system and Agent video surveillance for Android system.

A secondary camera is also used to import the gestures as a secondary input for a backup system to the accelerometer module is used to control the navigation of the wheelchair assisted by Machine learning, which is discussed in detail in the next section. The secondary camera which is referred as PiCam, is the camera module of the Raspberry Pi which is used in Machine learning. The PiCam is available as an additional external module that could be interfaced directly with a Raspberry Pi used in the proposed system.

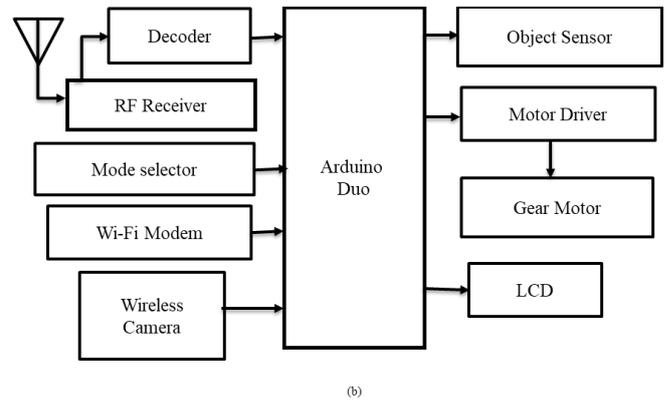
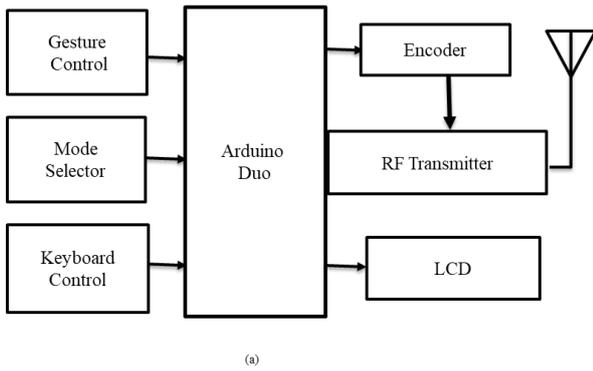


Fig. 2. (a) Transmitter end Module (b) Receiver end Module



Fig. 3. Camera module accessibility using the mobile App (a) Camera Navigation control interface (b) Audio communication interface

### G. Wi-Fi Module

A Wi-Fi module is required for the system for serving three purposes. Firstly, the user could control the motion of the wheel chair remotely by using a smartphone. This could be done by connecting the wheelchair through the Wi-Fi by using an android application. An application called as IoT Robot is utilized to access the control through an IP address over the Wi-Fi. The navigation commands forward, backward, stop, left and right could be used to steer the wheelchair in the desired position. Fig. 4 shows the Mobile interface for Navigation through Wi-Fi.

Secondly, the user could access the interactive camera module through Wi-Fi and could make necessary communication over the mobile through the camera application. The user could also make use of the necessary access of the 360-degree rotation of the camera through the application over Wi-Fi access. Thirdly, given the access, any number of users could be able to communicate through the specific IP address for accessing the video of the user through the interactive camera module. This access could be given to the caretakers or the family members of the disabled user to find the status or the well-being of the user. Thus, the system is open for access of the smart wheelchair in an IoT environment which could be accessed over the Wi-Fi.

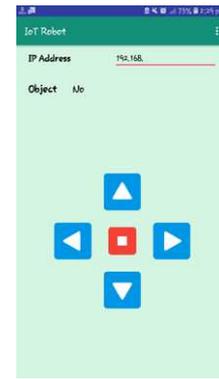


Fig. 4. Navigation of Wheelchair through Mobile interface accessed through Wi-Fi

The Wi-Fi module used here is ESP8266, which is empowered with full TCP/IP capability. The module is compatible to an Arduino module and is mostly used in embedded applications for IoT (Internet of Things) development. It has 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/ WPA2 authentication or open networks) and an integrated Micro Controller Unit capable of controlling the I/O digital pins through pseudo-code like programming language. The other features includes 16 general-purpose input/output pins, 64 KB of instruction RAM, 96 KB of data RAM, a 10-bit ADC, I<sup>2</sup>C (Inter-Integrated Circuit) and (Serial Peripheral Interface) serial communication protocol, I<sup>2</sup>S (Inter-IC Sound) interfaces with DMA (Direct Memory Access).

### H. Infra-Red Sensor Module

An infrared sensor is used as an Obstacle detector to detect the proximity of any object obstructing the motion of the wheelchair. The obstacle detector used in this system is FC-51 IR sensor which has a built-in IR transmitter and IR receiver. They transmit an infrared signal and detects the reflected IR energy at the infrared receiver for detecting the presence of any obstacle. The module could be adjusted for the detection range with an on board potentiometer. It produces about 2.8V at 15cm to 0.4V at 150cm with a supply voltage between 4.5 and 5.5 V dc. IR transmitters are normal LED (light emitting diode) that radiates infrared radiations. IR receivers are either photodiodes or phototransistors that detects the radiation emitted from an IR transmitter and constantly sends digital data in the form of 0 or 1 to V<sub>out</sub> pin of the sensor. The IR sensor sends a 0 state when it sights the presence of an obstacle and sends 1 state in case of an absence of object.

### I. Keypad Module

This is an addition control available for the user when the wheelchair is used. Here, pushbuttons indicating navigation commands forward, backward, stop, left and right are provided for managing the steering of the wheelchair, without using the smartphones. This module is intentionally included in the system to help users who are less aware of smart phones or its technical usability. The Arduino controls the operation as per the control received is transmitted and communicated to the Motor driver for the execution of the command.

### J. Motor Driver

The L293D is a 16-pin dual-channel H-Bridge motor driver IC which is capable of driving a pair of DC motors

simultaneously that makes it ideal for developing two-wheel robot platforms. As the output voltage of the Arduino is insufficient to drive motors of higher voltage rating, a motor driver is required to provide this requisite voltage required for the motors to be operated. The L293D motor driver has two power input pins Vcc1, for driving the internal logic circuitry which should be 5V and Vcc2 for powering the H-Bridge for driving the motors within a range between 4.5V to 36V and to provide a bidirectional drive currents of up to 600 mA. The motor driver is equipped with two types of control pins that allow the user to control the speed and changing the direction of the DC motors simultaneously.

The forward or backward direction of the motor could be controlled by controlling the switches of the H-Bridge circuit. ENA and ENB are the speed control enable pins that

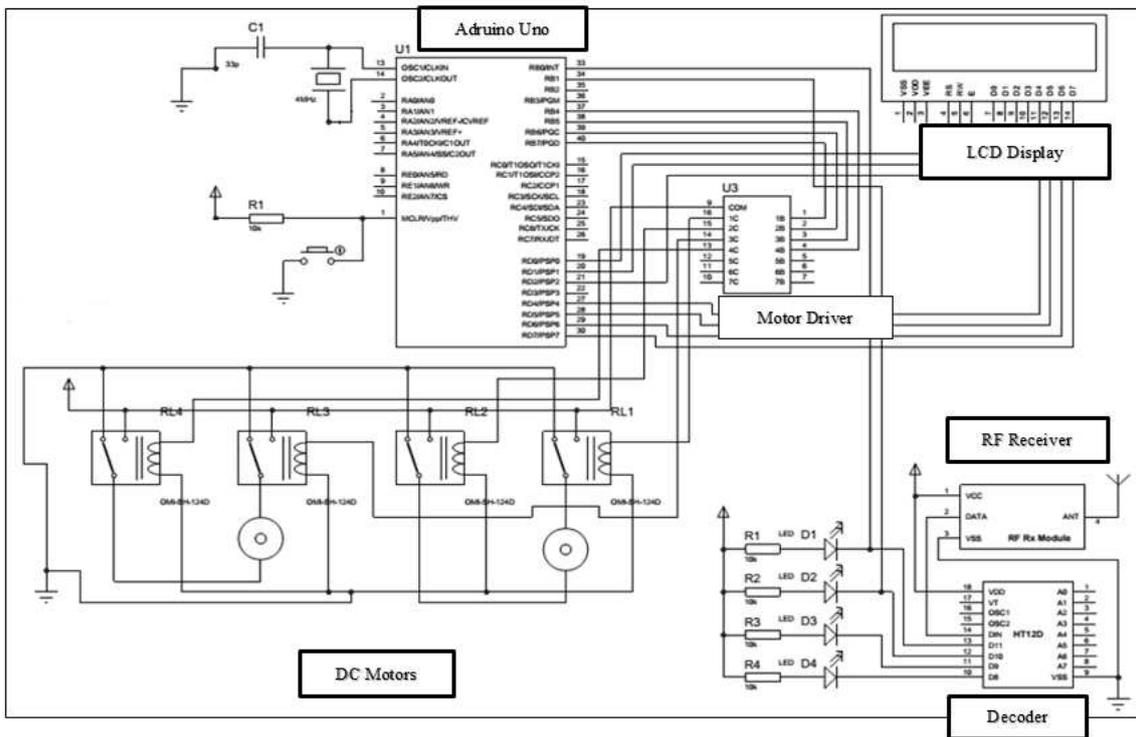


Fig. 5. Circuit diagram of the Receiver Module

are utilized to turn ON, OFF and control speed of motor A and motor B respectively. Suitable selection of these pins enables the wheel control allowing them to turn right or left and also for controlling the speed of the motors.

### K. DC Motor

A set of dc motors is used to give the electric locomotion to the wheelchair. The rating of the motor is 12V, 1.7A, 20.4W. The speed of the motor as well as its direction is controlled by using the motor driver. The PWM control ensures the change in speed while the change in direction is made possible by the H-Bridge, which effectively reverses the polarity of the supply to change the direction of the motor. For the movement of the wheelchair towards the left, the left side motor is stopped and only the motor in the right side is powered driving the wheelchair in an anticlockwise angular movement. The wheelchair movement could be steered towards the right, by stopping the right sided motor and energizing the left side motor which drives the wheelchair towards the right imposing a clockwise angular

movement. The circuit diagram of the receiver module is as shown in Fig. 5, depicting the connection between various integrated modules.

## V. GESTURE CONTROL BY MACHINE LEARNING

A wheelchair cannot be made smart by just assembling hardware, but by integrating exclusive computer algorithms that provide artificial intelligence required for split-second decision making. The gesture controls are assisted by a machine learning paradigm for taking up gesture inputs as commands for motion control of the wheelchair. This is a backup feature for the accelerometer based gesture recognition, which has accuracy disadvantages, when compared to the AI based gesture recognition technique. The number of custom gestures that could be processed in an accelerometer-based gesture control is lesser compared to its counterpart, which can take up any number of custom gestures. The processing time taken for identification of the gestures patterns is more in accelerometer-based control, while the response time is very much swifter in AI based

control. Also, the complexity in control increases in an accelerometer control when the number of custom gestures increases but it is less complex in an AI based control.

Using the mode selection module, the type of control could be selected for giving the input commands. A second camera which is used for this purpose is a PiCam. It is placed along with a transmitter module near the user to pick up the palm movements to detect gestures clearly with its 8MP resolution. At the transmitter module, the same transmitter side Arduino could be used for implementing, but the requirement is that the Arduino must be connected to a laptop or a mini PC. Implementing the machine learning aspects of a neural network is best suited for detecting custom gestures. As the training of these gesture patterns is

done with neural networks, it is required to have a module with a good computing power like a Graphics Processing Unit (GPU) or Central Processing Unit (CPU) for its hardware implementation. It could be effectively done using a Raspberry Pi. Fig.6 shows the machine learning Interface using a Raspberry Pi, which takes up custom hand gestures and impart machine learning through gesture pattern training algorithms and using Artificial Intelligent techniques could recognize patterns through classifiers. The resulting feed is sent to the receiver module for motor drive control. Raspberry Pi is a cheap, single-board computer (SBC) that runs Linux, with an ability to provide a set of general-purpose input/output (GPIO) pins that could be utilized for controlling electronic components for physical computing and to implement applications over the IoT environment.

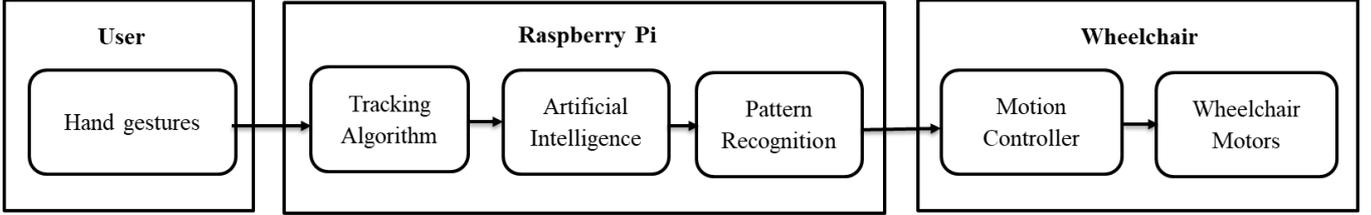


Fig. 6. Machine Learning Interface

The Raspberry Pi 3 Model B+ is used in this proposed system which is embedded with a 64-bit quad core processor with 1.4 GHz, dual-band 2.4 GHz and 5 GHz wireless LAN, Bluetooth, 4.2/BLE, faster Ethernet, and Power over Ethernet (PoE) capability. This has a Gigabit Ethernet over USB 2.0 with maximum throughput of 300 Mbps and 4 USB- 2.0 ports, a memory of 1GB SDRAM. This can be powered by supplying 5V/ 2.5A DC through a micro USB connector or a 5V DC through GPIO header. For a custom gesture recognition feature requirement, using the Convolutional Neural Network (CNN) [18], [19] could be the best.

#### A. Convolutional Neural Networks

Convolutional Neural Networks are built with a certain number of neurons where each one has learnable weights and biases. Every neuron acquires images as inputs. They dot product these images that sometimes results in non-linearity while a differentiable score function is given as output where pixels of the raw image are converted to class scores. Convolution is generally used in Digital Signal Processing which combines two time functions  $f(t)$  that can be expressed mathematically in a discrete domain of one variable as,

$$y[n] = x[n] * h[n] = \sum_k x[k] * h[n - k] \quad (3)$$

Where,  $k \in [-\infty, +\infty]$

For a discrete domain of two variables:

$$(f * g)(x, y) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f(u, v).g(x - u, y - v) \quad (4)$$

Hence, it could be found that Convolution is the sum of point wise products of function values, subject to traversal. Although the operation carried out on image inputs with CNNs is called convolution, actually it a slightly different

because one of the inputs is time-reversed and is called cross-correlation.

$$y[n] = x[n] * h[n] = \sum_k x[k] * h[n + k] \quad (5)$$

Where,  $k \in [-\infty, +\infty]$

The accurate method for convolving a Kernel (16 x 16) with input volume (RGB image of size 256 x 256 x 3) requires taking into account the patches from the input image of size equivalent to the kernel and computing the dot product between the values found in the patch and the values in the kernel matrix. The convolved value obtained by summing the resultant terms from the dot product forms a single entry in the activation matrix.

Features are patterns in the input data that helps in image analysis through which the CNN learns. When some patterns arise repeatedly from the data, they are termed as a feature by the CNN. In this work, hand gestures are taken as features. A filter is an operator applied to input which converts the data encoded in pixels. Usually, a kernel is a matrix of real value entries which is small in size compared to the image's input dimensions which when convolved with the input volume gives a map that indicates the 'activated' regions, which have the features particular to the kernel. The values of the kernel matrix are not permanent and keep changing with learning iteration over the training set. This shows that the network is learning to detect important regions for recognizing features in the input volume.

A Kernel (16 x 16) with the input volume (RGB image of size 256 x 256 x 3) takes 16 x 16 patches from the input image and calculates the dot product between the values found in the patch and the values in the kernel matrix. The outputs from the dot product are summed and this convolved value makes up a single entry in the activation matrix. The patch selection is then slid to the right by a

specific value known as the 'stride' value. This is repeated until the processing of the whole input volume is completed for all color channels and then the enumerated values of the activation matrix are divided by the sum of values in the kernel matrix for normalization. This is to ensure similar intensity range among input volume and the output volume.

Fig.7- Fig.12 are illustrations associated with the different conceptual operators involved in the computation of CNN.

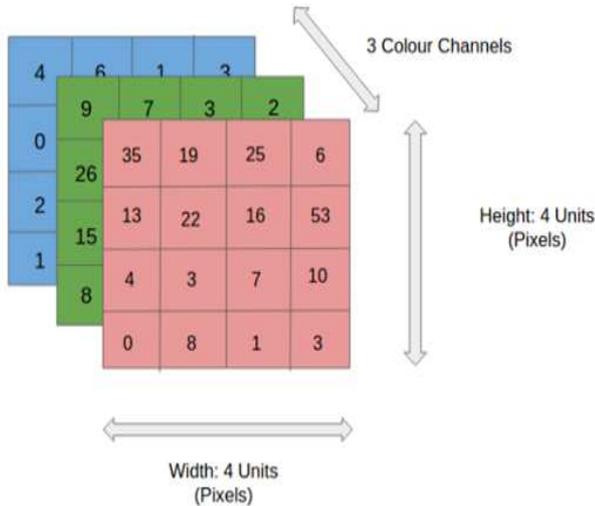


Fig. 7. The cross-section of an input volume (RGB image) of size comprised of 3 Colour channel matrices of the input image

stride length (S). A classic CNN has multi layered architecture and each layer does a particular role converting its input to a form that is useful for the next layer. Usually Neural networks use three main types of layers to build ConvNet architectures that include Convolutional Layer, Pooling Layer and Fully-Connected Layer. The raw pixels of the image are converted into final class cores, layer by layer where only a few layers consist of parameters. The CONV/FC layers perform transformations that are a function of both activations in the input volume and the parameters (the weights and biases of the neurons) trained with gradient descent. As a result the ConvNet computes class scores that are consistent with the training set labels for each image.

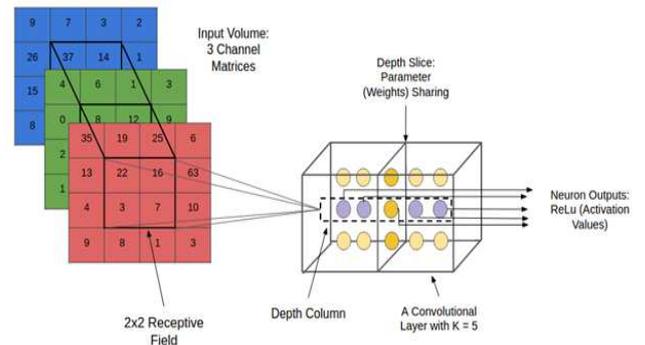


Fig. 9. The Concept of Receptive Field

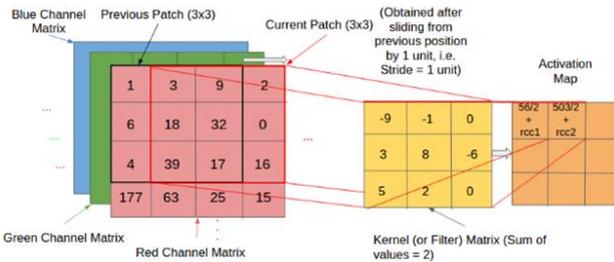


Fig. 8. The convolution value calculation by the dot product of the values in the Kernel and their channel matrices

Receptive field is a two-dimensional region (size 5x5 units) expanding to the full depth of the input (5x5x3 for a three color channel input). The pixels in the region are totally connected to the input layer of neural network as it is not possible to connect the entire neurons with all possible regions of the input volume. It is due to the fact that the connection will cause excessively large number of weights and high computational complexity. The network layer cross-sections called 'depth columns' operate over these small regions and produce the activation map.

Zero-padding is the method of symmetrically adding zeroes to the input matrix. This is a general improvement to adjust the input size as per our specification. When the input volume dimensions have to be preserved in the output volume, zero-padding is employed in designing the CNN layers. The properties of the structure of layers and neurons in CNN like spatial arrangement and receptive field values that uniquely specify the layers are called hyper parameters, which uniquely specify layers and they include field (R), zero-padding (P), the input volume dimensions (Width x Height x Depth, or W x H x D) and

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Fig. 10. A zero-padded 4 x 4 matrix becoming a 6 x 6 matrix

The RELU/POOL layers implement a fixed function. Gradient descent is used to train the CONV/FC layer parameters thereby making the training set labels for each image compatible with the class scores computed by the ConvNet.

Convolutional Layer is a 3-dimensional arrangement consisting of 2-dimensional layers of neurons that are stacked together, one for each channel depth. It plays a crucial role in training and subsequently firing the neurons present in the network by performing the convolution operation over the input image.

Each neuron is connected to the receptive field. For an input image is of dimensions 30x30x3 and receptive field is 4 x 4, then each neuron in the Convolutional layer is connected to a region of 4x4x3 in the input volume. Hence each neuron will have 48 weighted inputs. For a specific R Value, we have a cross-section of neurons called 'depth column' extending the entire depth of the Convolutional layer totally committed to taking inputs from this region.

Shared Weights are used to optimize Convolutional layer by reducing the number of unique weights to train and to perform matrix calculations.

The ReLu (Rectified Linear Unit) Layer is the usual activation function for the outputs of the CNN neurons which can be mathematically described as a smoothed Softplus function

$$f(x) = 1 + e^{-x} \quad (5)$$

The Pooling layer lowers the spatial dimensions (width x height) of the Input image for the next Convolutional Layer and has no effect on the depth dimension of the volume.

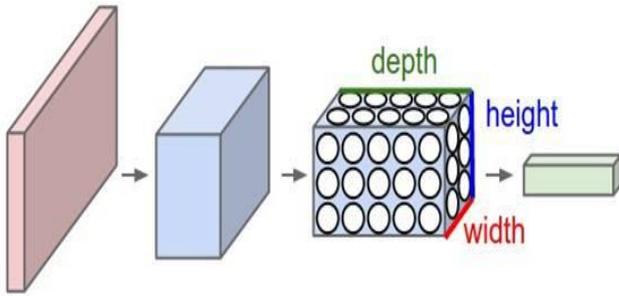


Fig. 11. A ConvNet architecture with its basic building layers

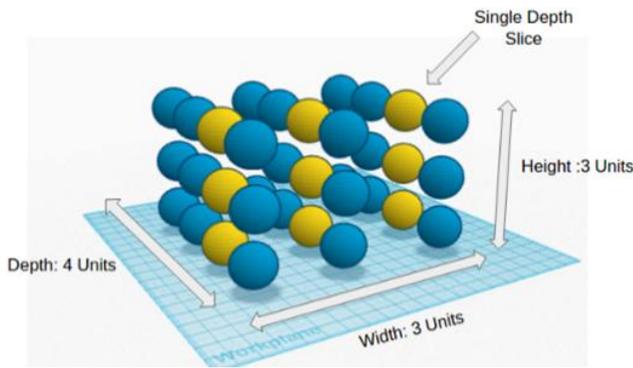


Fig. 12. A 3D representation of the Convolutional layer with size 3 x 3 x 4 = 36 neurons

The derivative of the Softplus function is a sigmoid function.

$$f'(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

Lowering of size results in information loss and hence the operation is called down sampling. It reduces the size and lessens computational complexity. The Fully Connected layer is completely connected to the output of the previous layer. It is essentially employed in the last stages of the CNN, connecting to the output layer and constructing desired number of outputs.

The spatial dimensions [width, height] of the Conv Layer as a function of the input volume size and the hyperparameters can be written as, for each ( $i^{\text{th}}$ ) dimension of the input volume,

$$W_{\text{out}}(i) = 1 + \frac{W_{\text{in}}(i) - R + 2P}{1 + e^{-x}} \quad (7)$$

where,  $i$  is the  $i^{\text{th}}$  input dimension,  $R$  = receptive field value,  $P$  = padding value.

CNNs give efficient control over performance using theoretical and mathematical insights and aids in solving real-world problems. CNNs and their applications can be extended to computer vision, self-driving cars, etc.

### B. Training and Interfacing

The hand gesture paradigm through a Convolutional Neural Network based classifier built using the MobileNet architecture. Using MobileNet we are able to run the hand gesture classifier on a single-board computer like a Raspberry Pi. Here a transfer learning-based technique is implemented on the classifier which could be trained to detect custom hand gestures from a PiCam module connected to the Raspberry Pi unit, near the user to detect palm movement. Using transfer learning enables us to build the classifier using just a few training image samples of the hand gestures. The Raspberry Pi loaded with this classifier can then be used to detect hand gestures in real-time using a camera module. Teachable Machine [20] is a web-oriented tool and plays a major role in making machine learning models faster, easier and available for free accessibility.

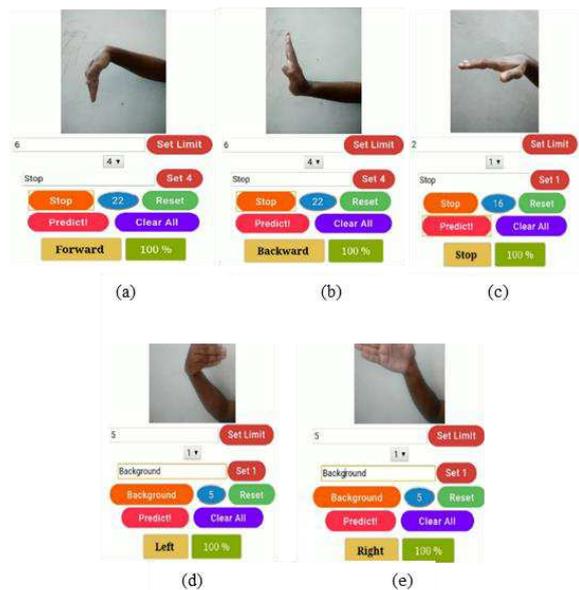


Fig. 13. Custom Hand Gestures for (a) forward, (b) backward, (c) stop, (d) left and (e) right

Accessing this, a custom hand gesture recognition classifier is created using this Google tool. It could be found that the classifier is able to successfully learn any new hand gesture with few training image samples.

Fig.13 shows the custom hand gestures created for various commands for the motor drive for driving the motor forward, backward, stop, left and right. Once training is done, the hand gestures which were processed by the CNN based Image Classifier on a Raspberry Pi 3B+ Board, can communicate with the receiver Arduino module. Once the gesture is detected, The Raspberry uses its onboard 2.4GHz to 5GHz wireless LAN capability to

send the corresponding motion instruction to the Arduino RF receiver. Once the appropriate hand gesture is detected, the Raspberry Pi, using RF receiver Arduino sends the command based on the gesture detected, to the motor controller so as to move the motors attached to the wheels of the wheelchair. It was found that the accuracy for the custom gestures using the Machine learning is more, with a faster time for response and that any number of custom gestures could be taken up for training and identification.

## VI. RESULTS AND DISCUSSIONS

The objective of the work was to design and implement a smart wheel chair operated on gestures. Various results in terms of simulating the hardware design, the developed hardware design and the tests results after working it with several samples are discussed here.

### A. Simulation Results

The effectiveness of the design implementation could be accessed through simulating the circuit through Proteus, a software program with a simulation tool containing a detailed library of most of the electronic component's parts. As shown in Fig.14, it has the capability to simulate the microcontroller module for testing the circuits throughout the design, along with various other electronic components showing the simulation circuit diagram of the system.

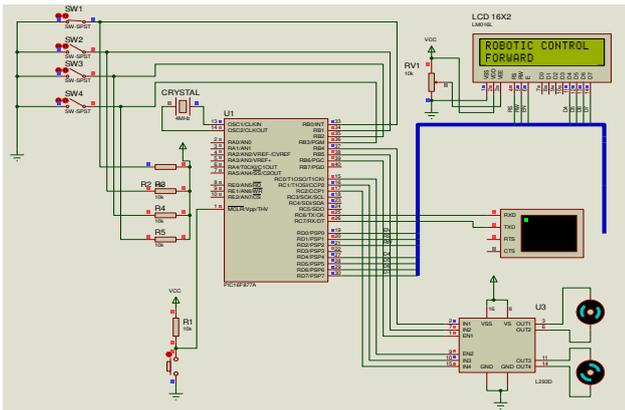


Fig. 14. Simulation Circuit showing output for various input command

The system simulated consists of 4 switches (SW1 to SW4), each of the switch representing each of the gestures. The circuit also consists of an LCD display, which displays the status of the gesture, a microcontroller module that represents the Arduino module of the hardware that controls the motor drive. It also consists of a Crystal Oscillator, Power Supply, Transmitter-Receiver circuit, Motor driver and 2 dc Motors. When none of these switches (SW1 to SW4) are connected, it indicates the Stop gesture. When switch SW1, is connected, the system takes the command as Forward, as indicated by the LCD display, results in driving the motors in the forward direction. When switch SW2, is connected, the system takes the command as Backward, which will be indicated in the LCD display, while the driver drives the motors in the reverse direction. The powering of switch SW3, makes the system recognize the command as Right, which stops the second dc motor, while the first dc motor steers the entire system into the clockwise rotation of the wheel chair. The switch SW4, when turned ON, the system recognizes the command as Left, which stops the first dc

motor, while the second dc motor drives the entire system into the anti-clockwise rotation of the wheel chair. Thus, the simulation results are satisfactory for the proposed design to be implemented as a hardware prototype.

### B. Hardware Design Results

The hardware implementation as discussed under Section IV & V resulted in developing a hardware assembly. Fig. 15 illustrates the hardware model based on the Accelerometer control module in the Transmitter end that includes the Accelerometer mounted on a glove connected with the Arduino Uno which connects with the encoder-RF transmitter assembly and also controls the display in the LCD display and the Keypad Interface module.

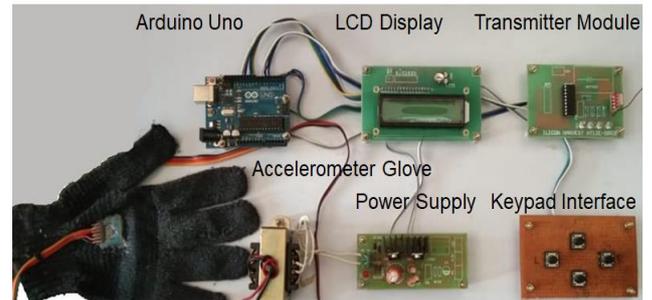


Fig. 15. Accelerometer control Module

All these components are powered by the Power supply module.

The receiver end module of the hardware as shown in Fig.16 is attached to the wheel chair assembly. This module consists of a Mode selector switch to decide upon the mode of navigation, the decoder- RF Receiver assembly, Arduino Uno, LCD display, Wi-Fi module, IR Transmitter- Receiver assembly and Motor driver connected with two dc motors.

The overall wheel chair assembly is presented in Fig. 17, which shows the receiver end module securely fastened in the lower portion of the chair. The wheelchair assembly is made with a grill type pattern to reduce the weight of the system as well as to indicate the receiver end module. The interactive camera is mounted on the top of the wheelchair assembly to help the user to indicate the user about the traffic while travelling outdoors and indicate the obstacles while navigated indoors. A smartphone could be placed near the arm-rest, which could help the user to navigate as well as to interact with the camera interface, if required.

The Raspberry Pi module along with PiCam interface is shown in Fig.18. The Raspberry Pi used here is 3B plus model where a camera module PiCam which is of 8-megapixel resolution could be directly interfaced to the Raspberry Pi module that could take up the hand gestures as inputs for communicating the command for the drive control.

### C. Experimental Results

The gesture patterns are tested for its response and accuracy for an Accelerometer Control.

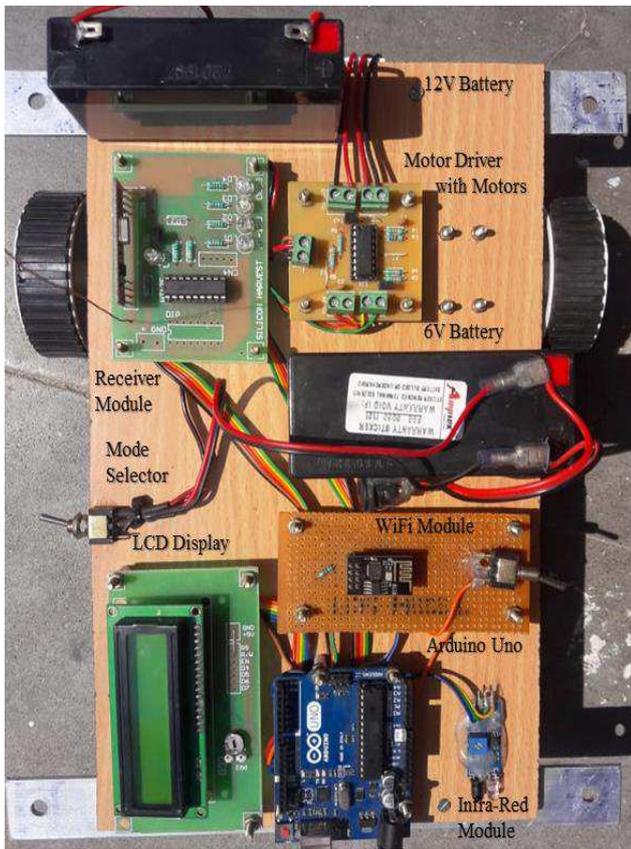


Fig. 16. Receiver End Module



Fig. 17. Overall Wheel Chair Assembly

The number of input gestures that are recognized correctly for the different gestures are tabulated in Table II. Each gesture was tested for 50 samples for its correctness and it was found that gestures forward and left was recognized most and gesture backward had lesser response comparatively. A similar testing for response and accuracy was conducted for the CNN based Machine Learning Control and the number of input gestures that are recognized correctly for the different gestures are tabulated in Table III. 50 samples were tested for each gesture for its

correctness and it was found that the gestures backward and stop had relatively lesser response.

A Comparison of accuracy is made with respect to recognition capability for various gestures by Accelerometer Control and Machine Learning Control and tabulated as shown in Table IV.

Table V enumerates the overall accuracy in gesture recognition by both Accelerometer Control and Machine Learning Control for an overall testing sample of 250 each for individual controls inclusive for all gestures.

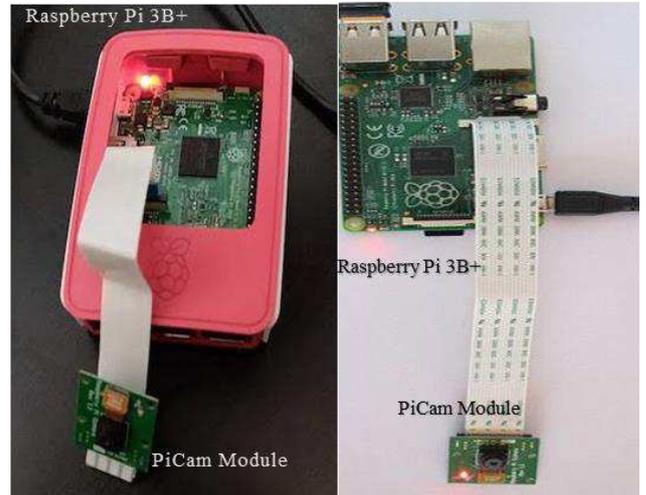


Fig. 18. Raspberry Pi Module with PiCam

TABLE II. RECOGNITION OF GESTURES FOR ACCELEROMETER CONTROL

Recognized Gestures	Experimental Results	Input Gestures				
		Backward	Forward	Stop	Left	Right
Backward		43	3	2	2	3
Forward		4	45	4	1	1
Stop		3	2	44	2	2
Left		0	0	0	45	0
Right		0	0	0	0	44

TABLE III. RECOGNITION OF GESTURES FOR MACHINE LEARNING CONTROL

Recognized Gestures	Experimental Results	Input Gestures				
		Backward	Forward	Stop	Left	Right
Backward		47	0	1	0	0
Forward		1	48	2	1	1
Stop		2	2	47	1	1
Left		0	0	0	48	0
Right		0	0	0	0	48

TABLE IV. COMPARISON OF ACCURACY FOR VARIOUS GESTURES

Methodology	Backward	Forward	Stop	Left	Right
Accelerometer Control	86.0	90.0	88.0	90.0	88.0
Machine Learning Control	94.0	96.0	94.0	96.0	96.0

TABLE V. COMPARISON OF OVERALL ACCURACY IN GESTURE RECOGNITION

Methodology	Average Accuracy (%)	No. of Testing Samples
Accelerometer Control	88.4	250
Machine Learning Control	95.2	250

It could be found that though the Accelerometer Control is reasonably accurate in hand gesture recognition, the accuracy in pattern recognition of the CNN based Machine Learning Control yielded far better results as observed from Tables IV & V. As discussed in the previous sections there are lots of advantages of Machine Learning Control with respect to its capability in pattern recognition and computation speed resulting in faster response and accuracy. Also, with the advent of onboard units like Raspberry Pi with advances in its processing speed, memory and transfer capabilities, any number of custom gestures could be taken as inputs without any complexity in pattern identification or in transferring data which could be extended for remote access for application development in an IoT environment.

## VII. CONCLUSION AND FUTURE SCOPE

This paper describes about the smart wheelchair control based on hand gesture inputs of the user results in the motion control command for the wheelchair for navigation in the desired directions. The input hand gestures are recognized either by using a system based on MEMS accelerometers control or by a CNN based machine learning control. The simulations for checking the hardware implementation credibility were carried out. Both the controls were implemented and are developed into a hardware prototype. The testing of the hardware with respect to its accuracy in recognition of the hand gesture inputs was made and it was found that the machine learning control had reasonably better accuracy in pattern identification. Thus, the implementation of the smart wheelchair takes various aspects of the disabled and the elderly into consideration with regards to the wellbeing of the user in terms of providing various choices of control, an interactive camera for navigation and two way communication, access to change angle of camera rotation, LCD display indicating the mode and the gesture, obstruction indicator, keypad interface for technical lame user, traffic access, etc. Providing an access of the camera over the IP address for the caretaker of the user leads to an added security and ensuring the wellness of the user. The implementation and testing proved to be satisfactory in terms of results and it is imperative that this system will benefit the society, where 30.6 million people globally have difficulty in walking.

Aspects regarding accidental fall, like providing protection during a fall, passing information to caretakers and health workers through a GPS module could be added as additional features. Efforts also could be made for including a health monitoring system module for the user could also pave way serving the user better. These could be features that could be investigated with our future work.

## Declarations:

Funding: Not applicable

Conflicts of interest/Competing interests: No

Availability of data and material: Nil

Code availability: Software Application

## VIII. REFERENCES

- [1] R. C. Simpson, "Smart wheelchairs: A literature review", *Journal of Rehabilitation Research & Development (JRRD)*, Vol 42, Number 4, pp. 423–436, July/August 2005.
- [2] L. P. Reis, B. M. Faria, S. Vasconcelos, and N. Lau, "Invited paper: Multimodal interface for an intelligent wheelchair volume 325 of Lecture Notes in Electrical Engineering", pages 1–34. Springer International Publishing, 2015
- [3] D. Sinyukov, R. Desmond, M. Dickerman, J. Fleming, J. Schaufeld, and T. Padir, "Multi-modal control framework for a semi-autonomous wheelchair using modular sensor designs", *Intelligent Service Robotics*, Vol 7, Number 3, pp.145–155, Jul. 2014.
- [4] D.K. Rathore, P. Srivastava, S. Pandey, and S. Jaiswal, "A novel multipurpose smart wheelchair" *Proceedings of IEEE Students' Conference on Electrical, Electronics and Computer Science*, pages 1–4, Bhopal, Mar. 2014.
- [5] R. Simpson, "Smart Wheelchair Component System", *Journal of Rehabilitation, Research and Development*, Vol 41, Number 3B, pp.429–442, 2004.
- [6] O. Mazumder, A.S. Kundu, R. Chattaraj, and S. Bhaumik, "Holonomic wheelchair control using EMG signal and joystick interface", *Proceedings of Recent Advances in Engineering and Computational Sciences (RAECS)*, Chandigarh, India, pp.1–6, Mar. 2014.
- [7] S.Guo, R. A. Cooper and G. G. Grindle, "Development of Head-Operated, Isometric Controls for Powered Mobility", *Proceedings of RESNA 27th International Annual Conference*, Orlando, Florida, 2004.
- [8] M.E. Lund et al., "Inductive tongue control of powered wheelchairs", *Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Buenos Aires, Argentina, pp. 3361 - 3364, 2010.
- [9] I. Mougharbel, R. El-Hajj, H. Ghamlouch and E. Monacelli, "Comparative study on different adaptation approaches concerning a sip and puff controller for a powered wheelchair", *Proceedings of IEEE Science and Information Conference (SAI)*, London, pp. 597 - 603, 2013.
- [10] Y. Matsumoto, T. Ino and T. Ogsawara, "Development of intelligent wheelchair system with face and gaze based interface", *Proceedings of 10th IEEE International Workshop on Robot and Human Interactive Communication*, Bordeaux, Paris, France, pp. 262 – 267, 2001.
- [11] J. del R Millán, "BMI: Lessons from tests with impaired users," *2014 International Winter Workshop on Brain-Computer Interface (BCI)*, Jeongsun-kun, 2014, pp. 1-1. , Feb. 2014

- [12] S. D. Suryawanshi, J. S. Chitode and S. S. Pethakar, "Voice Operated Intelligent Wheelchair", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Volume 3, Issue 5, pp. 487 – 490, May, 2013
- [13] S. Paulose, M.P.F. Anooda, G. Mohan, M. S. Sajana and K. A. Anupama, "Automatic Wheelchair using Gesture Recognition Along with Room Automation", Transactions on Engineering and Sciences, Vol. 2, Issue 5, pp. 40 - 43, May 2014.
- [14] J. Philips et al., "Adaptive Shared Control of a Brain-Actuated Simulated Wheelchair", Proceedings of IEEE 10th International Conference on Rehabilitation Robotics, ICORR, Noordwijk, Netherlands, pp. 408 – 414, 2007
- [15] M. Mazo et al., "Wheelchair for physically disabled people with voice, ultrasonic and infrared sensor control", Autonomous Robots, Volume 2, Issue 3, pp. 203-224, 1995.
- [16] R. Xu, S. Zhou and W. J. Li, "MEMS Accelerometer Based Nonspecific-User Hand Gesture Recognition," in IEEE Sensors Journal, vol. 12, no. 5, pp. 1166-1173, May 2012.
- [17] M. Alam and M. A. Yousuf, "Designing and Implementation of a Wireless Gesture Controlled Robot for Disabled and Elderly People," 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, 2019, pp. 1-6.
- [18] Andrej Karpathy, "CS231n: Convolutional Neural Networks for Visual Recognition" Available from: <<https://cs231n.github.io/convolutional-networks/>>
- [19] Abhinav Saxena, "Convolutional Neural Networks (CNNs): An Illustrated Explanation" Available from: <<https://blog.xrds.acm.org/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/>>
- [20] Teachable Machine Available from: <<https://teachablemachine.withgoogle.com/>>

# Figures



Figure 1

Hand Gestures for (a) backward, (b) forward, (c) stop, (d) left and (e) right

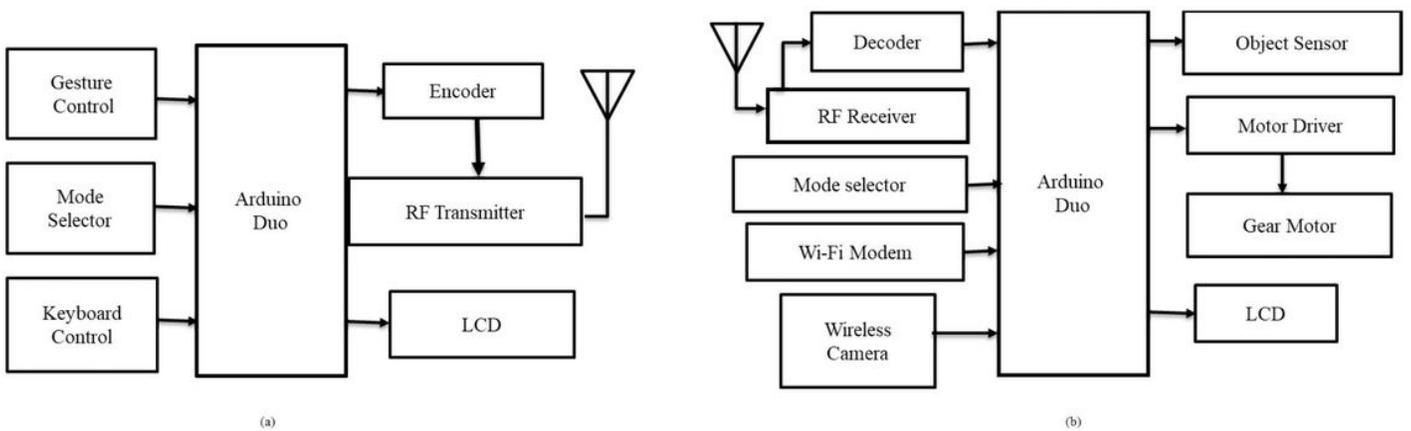
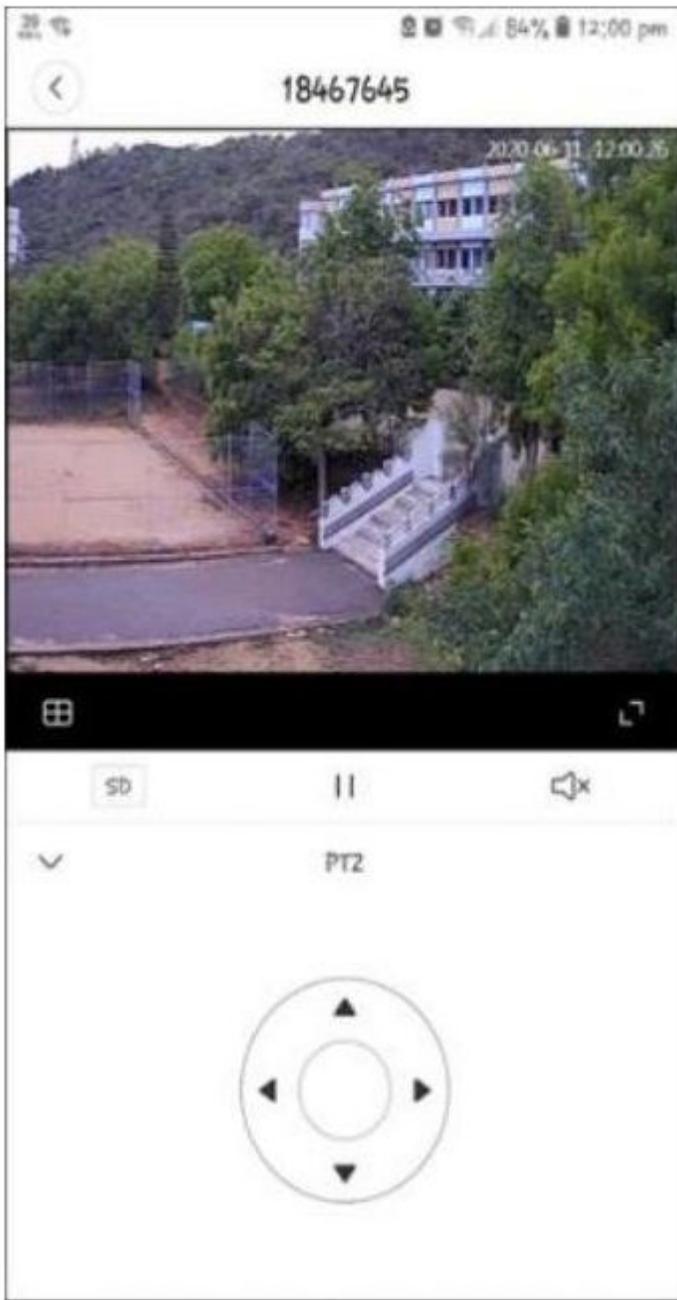
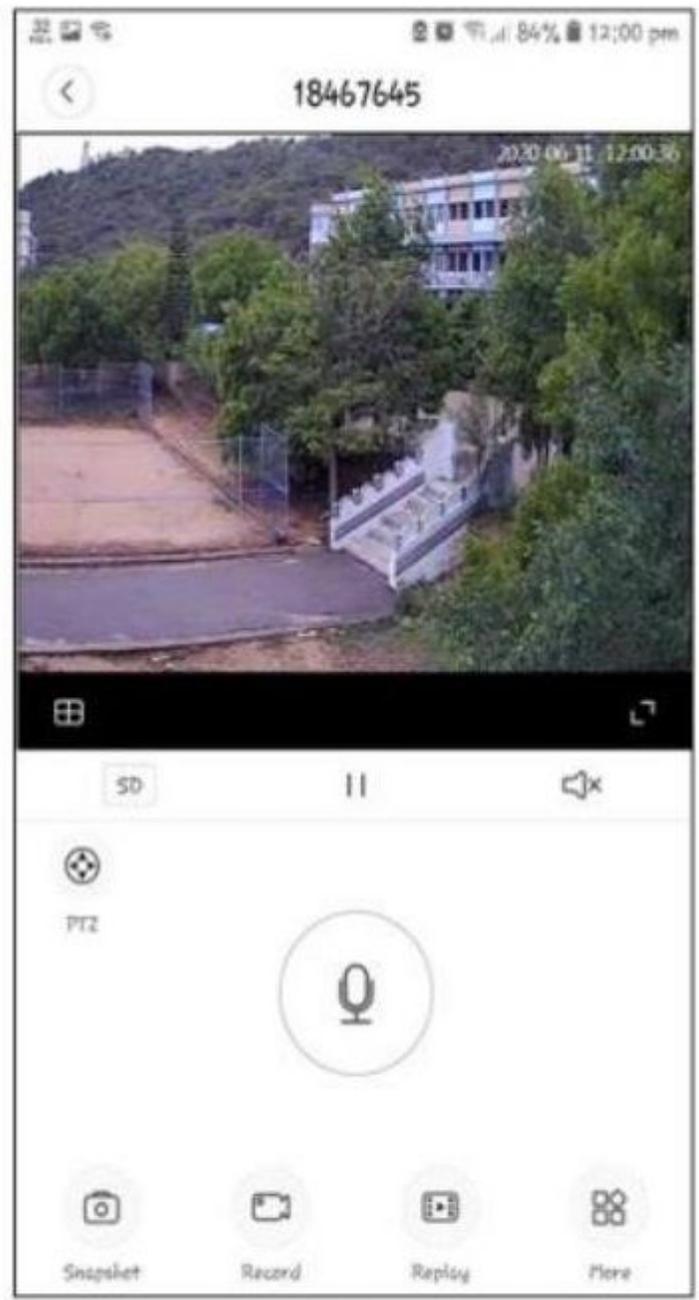


Figure 2

(a) Transmitter end Module (b) Receiver end Module



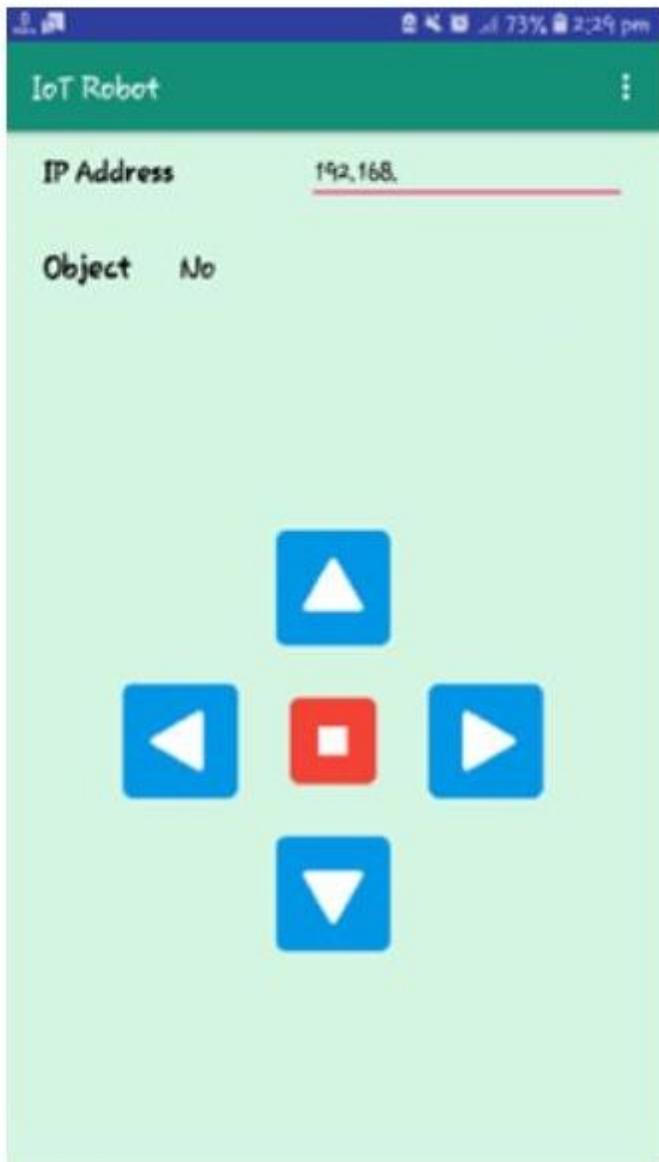
(a)



(b)

**Figure 3**

Camera module accessibility using the mobile App (a) Camera Navigation control interface (b) Audio communication interface



**Figure 4**

Navigation of Wheelchair through Mobile interface accessed through Wi-Fi

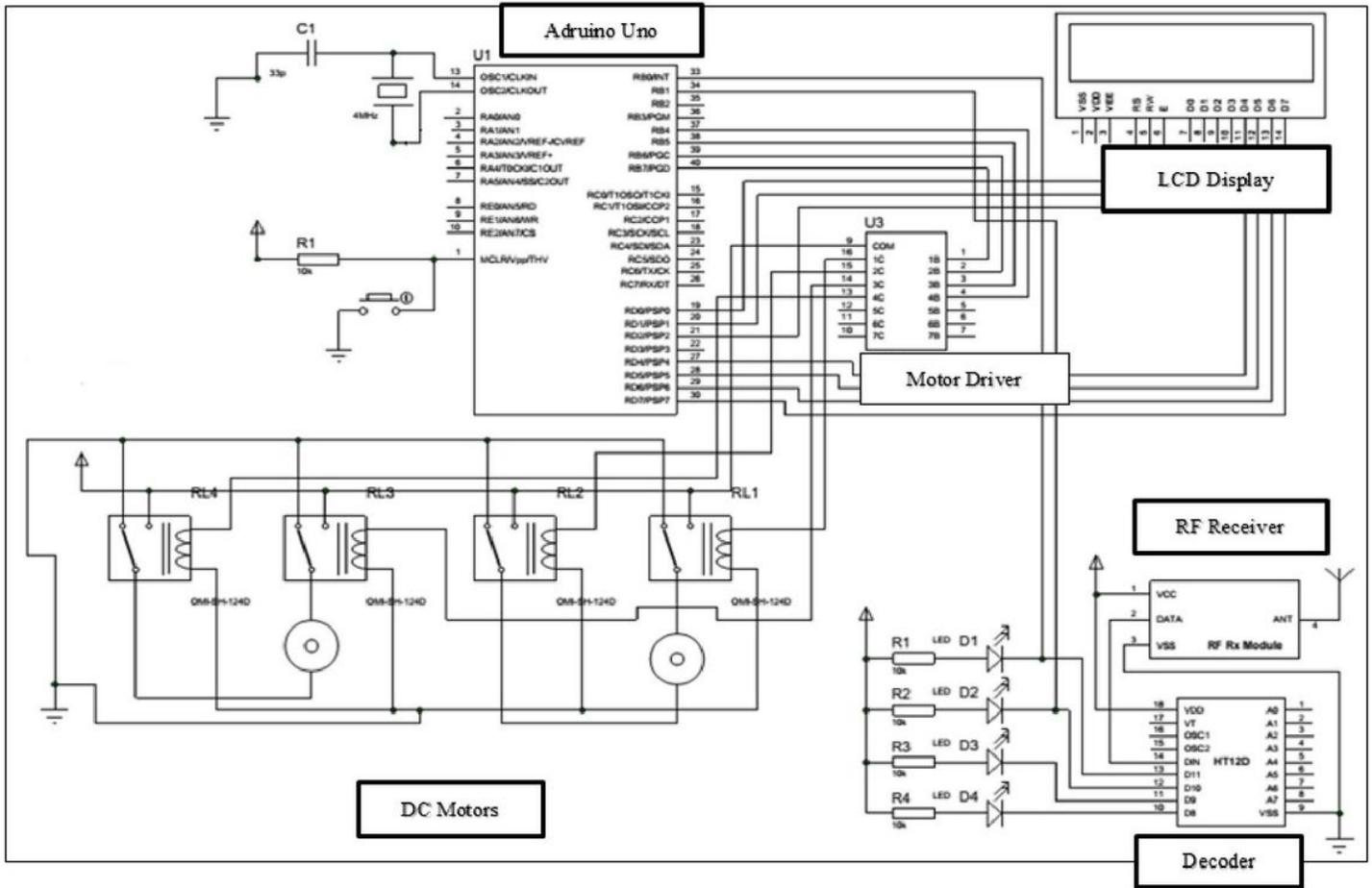


Figure 5

Circuit diagram of the Receiver Module

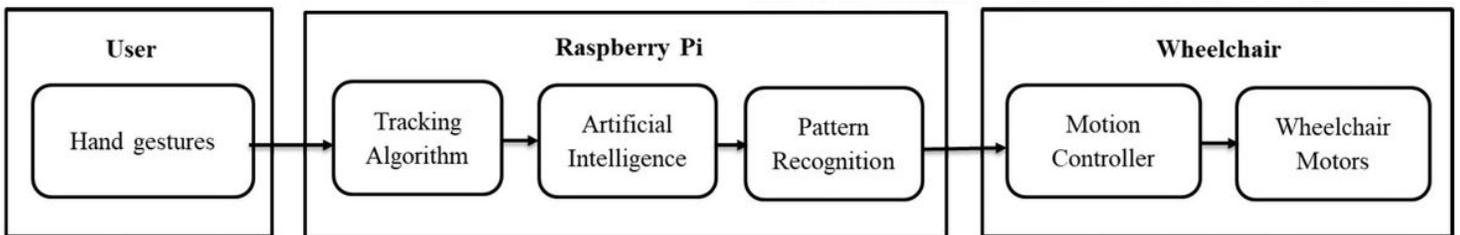
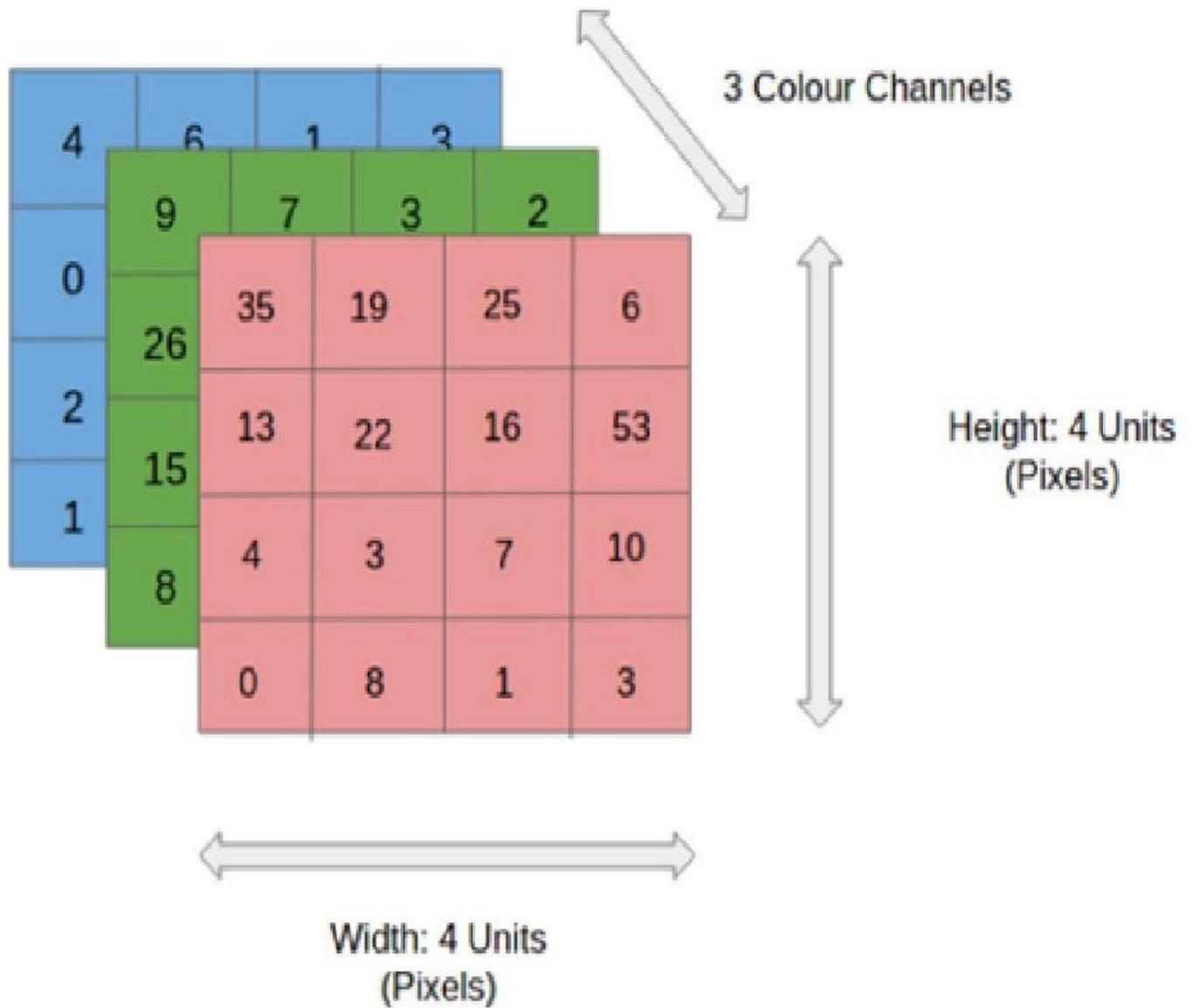


Figure 6

Machine Learning Interface



**Figure 7**

The cross-section of an input volume (RGB image) of size comprised of 3 Colour channel matrices of the input image

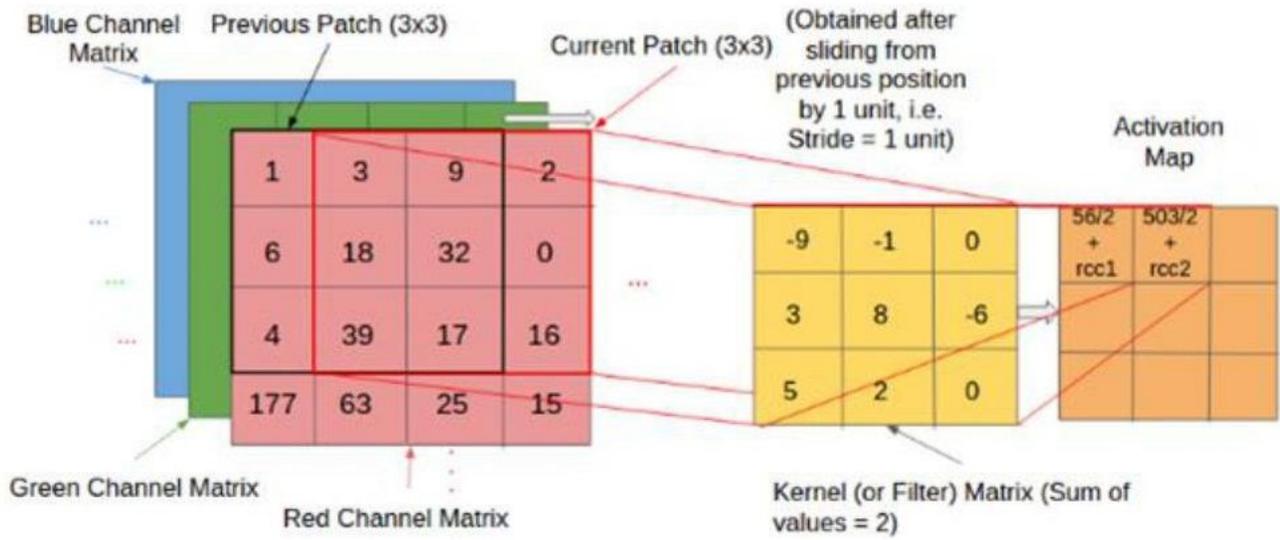


Figure 8

The convolution value calculation by the dot product of the values in the Kernel and their channel matrices

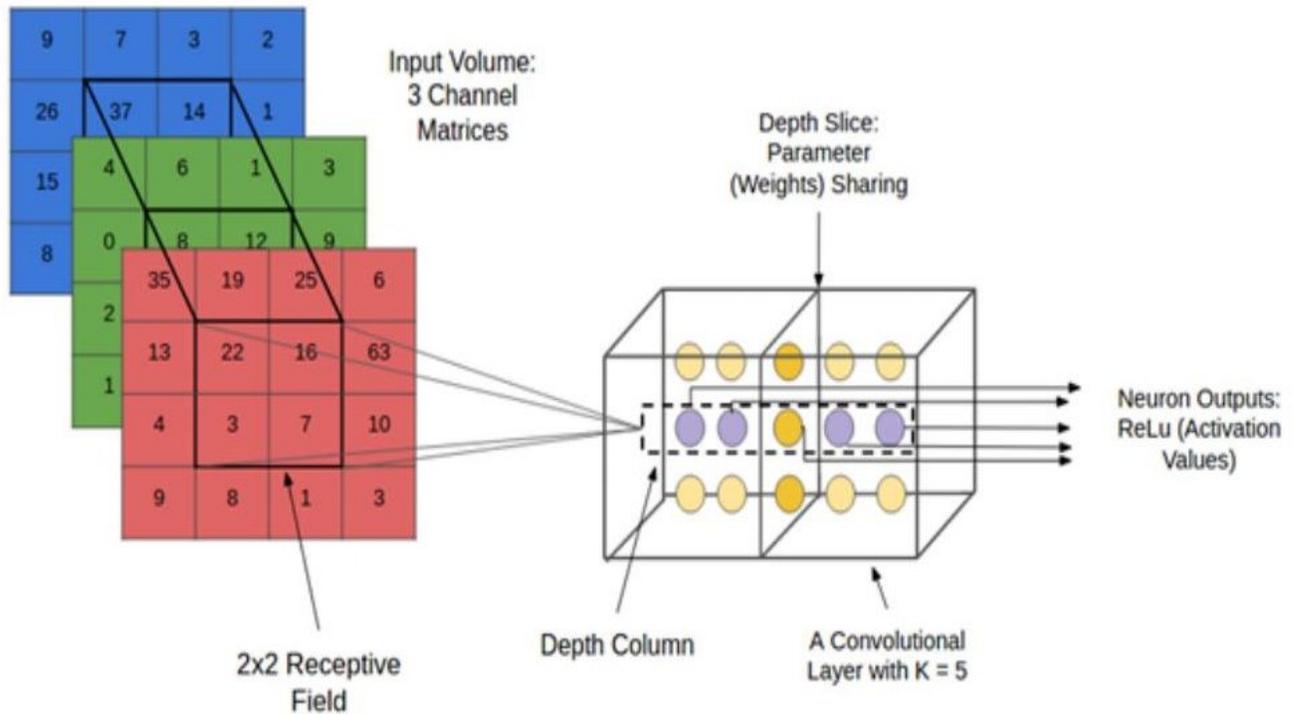


Figure 9

The Concept of Receptive Field

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

**Figure 10**

A zero-padded 4 x 4 matrix becoming a 6 x 6 matrix

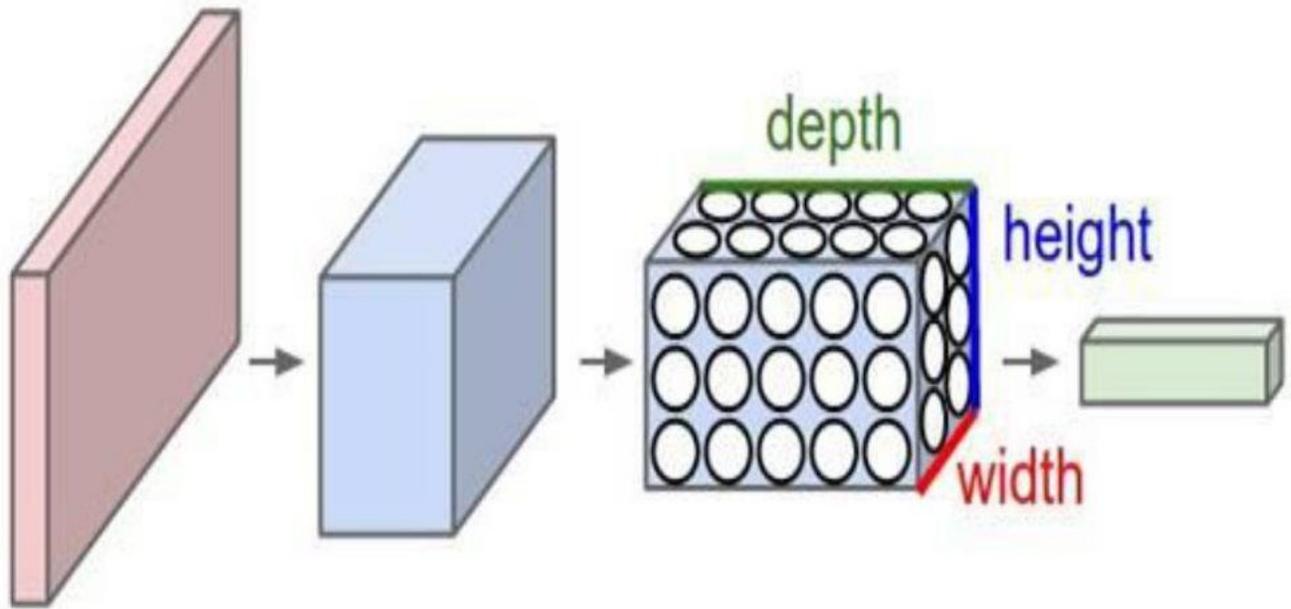


Figure 11

A ConvNet architecture with its basic building layers

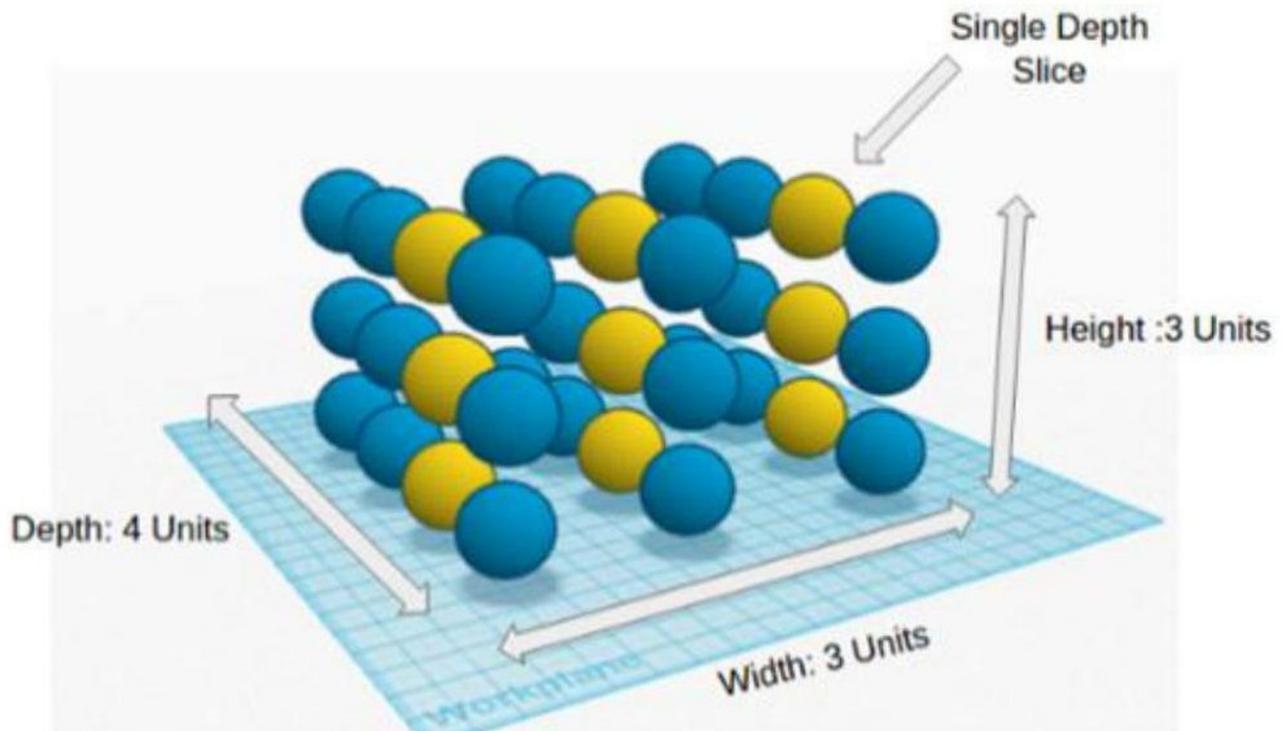
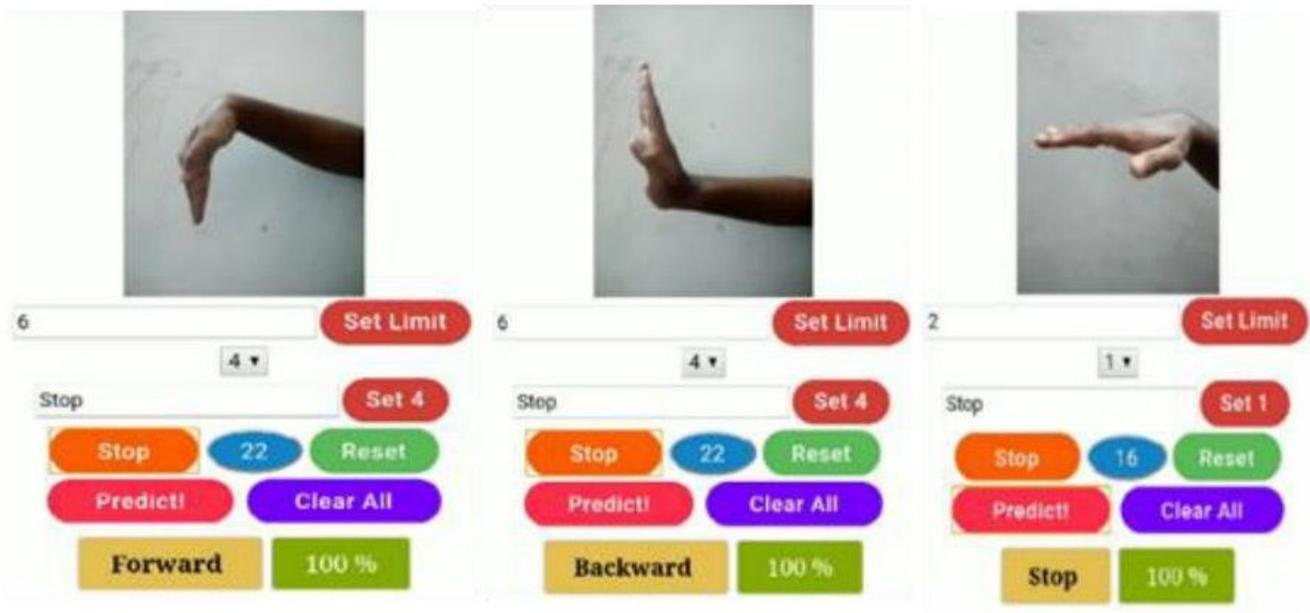


Figure 12

A 3D representation of the Convolutional layer with size  $3 \times 3 \times 4 = 36$  neurons



(a)

(b)

(c)



(d)

(e)

Figure 13

Custom Hand Gestures for (a) forward, (b) backward, (c) stop, (d) left and (e) right

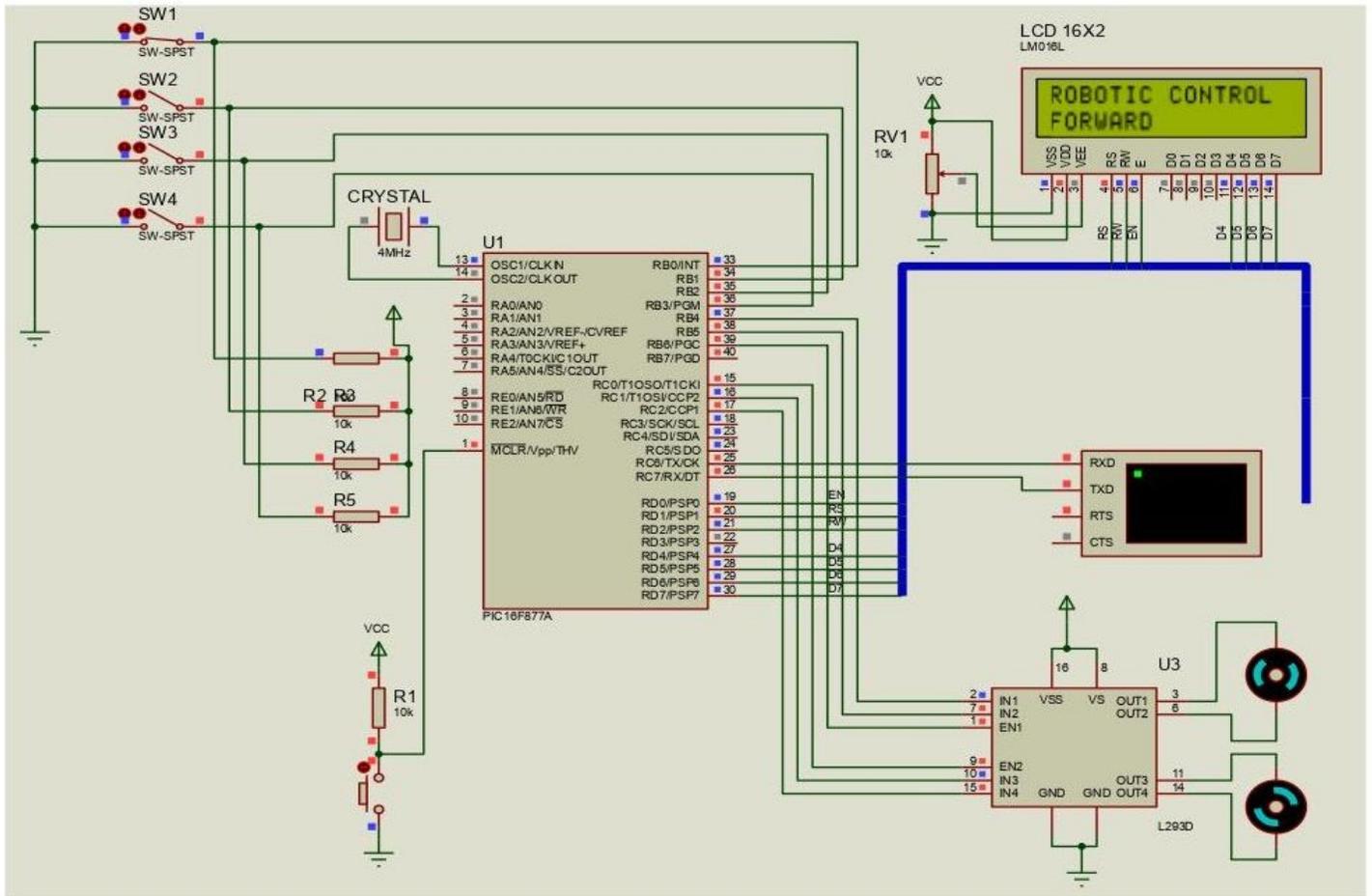


Figure 14

Simulation Circuit showing output for various input command

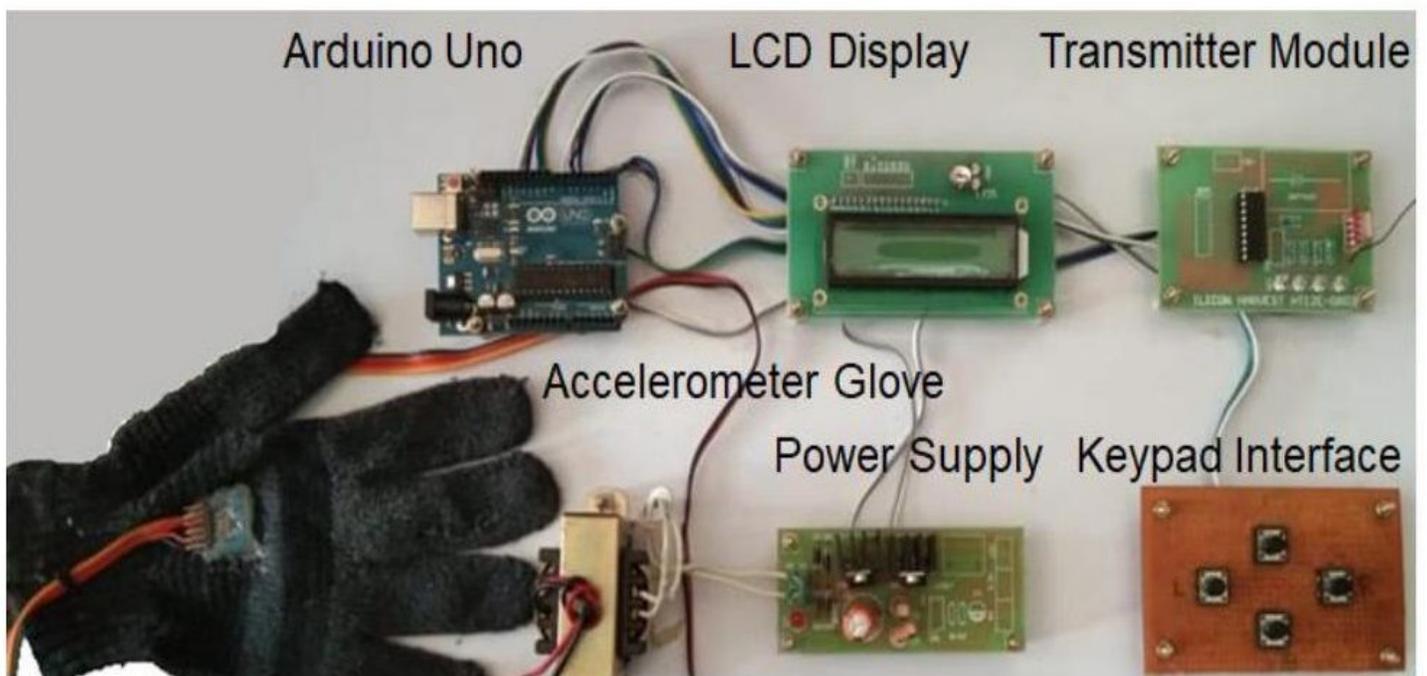


Figure 15

Accelerometer control Module

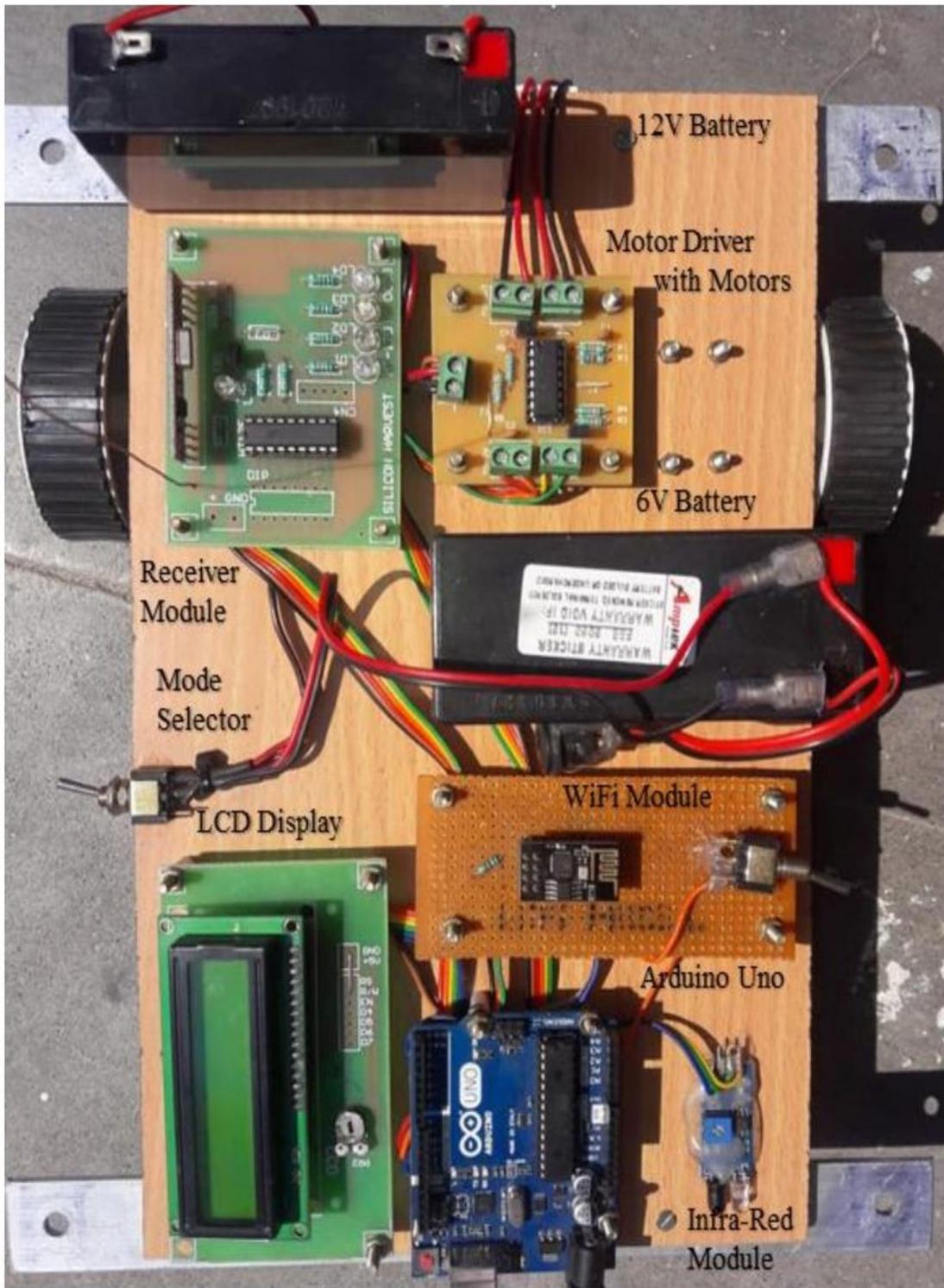


Figure 16

Receiver End Module



**Figure 17**

Overall Wheel Chair Assembly

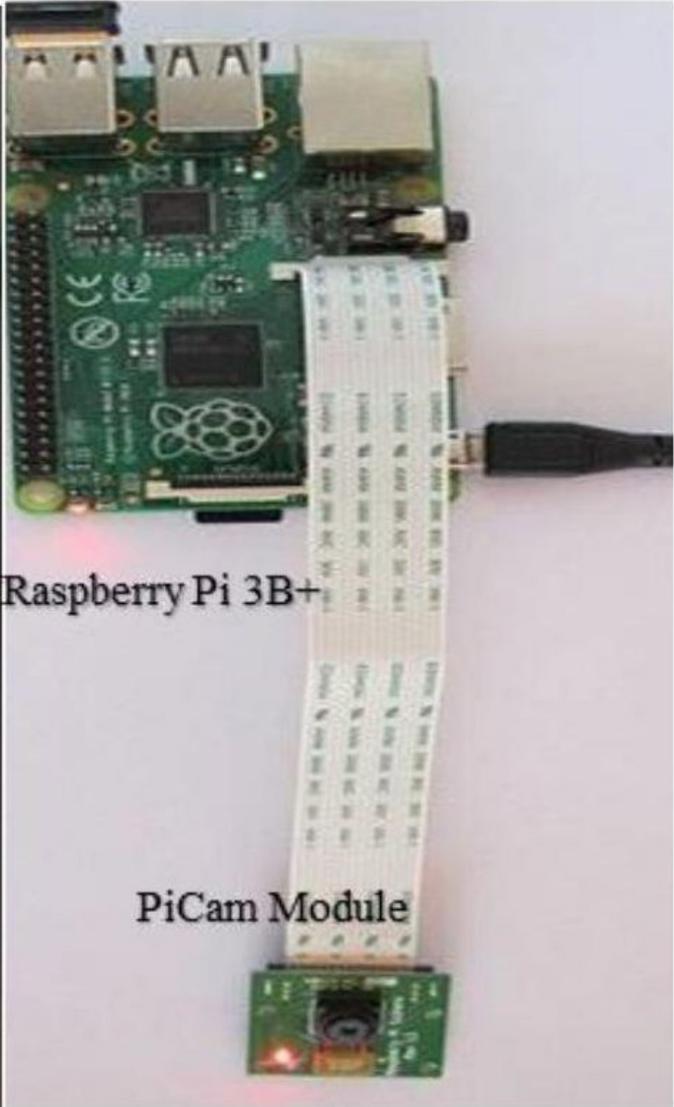


Figure 18

Raspberry Pi Module with PiCam