HOME APPLIANCE CONTROL FOR VISUALLY AND VERBALLY IMPAIRED USING DEEP LEARNING AND IoT FOR GESTURE RECOGNITION

A PROJECT REPORT

Submitted by

MEENAKSHI. S. S (19121012)

RAKESH. K (19121016)

Under the guidance of

DR. P. SANKAR

in the partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE,

CHENNAI- 603 103

MAY 2023

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)
CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report **HOME APPLIANCE CONTROL FOR VISUALLY AND VERBALLY IMAPIRED USING DEEP LEARNING AND IoT FOR GESTURE RECOGNITION** is the bonafide work of “**MEENAKSHI. S.S (19121012) and RAKESH. K (19121016)”** who carried out the project work under my supervision during the academic year 2021-2022

SIGNATURE	SIGNATURE
Dr.AL.Vallikannu	Dr. P. Sankar
Professor & Head of the Department	Professor
Department of Electronics and	Department of Electronics and
Communication Engineering	Communication Engineering

INTERNAL EXAMINER	**EXTERNAL EXAMINER**
Name: ______________	Name: _______________
Designation: ________	Designation: _________
	Institute Name: _______

Project Viva-voice conducted on: ________

ACKNOWLEDGEMENT

It is our extreme pleasure to thank our Chancellor **Dr. (Mrs.) Elizabeth Verghese** and Vice-Chancellor **Dr.S.N. Sridhara**, Hindustan Institute of Technology & Science for providing a conducive environment which helped us to pursue our project in a diligent and efficient manner.

We wish to express our sincere gratitude to Dean (Engineering & Technology) **Dr Angelina Geetha**, Hindustan Institute of Technology & Science for her valuable directions, suggestions, and support.

We are thankful to **Dr AL.Vallikannu**, Head of the Department, of Electronics and Communication Engineering for having evinced keen interest in our project and for her continued support.

We are indebted to our Project Guide **Dr. P. Sankar,** professor and Project Coordinator **Ms. Vimalarani. G**, Assistant Professor, Electronics and Communication Engineering for their valuable guidance and technical support in the accomplishment of our project.

Our sincere thanks to all the teaching and non-teaching staff who have been constantly supporting us throughout the accomplishment of this project

ABSTRACT

Emerging technologies make human life easier by giving us at most comfortat the tip of our hands. Yet, accessibility of these smart devices by visually or verbally impaired patients is still a tedious process. For instance, Smart home automation devices such as Alexa or Siri require vocal activation which is impossible for a mute person. One-to-one communication between these devices and man requires additional support for anyone with a lack of hearing abilities. An alternative to speech recognition for appliance management is gesture recognition. Gesture recognition techniques have been used for small-scaleapplications of home automation but a model exclusive for impaired patients hasalways only been theoretical. Gesture recognition technology based on infrared light uses the principle of infrared radiation to recognize gestures. The disadvantages are that the gestures cannot be recognized in non-line-of-sight conditions. But embedding this with an IoT module helps us in saving the data onthe cloud and aids with easy simultaneous accessibility. This proposed module uses hand gesture recognition techniques for appliance control and uploads the obtained data to the cloud server using an internet camera module. Any impairedperson will be able to control any device connected to the local network without any external aid. This module can be tested in a homestay for the impaired to make the stay more technologically viable.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER – 1
INTRODUCTION

The 21st century has witnessed the digitalization of all things around us. All appliances can be electronically managed in the comfort of one place using forefronting applications such asmachine learning and the Internet of Things (IoT). the rapid developments of big data, cloud computing and the Internet of Things have promoted the development of various intelligent technologies, such as smart homes, smart schools, smart cars, and robots

Controlling smart home appliances is a fore-fronting application of deep learning. Smart home devices such as Alexa or Google Home make use of speech for controlling smart appliances. For verbally impaired people this is a hindrance. An alternative to speech recognition for appliance management is gesture recognition. We propose a device that a verbally or visually impaired person can use to control an appliance using hand gestures. The device makes use of an internet camera module. The internet camera is connected to the same network as the other appliances that an impaired person wishes to control. The main focusof the project is to provide easy accessibility of smart devices to visually and verbally impairedpeople. The secondary focus of this project is to make the interface wireless. Any communication between the camera module to the appliances is carried out via wireless communication for which IOT is utilized. The project aims in designing a device that incorporates control of smart appliances such as TV, Dishwashers, washing machines etc usinggestures for verbally and visually impaired patients. The prerequisite behind any smart device is reading and understanding of the gesture and corresponding it to its allotted functions. The device aims to achieve this by using an internet camera which is connected to a common network as the appliance one wants to control. The camera reads the associated gestures of theperson using a deep learning algorithm and enables control of the appliances according to the given gestures using the Internet of Things (IoT). The project revolves around a few major phases: the set-up phase, testing phase, working phase, and advancement phase. The initial phase of the projectinvolves the training of a neural network model that can read, recognize, and associate the gestures with specific functions such as ON/OFF, Volume UP/DOWN etc.

The next phase involves setting up an internet camera module that is wearable by an impaired person. The camera is used for capturing the gesture made and for controlling the appliances which is connected to a network the same as the appliance. As a proof of concept, the project is implementedfor just one appliance. However, a few issues that we hope to resolve include Power consumption. Since the project is helpful in assisting impaired people, it requires having tostay ON state for long hours. This may cause additional power usage. Around the states of India, Tamil Nadu has an association for the Visually Impaired where they provide an id, and homestay for the blind. This device can be utilized for making the association more technologically viable.Since the device uses gestures for appliance control any impaired person around the homestaycan get the help, they need at any instant possible without having to be dependent on another individual for aid. The same technology can further be used to make wheelchairs, which, upon making a certain gesture, will bring themselves towards an impaired person.

1.1 PROBLEM STATEMENT

Existing smart appliances are formulated locally adhering in mind the two assumed facts:

- Any kind of controlling of the appliance can be attained by vocal activation whichis a hindrance for a verbally impaired person
- The user should stay in the vicinity of the device they wish to control which is a major con for a visually impaired person

1.2 OBJECTIVE

To formulate a module to control smart home electrical appliances using gesture recognition techniques for visually and verbally impaired patients

1.3 ORGANIZATION OF THE REPORT

The report is organized as follows: The previously emerged works is been mentioned briefly in Chapter2 as Literature Survey. Chapter 3 focuses on the flow of control of data in the proposedsystem and the explanatory part of the proposed diagram. The hardware and the software components required are briefly explained here as well. The following chapter deals with the methodology used. And the algorithm behind the neural network model is explained henceforth.Furthermore, result and analysis of the module is shown in Chapter 5 with real-time images of the output. The final chapter deals with the conclusion, future advancements, merits as well demerits of the projects. The report ends with an appendix part which the program code of theserver side and the client side is entitled.

CHAPTER - 2

LITERATURE SURVEY

The existing systems help us in using vocal recognition techniques for appliance control which is a tedious process to be used by a mute person. Hence wearable devices within proximity of the appliance were developed. This was a hindrance for blind people as they could not be within the proximity of the appliance they wish to operate. [10]

2.1 REAL-TIME HAND GESTURE RECOGNITION USING SURFACE ELECTROMYOGRAPHY AND MACHINE LEARNING

In a paper proposed by **Andrés G. Jaramillo; Marco E. Benalcázar** surface electromyography (EMG) and Machine Learning techniques are used as well.. The recognition of gestures using EMG is not a trivial task because there areany several physiological processesin the skeletal muscles underlying their generation. They have limitations both in the number of gestures to be recognized (i.e., classes) and in the processing time. Generally, the main difficulties of the hand gesture recognition models with EMG using Machine Learning are: thenoisy behaviour of EMG signal, and the small number of gestures per person relative to the number of generated data by each gesture (i.e., curse of dimensionality).

2.2 A WEARABLE BIOSENSING SYSTEM WITH IN-SENSOR ADAPTIVEMACHINE LEARNING FOR HAND GESTURE RECOGNITION

Further in the paper proposed by **Ali Moin, Andy Zhou**,Wearable devices that monitor muscleactivity based on surface electromyography could be of use in the development of hand gesture recognition applications. Such devices typically use machine-learning models, either locally orexternally, for gesture classification. However, most devices with local processing cannot offertraining and updating of the machine-learning model during use, resulting in suboptimal performance under practical conditions.

2.3 SURFACE EMG HAND GESTURE RECOGNITION SYSTEM BASED ON PCAAND GRNN

In the thesis proposed by **Jinxian Qi, Guozhang Jiang, Gongfa Li** The principal component analysis method and GRNN neural network are used to construct the gesture recognition system, so as to reduce the redundant information of EMG signals, reduce the signal dimension, improve the recognition efficiency and accuracy, and enhance the feasibility of real-time recognition. After dimension reduction and neural network learning, the overall recognition rate of the system reached 95.1%, and the average recognition time was 0.19 s. Finally deep learningmethods came into implementation. dynamic hand gesture recognition systems use different techniques to extract handcrafted features followed by a sequence modelling technique such as a hidden Markov model (HMM). However, the recent success of deep learning techniques in image classification, object recognition, speech recognition, and human activity recognition has encouraged many researchers to exploit them for hand gesture recognition.

2.4 HAND-GESTURE RECOGNITION USING TWO-ANTENNA DOPPLER RADARWITH DEEP CONVOLUTIONAL NEURAL NETWORKS

Antenna systems were embedded along with deep learning algorithms to provide point to pointcommunication between the user and the devices in a paper proposed by **Sruthy Skaria; Akram Al-Hourani; Margaret Lech; Robin J. Evans** where they used a miniature radar sensor to capture Doppler signatures of 14 different hand gestures and train a deep convolutional neural network (DCNN) to classify these captured gestures. he classification results of the proposed architecture show a gesture classification accuracy exceeding 95% and a very low confusion between different gestures. This is almost 10% improvement over the single-channel Doppler methods reported in the literature.

2.5 PERFORMANCE EVALUATION OF CONVOLUTIONAL NEURAL NETWORKFOR HAND GESTURE RECOGNITION USING EMG

Electromyography (EMG) is a measure of electrical activity generated by the contraction of muscles which was a model proposed by **Ali Raza Asif, Asim Waris, Syed Omer Gilani, Mohsin Jamil, Hassan Ashraf, Muhammad Shafique, and Imran Khan Niazi**. It was observed that regardless of network configuration some motions (close hand, flex hand, extend the hand and fine grip) performed better (83.7%

± 13.5%, 71.2% ± 20.2%, 82.6% ± 13.9% and 74.6% ± 15%, respectively) throughout the course of study. So, a robust and stablemyoelectric control can be designed on the basis of the best performing hand motions. With improved recognition and uniform gain in performance, the deep learning-based approach has the potential to be a more robust alternative to traditional machine learning algorithms.

2.6 OTHER EXISTING SYSTEMS

Different EMG-based recognition techniques using various techniques such as ANN, SVM, RF and LR and their statistical comparisons were performed by Kyung Hyun Lee, Ji Young Min and Sangwon Byun where classification using ANN showed the most accuracy by just considering the TD features. ANN was confirmed to be the least affected by individual variability. Future works include a more heterogeneous population to see the extent to which the ANN algorithm holds good for recognition. [3] EMG showed optimism in terms of finger gesture recognition. The sensing and Classification based FGR model was proposed by Jianting Fu, Shizhou Cao, Linqin Cai and Lechan Yang where the Sensing

The existing systems help us in using vocal recognition techniques for appliance control whichis a tedious process to be used by a mute person. Hence wearable devices within proximity of the appliance were developed. This was a hhindrancefor blind people as they could not be withinthe proximity of the appliance they wish to operate. was carried out by EMG signals and the Classification was carried out by a well-trained CNN algorithm. The thesis lacked to show its adoption with latent factor analysis, cognitive computing and attention mechanism. The innovation of computer vision techniques was briefly explored by Munir Oudah, Ali Al-Naji and Javaan Chahl. CV techniques have their drawbacks but establishing a more reliant hardware device along with it has proven to be more accurate and reliable.[3] Hence, Zhi-Hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang, and Yu-Bo Yuan contributed to work with finger segmentation algorithms by sending real-time gestures captured using a camera to CV frameworks. Its accuracy depended on its ability to ignore background frameworks and detect the palm and fingers separately. This module addressed all of the cons in the above-mentioned papers but lacked to show a reliable accuracy rate when implemented

2.7 CONCLUSION

This chapter elucidates previously published papers and existing projects that utilizes gesture recognition using various technologies, along with their drawbacks. It provides an overview of the project and the technology that has been used in these papers. It provides an insight about the merits and demerits of the project. This chapter also provides the accuracy ofthe different models that has been discussed

CHAPTER – 3

PROPOSED SYSTEM

3.1 HARDWARE COMPONENTS USED

Logitech C615 HD

Fig 3.1- a simple webcam module

A webcam is a video camera that feeds or streams an image or video in real-time to or through a computer network, such as the Internet. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. As a Web camera is a common component and has better clarity than a common Pi Camera it has been chosen. The Logitech C615 webcam is an HD camera that can capture images at 1080p. As Webcam's are highly common and efficient compared to traditional cameras such as Raspberry cameras. The module captures images in JPEG format and fed to the raspberry pi module. The Features includes

- 1080p video recording and 720p video steaming with 30 fps and has about 78-degree field of view.
- 8Mega pixel photos.
- Consists of glass element lens with auto focus feature
- Onboard ESP32-S module, supports WiFi + Bluetooth
- OV2640 camera with flash
- Onboard TF card slot, supports up to 4G TF card for data storage
- Supports WiFi video monitoring and WiFi image upload
- Supports multi-sleep modes, deep sleep current as low as 6mA

Raspberry pi

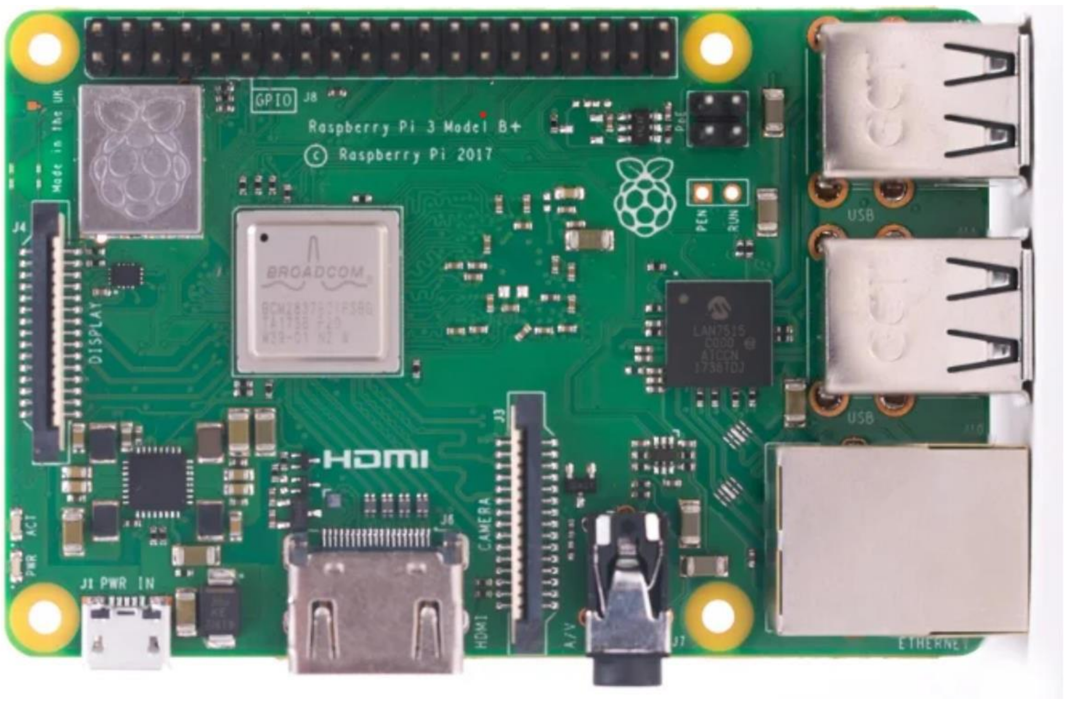


Fig 3.2 – Raspberry pi module

Raspberry pi is a microprocessor used for the programming aspects due to its compactness**.** The minicomputer can receive data from the sensors and pre-process it to the program criterion to generate and produce the desired output. The board consists of two 5V pins, two 3V pins, and 9 ground pins (0V), which are unconfigurable.

5V: The 5v pins directly deliver the 5v supply coming from the mains adaptor. This pin can useto power up the Raspberry Pi, and it can also use to power up other 5v devices.

3V: The 3v pin is there to offer a stable 3.3v supply to power components and to test LED

GND: Ground is commonly referred to as GND. All the voltages are measured with respect to the GND voltage.

Raspberry Pi 3 Input/Outputs pins:

A GPIO pin that is set as an **i**nput will that will allow a signal to be received by the Raspberry Pi by a device connected to this pin. A voltage between 1.8V and 3.3V will be read by the Raspberry Pi as HIGH and if the voltage is lower than 1.8V will be read as LOW.

A GPIO pin set as an output pin sends the voltage signal as high (3.3V) or low (0V). When thispin is set to HIGH, the voltage at the output is 3.3V and when set to LOW, the output voltage is 0V. Along with the simple function of input and output pins, the GPIO pins can also performa variety of alternative functions. Some specific pins used for the model construction are

SPI PIN

SPI (Serial Peripheral Interface) is another protocol used for master-slave communication. It isused by the Raspberry pi board to quickly communicate between one or more peripheral devices. Data is synchronized using a clock (SCLK at GPIO11) from the master (RPi) and the data is sent from the Pi to our SPI device using the MOSI (Master Out Slave In) pin. If the SPIdevice needs to communicate back to Raspberry Pi, then it will send data back using the MISO (Master In Slave Out) pin. There are 5 pins involved in SPI communication:

GND: Connect all GND pins from all the slave components and the RaspberryPi 3 board together.

SCLK: Clock of the SPI. Connect all SCLK pins together.

MOSI: It stands for Master Out Slave In. This pin is used to send data from themaster to a slave

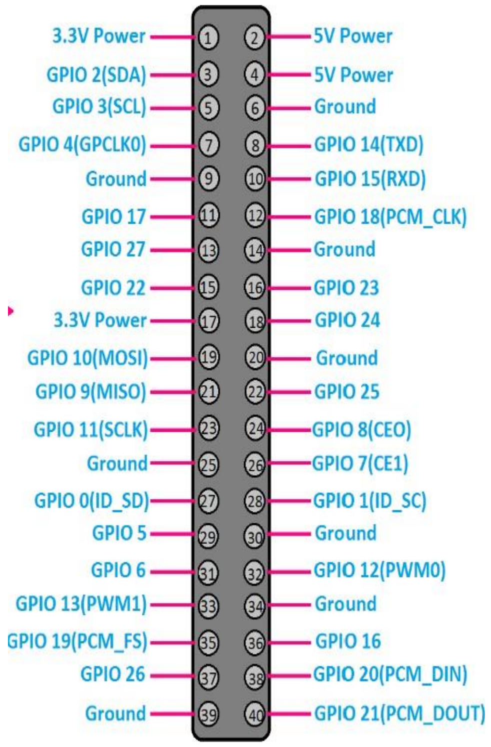


Fig 3.3 – Raspberry pi pin diagram

3.2 SOFTWARE USED:

The following software which are open sourced are used for the simulation of the program code with its hardware devices.

- **Arduino IDE:** ESP8266 and ESP32 Camera utilize Arduino IDE as their programming IDE. Arduino IDE isused to input the required IP to which the devices are connected and communicated.
- **Python**: Neural networks are programmed using python. The captured image is processed using Pythonlanguage.
- **Tensorflow:** This is a neural network framework which keeps tracks of the network graphs and subgraphs.It handles how the weights of the hidden layers are formulated with the input image pixels

3.3 WORK FLOW

An IoT-based camera is placed on the user’s hand, which actively transmitsthe captured video to a computer server. This server runs a multi-layer convolutional neural network. First, the neural network searches for the presence of the user's hand inthe image feed. The network is a TensorFlow Lite-based neural network which uses theSingle Shot detector that scales down the received image to 192, 192 pixels and then classifies if the image consists of a palm. Once a palm is found, another neural networkis activated that returns the key points of the hand above the wrist, including the relativelocation of individual fingers. This neural network is also built using the TensorFlow Lite framework and it returns 21 three-dimensional coordinates of the hand. Using relative cartesian distance the orientation of the fingers is figured. With this data, the algorithm can predict which number the user is holding. Based on previously paired, numbers to device data that specific device is activated using the IoT platform.

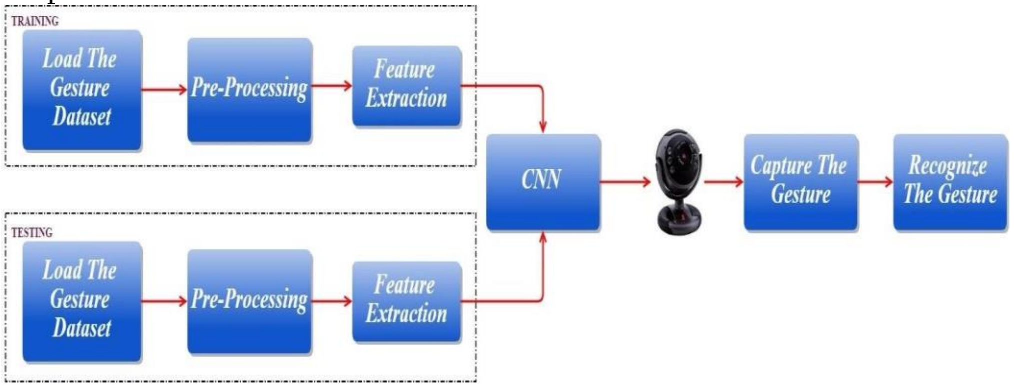


Fig 3.4 system model

The workplan of the project involves of around few major phases: set-up phase, testing phase, working phase, advancement phase

- The set-up phase of the project involves with training of a neural network model that can read, recognize
- The testing phase is to associate the gestures with specific functions such as ON/OFF,Volume UP/DOWN etc
- The working phase involves setting up an internet camera module that is wearable by an impaired person where the gesture made is captured and is used for controlling the appliances which is connected on a network same as the appliance
- In the advancement phase the aim is to make the device cost efficient for easy feasibility.

Convolutional Neural Networks (CNN)

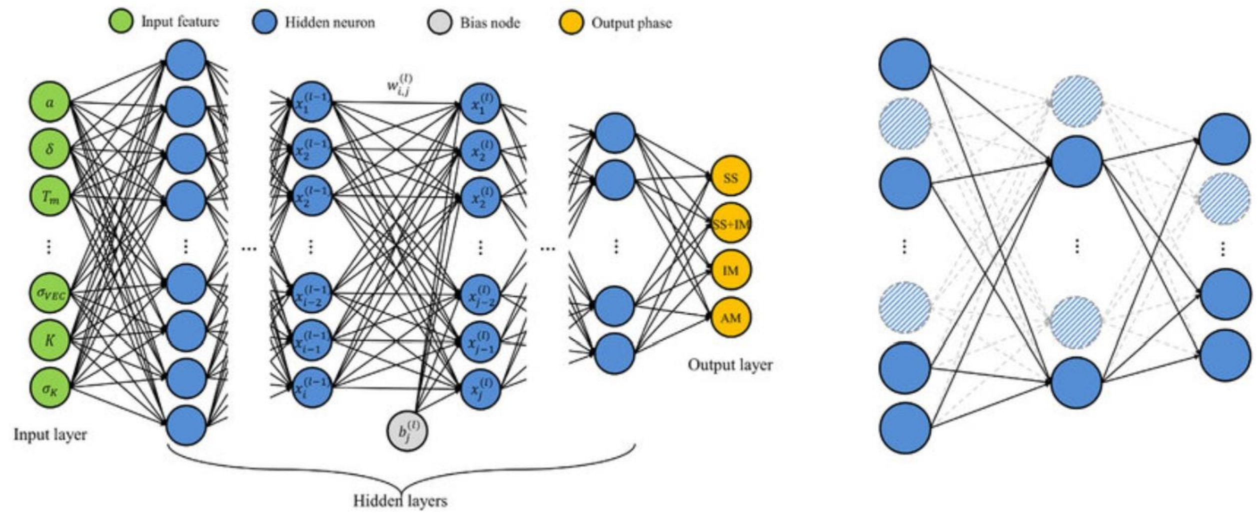
Convolutional Neural Network (CNN) is the extended version of artificial neural networks (ANN) which is predominantly used to extract the feature from the grid-like matrix dataset. One of the most popular deep neural networks is Convolutional Neural Networks (also known as CNN or ConvNet) in deep learning, especially when it comes to Computer Vision applications. It uses a special technique called Convolution. In mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other. The role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction. Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer.

The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. Based on the activation map of the final convolution layer, the classification layer outputs a set of confidence scores (values between 0 and 1) that specify how likely the image is to belong to a “class.” In a regular Neural Network there are three types of layers:

Input Layers: It’s the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).

Hidden Layer: The input from the Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.

Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.



(a) (b)

Fig 3.5 Schematic diagram of the deep neural network: (a) an architecture of DNN model comprised of input, hidden, and output layers
(b) dropout regularization method that controls the connection of the neurons in the hidden layers

The data is fed into the model and output from each layer is obtained from the above step is called feedforward, we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. The error function measures how well the network is performing. After that, we backpropagate into the model by calculating the derivatives. This step is called Backpropagation which basically is used to minimize the loss.

Convolutional Neural Network (CNN) is the extended version of artificial neural networks (ANN) which is predominantly used to extract the feature from the grid-like matrix dataset. Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers. The Convolutional layer applies filters to the input image to extract features, the Pooling layer down samples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent

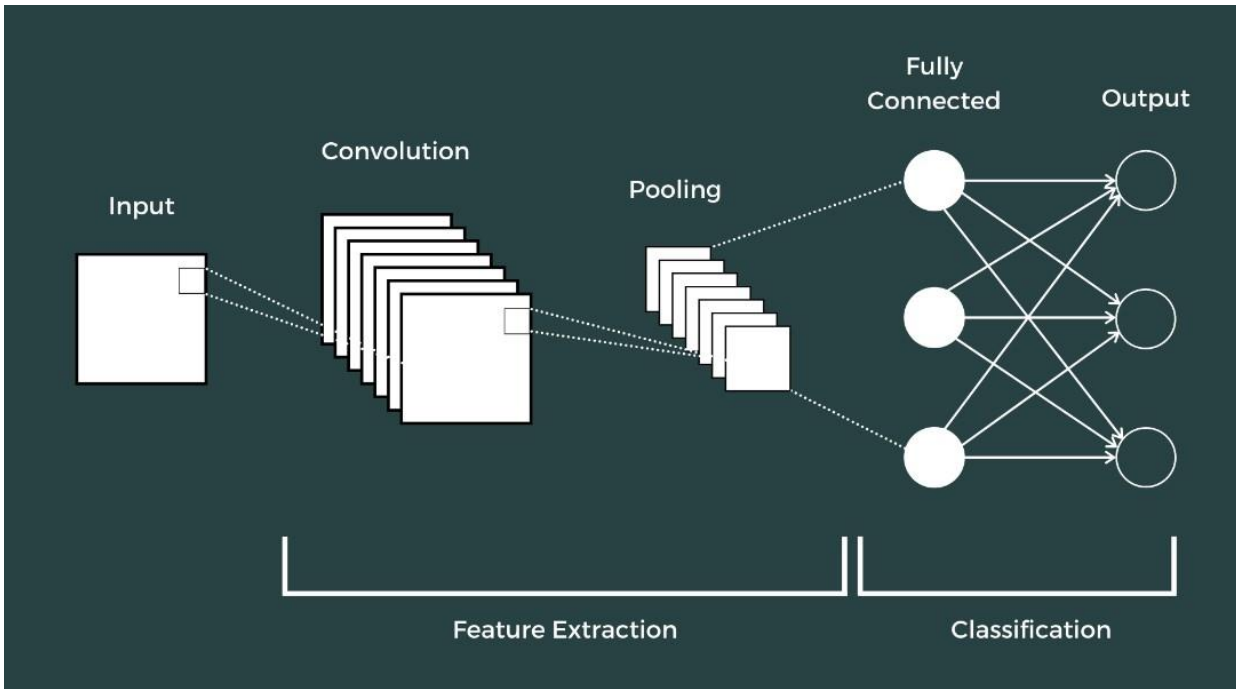


Fig 3.6 – Backpropagation Algorithm

The **Convolutional Layer** is the first layer in a CNN. This layer takes in an input image and performs a series of convolution operations on the image. In other words, the convolutional layer takes in an image tensor as an input, applies a specific number of convolutional filters (kernels) on the image tensor, adds a bias and applies a non-linear activation function (typically, ReLU) to the output. Till now, we had only applied a single kernel to an image tensor but in a convolutional layer, multiple convolutional filters are used and each filter has its own set of pixel values. However, the process of applying the convolution operation is the same for each filter irrespective of the number of filters used. The objective of convolutional layers is to extract patterns and information from an image. The Convolutional filters/kernels at the starting of the network are responsible for capturing the low-level features such as color, gradient orientation, etc. The convolutional filters/kernels deeper down the network are responsible for capturing the high-level features such as edges in the image. The convolution layer is the core building block of the CNN. It carries the main portion of the network’s computational load.

This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in-depth. This means that, if the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all three channels. During the forward pass, the kernel slides across the height and width of the image-producing the image representation of that receptive region. This produces a two-dimensional representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image. The sliding size of the kernel is called a stride. If we have an input of size W x W x D and Dout number of kernels with a spatial size of F with stride S and amount of padding P, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

Fig 3.7– Formula for Convolution Layer

This will yield an output volume of size Wout x Wout x Dout.

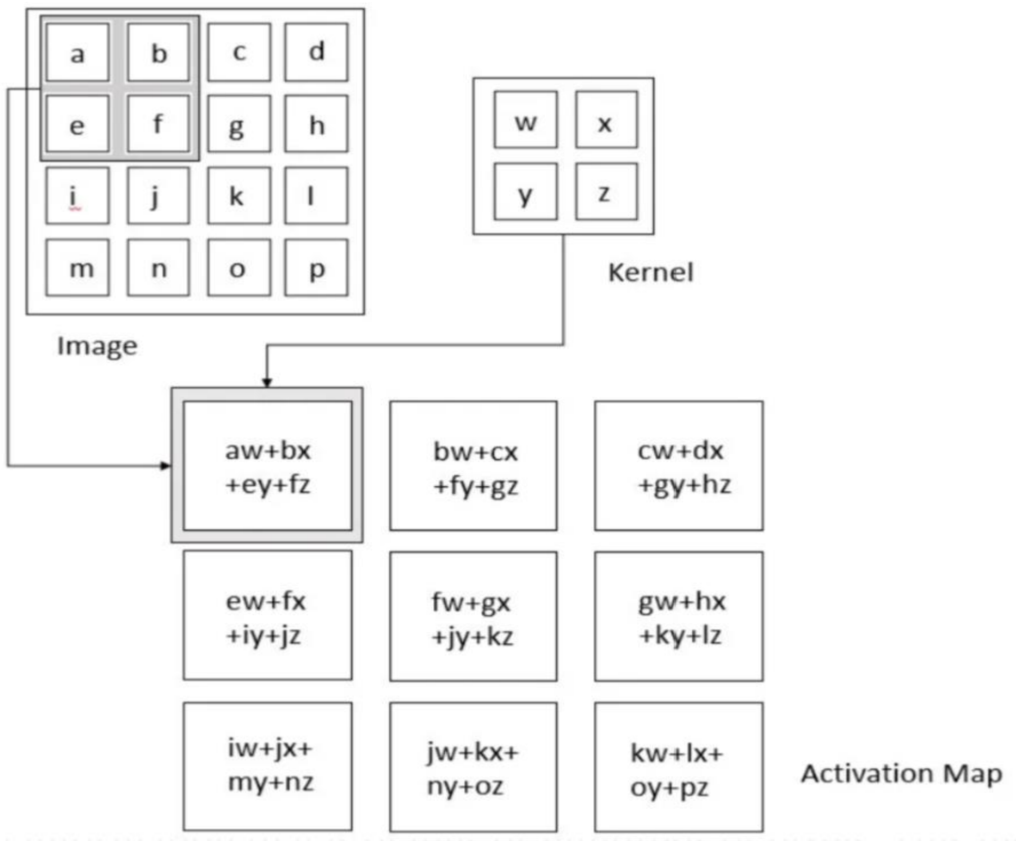


Fig 3.8– Convolution Operation

Convolution leverages three important ideas that motivated computer vision researchers: sparse interaction, parameter sharing, and equivariant representation. Trivial neural network layers use matrix multiplication by a matrix of parameters describing the interaction between the input and output unit. This means that every output unit interacts with every input unit. However, convolution neural networks have sparse interaction. This is achieved by making kernel smaller than the input e.g., an image can have millions or thousands of pixels, but while processing it using kernel, we can detect meaningful information that is of tens or hundreds of pixels. This means that we need to store fewer parameters that not only reduces the memory requirement of the model but also improves the statistical efficiency of the model.

If computing one feature at a spatial point (x1, y1) is useful then it should also be useful at some other spatial point say (x2, y2). It means that for a single two-dimensional slice i.e., for creating one activation map, neurons are constrained to use the same set of weights. In a traditional neural network, each element of the weight matrix is used once and then never revisited, while convolution network has shared parameters i.e., for getting output, weights applied to one input are the same as the weight applied elsewhere.Due to parameter sharing, the layers of convolution neural network will have a property of equivariance to translation. It says that if we changed the input in a way, the output will also get changed in the same way.

The **pooling layer** is responsible for performing a series of pooling operations (typically, max-pooling) on an image. It takes in an image tensor as an input and outputs a tensor after applying the specified pooling operation. The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This helps in reducing the spatial size of the representation, which decreases the required amount of computation and weights. The pooling operation is processed on every slice of the representation individually.

There are several pooling functions such as the average of the rectangular neighbourhood, L2 norm of the rectangular neighbourhood, and a weighted average based on the distance from the central pixel. However, the most popular process is max pooling, which reports the maximum output from the neighbourhood.

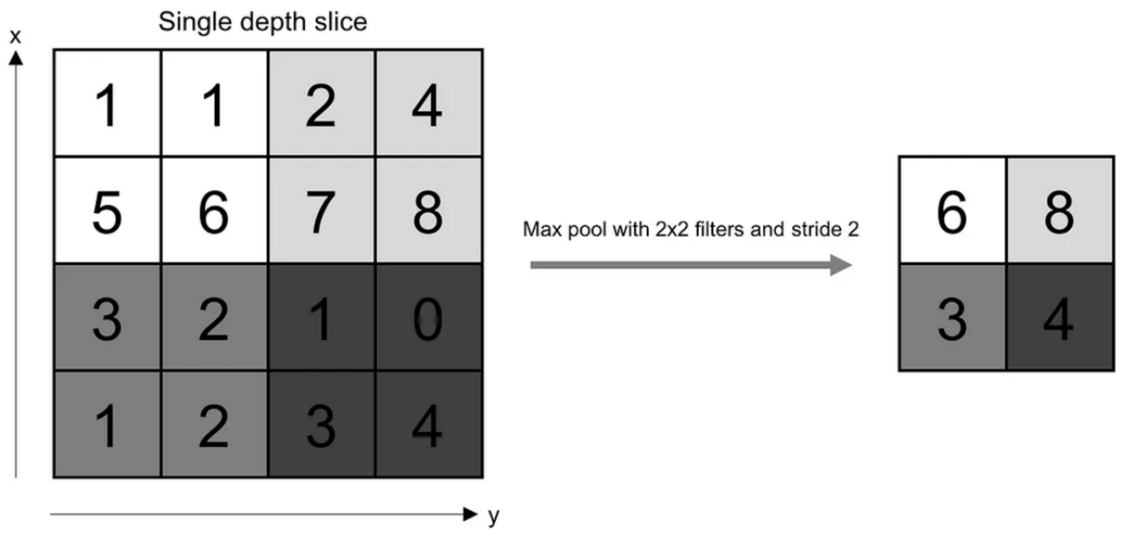


Fig 3.9– Pooling Operation

If we have an activation map of size W x W x D, a pooling kernel of spatial size F, and stride S, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F}{S} + 1$$

Fig 3.10– Formula for Padding Layer

This will yield an output volume of size Wout x Wout x D. In all cases, pooling provides some translation invariance which means that an object would be recognizable regardless of where it appears on the frame.Main objectives of a pooling layer in a CNN can be summarized as:

- **To reduce the computational cost:** A pooling layer reduces the size of image tensor, therefore, reducing the number of parameters and computations required in the network.
- **To make the network more generic:** Pooling helps the network to be more generic because it effectively combines several pixel values into one (max-pooling or average pooling). This decreases the chances of the network being biased towards pixels (over-fitting).

Fully connected layer. The FC layer is where image classification happens in the CNN based on the features extracted in the previous layers. Here, fully connected means that all the inputs or nodes from one layer are connected to every activation unit or node of the next layer.
All the layers in the CNN are not fully connected because it would result in an unnecessarily dense network. It also would increase losses and affect the output quality, and it would be computationally expensive. Neurons in this layer have full connectivity with all neurons in the preceding and succeeding layer as seen in regular FCNN. This is why it can be computed as usual by a matrix multiplication followed by a bias effect. The FC layer helps to map the representation between the input and the output.

Non-Linearity Layers

Since convolution is a linear operation and images are far from linear, non-linearity layers are often placed directly after the convolutional layer to introduce non-linearity to the activation map.

There are several types of non-linear operations, the popular ones being:

1. Sigmoid

The sigmoid non-linearity has the mathematical form $\sigma(\kappa) = 1/(1+e^{-}\kappa)$. It takes a real-valued number and "squashes" it into a range between 0 and 1.However, a very undesirable property of sigmoid is that when the activation is at either tail, the gradient becomes almost zero. If the local gradient becomes very small, then in backpropagation it will effectively "kill" the gradient. Also, if the data coming into the neuron is always positive, then the output of sigmoid will be either all positives or all negatives, resulting in a zig-zag dynamic of gradient updates for weight.

2. Tanh

Tanh squashes a real-valued number to the range [-1, 1]. Like sigmoid, the activation saturates, but unlike the sigmoid neurons its output is zero centred.

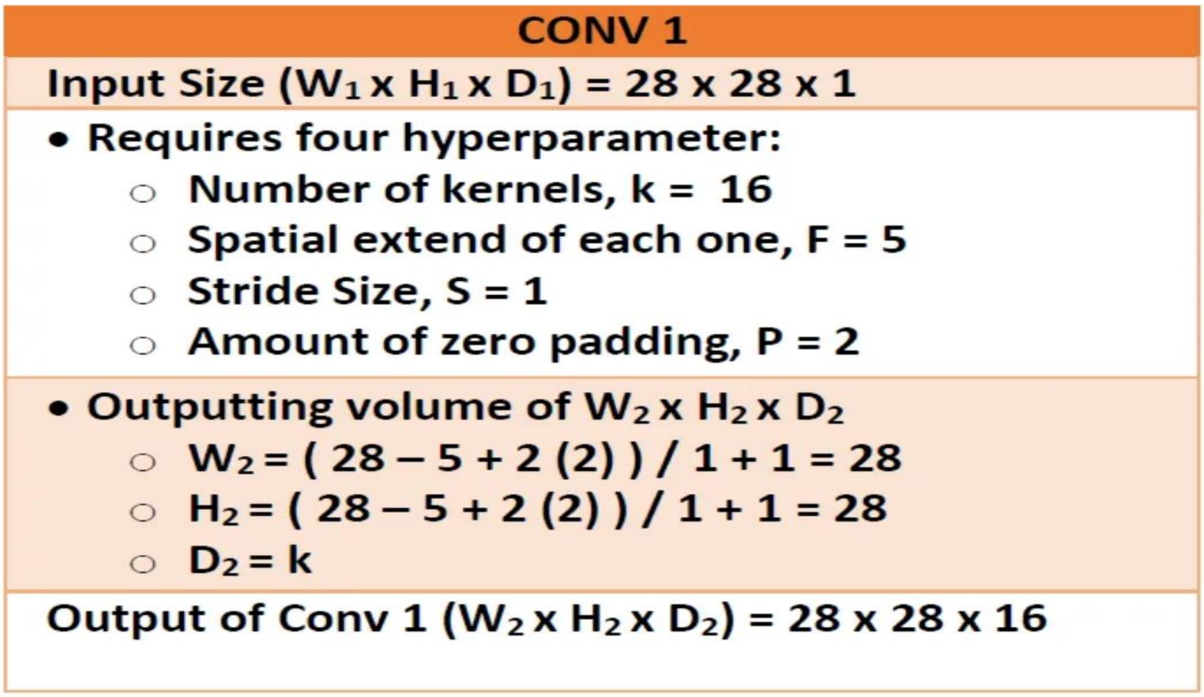
3. ReLU

The Rectified Linear Unit (ReLU) has become very popular in the last few years. It computes the function $f(\kappa)=\max(0,\kappa)$. In other words, the activation is simply threshold at zero.

In comparison to sigmoid and tanh, ReLU is more reliable and accelerates the convergence by six times. Unfortunately, a con is that ReLU can be fragile during training. A large gradient flowing through it can update it in such a way that the neuron will never get further updated. However, we can work with this by setting a proper learning rate. The following way is formulated to design the neural network which has architecture as follows:

[INPUT]→[CONV 1] → [BATCH NORM] → [ReLU] → [POOL 1]
→ [CONV 2] → [BATCH NORM] → [ReLU] → [POOL 2]
→ [FC LAYER] → [RESULT]

For both conv layers, we will use kernel of spatial size 5 x 5 with stride size 1 and padding of 2. For both pooling layers, we will use max pool operation with kernel size 2, stride 2, and zero padding.



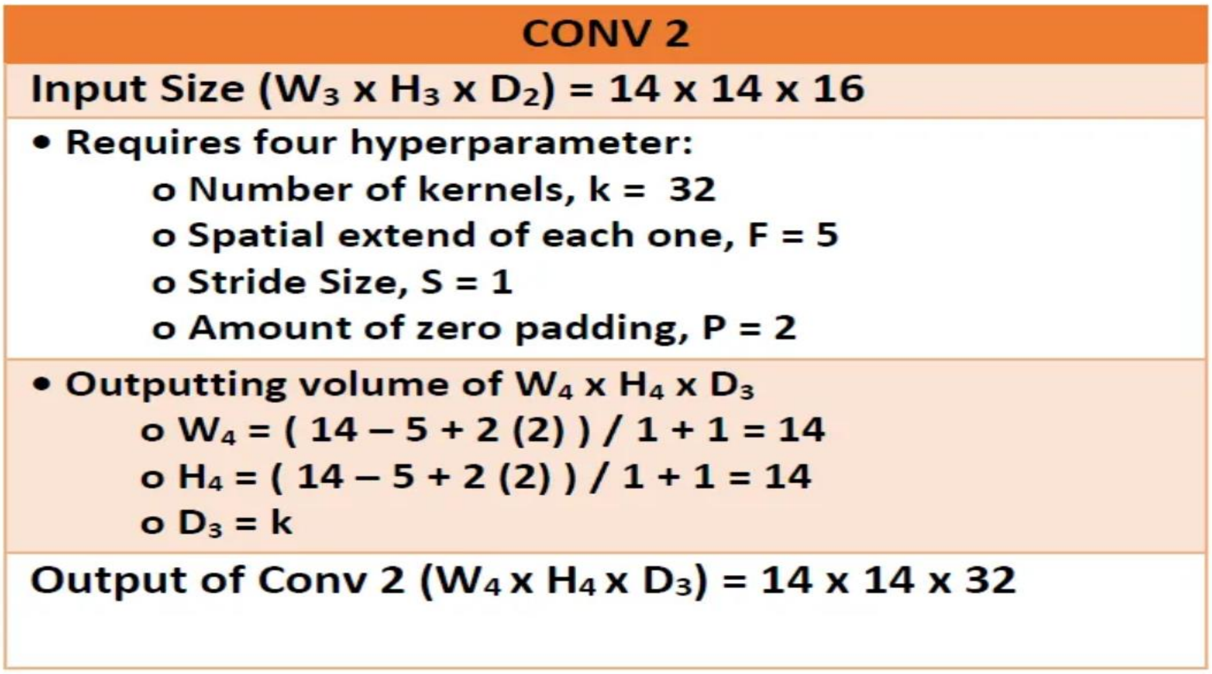
CONV 1

Input Size (W_1 x H_1 x D_1) = 28 x 28 x 1

- Requires four hyperparameter:
  - Number of kernels, k = 16
  - Spatial extend of each one, F = 5
  - Stride Size, S = 1
  - Amount of zero padding, P = 2
- Outputting volume of W_2 x H_2 x D_2
  - W_2 = (28 – 5 + 2 (2)) / 1 + 1 = 28
  - H_2 = (28 – 5 + 2 (2)) / 1 + 1 = 28
  - D_2 = k

Output of Conv 1 (W_2 x H_2 x D_2) = 28 x 28 x 16

Fig 3.11–Calculations for Conv 1 Layer



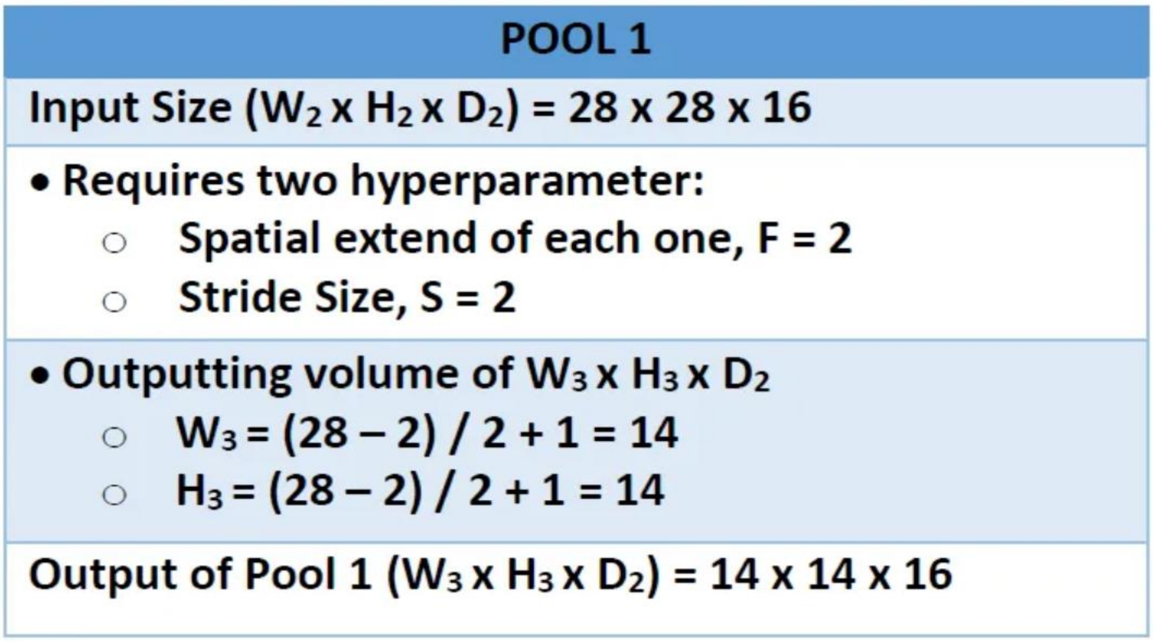
CONV 2

Input Size (W_3 x H_3 x D_2) = 14 x 14 x 16

- Requires four hyperparameter:
  - Number of kernels, k = 32
  - Spatial extend of each one, F = 5
  - Stride Size, S = 1
  - Amount of zero padding, P = 2
- Outputting volume of W_4 x H_4 x D_3
  - W_4 = (14 – 5 + 2 (2)) / 1 + 1 = 14
  - H_4 = (14 – 5 + 2 (2)) / 1 + 1 = 14
  - D_3 = k

Output of Conv 2 (W_4 x H_4 x D_3) = 14 x 14 x 32

Fig 3.12–Calculations for conv 2 layer



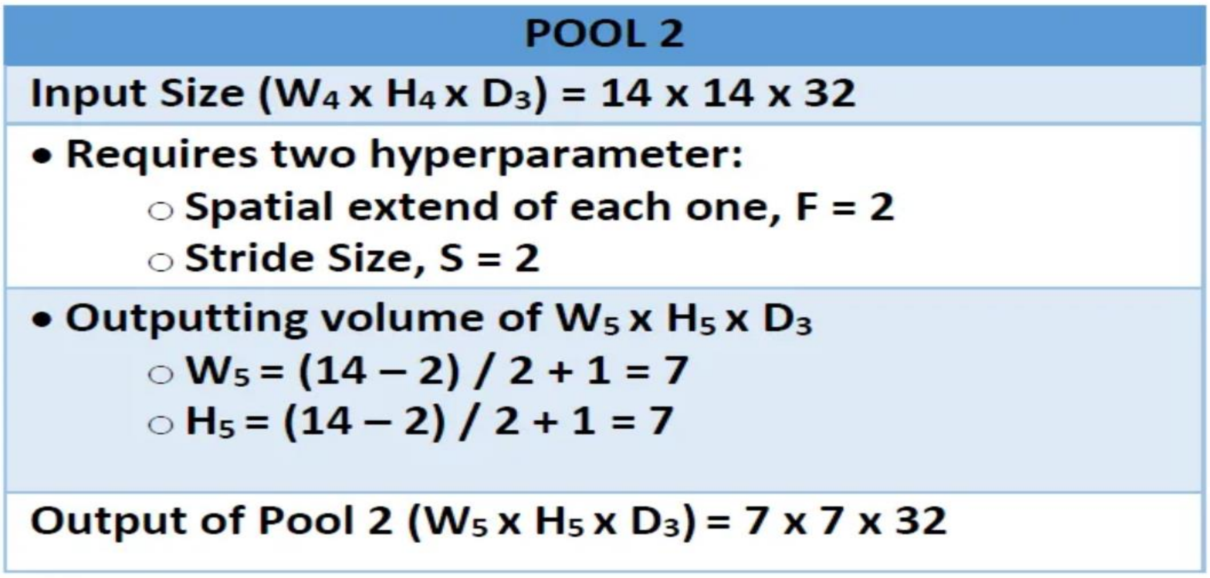
POOL 1

Input Size (W_2 x H_2 x D_2) = 28 x 28 x 16

- Requires two hyperparameter:
  - Spatial extend of each one, F = 2
  - Stride Size, S = 2
- Outputting volume of W_3 x H_3 x D_2
  - W_3 = (28 – 2) / 2 + 1 = 14
  - H_3 = (28 – 2) / 2 + 1 = 14

Output of Pool 1 (W_3 x H_3 x D_2) = 14 x 14 x 16

Fig 3.13–Calculations for Pool1 Layer



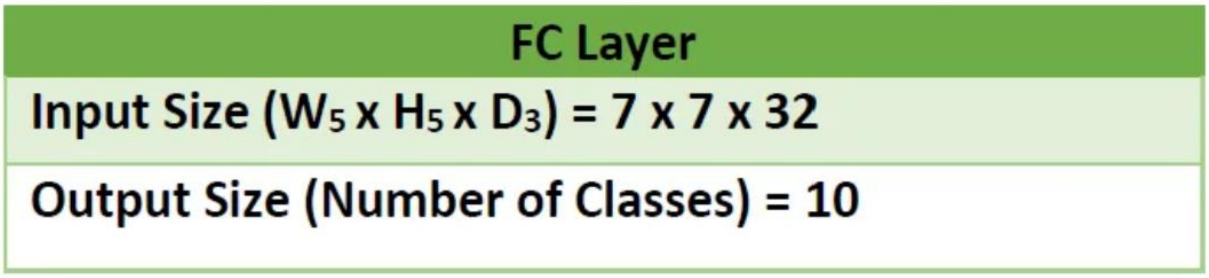
POOL 2

Input Size (W_4 x H_4 x D_3) = 14 x 14 x 32

- Requires two hyperparameter:
  - Spatial extend of each one, F = 2
  - Stride Size, S = 2
- Outputting volume of W_5 x H_5 x D_3
  - W_5 = (14 – 2) / 2 + 1 = 7
  - H_5 = (14 – 2) / 2 + 1 = 7

Output of Pool 2 (W_5 x H_5 x D_3) = 7 x 7 x 32

Fig 3.13 Calculations for Pool2 Layer



FC Layer
Input Size (W_5 x H_5 x D_3) = 7 x 7 x 32
Output Size (Number of Classes) = 10

Fig 3.14 Size of Fully Connected Layer

3.4 WORKFLOW:

An internet camera is used to capture the real time images of any gesture made by a visually or verbally impaired person from any moment of time and space. These images are further uploaded to the cloud server using the inter connectivity of a local network. The webcam captures the gestures using a deep learning model trained for capturing the palm detection whichfurther converts all the 3-D vector images into a 2-D vector plane which helps us in capturing the key features of the palm. This way a reference point in a palm is taken and all the vector distances are captured such that the user’s finger signs are recognized. Any device connected on the local network of the internet camera module is eligible to send an API request for pairing of the devices with the cloud server. The devices after authorizing the request received generate an output function with respect to the gesture pointed at the camera module. This wayany impaired person is able to operate and control any device connected to the local network by associating the gesture with the function they want to perform on the device**.**

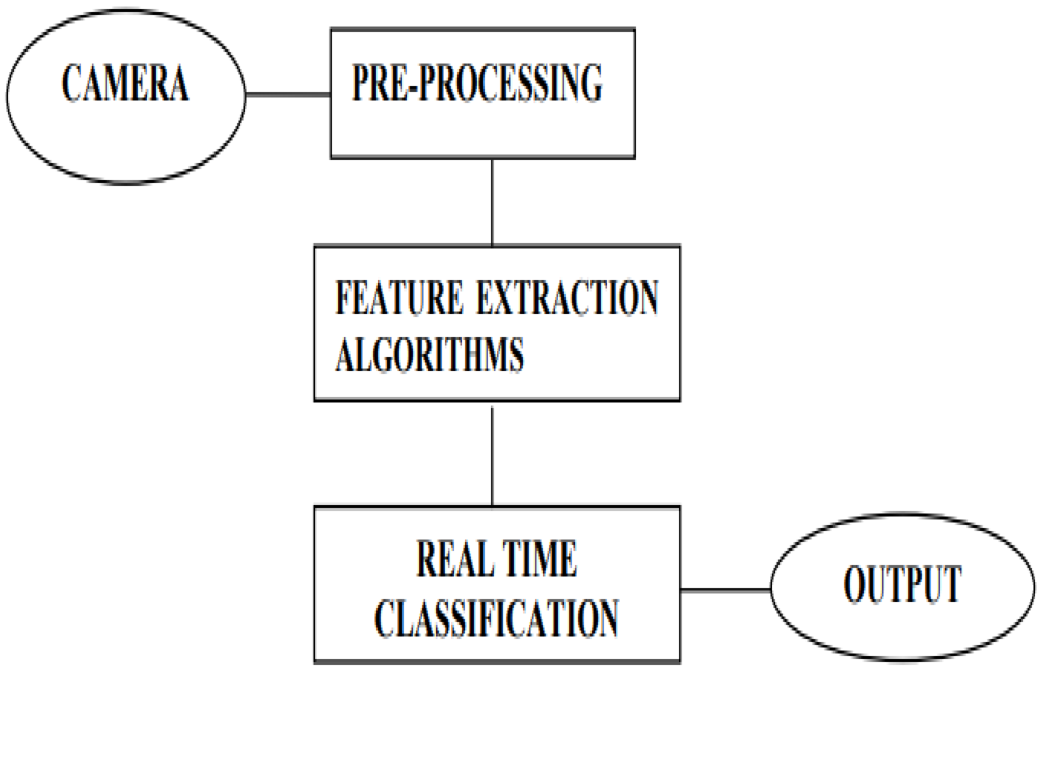


Fig 3.15 System implementation

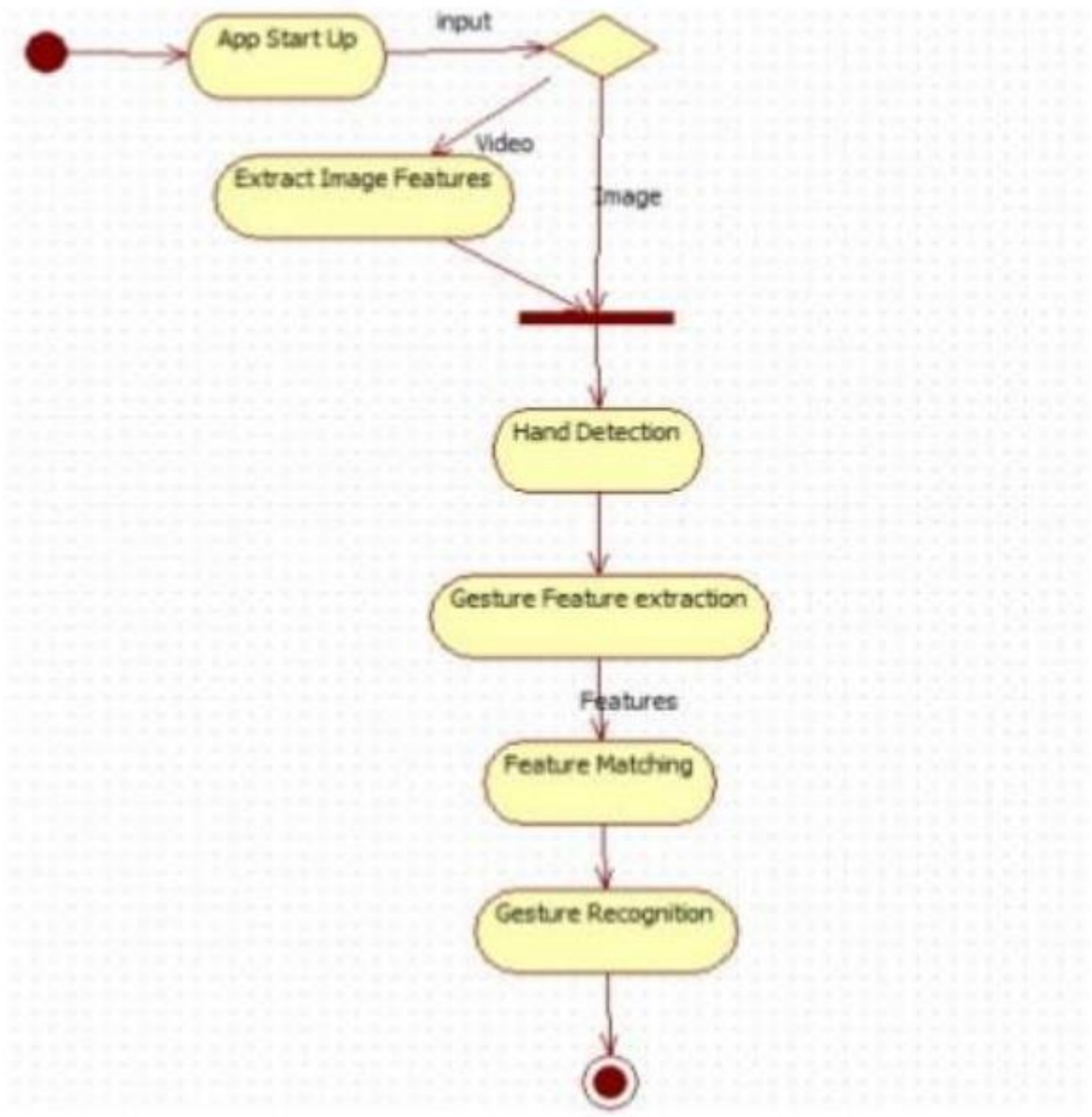


Fig 3.16 Activity Diagram

Recognition of hand Gestures includes various activities to be performed. As shown in the figure the first activity is to start the camera to capture the image. This activity automatically invokes the camera using the OpenCV library in Python as the execution of the program starts. Then on the basis of the captured image, the gesture information is extracted. This information is used to extract the features such as contours, convex hull and the point of the defect based on which the number of fingers in front of the camera is Recognized. As the Finger information is extracted the application automatically performs the action

The shown Sequence diagram explains the flow of the program. As this System is based on Human-Computer Interaction so it basically includes the user, computer and the medium to connect both digitally which is a web camera. As the execution of the program starts, it first invokes the web camera to take an RGB image of the hand. Then the image is segmented and filtered to reduce the noise in the image. After the removal of the noise the hand gestures are detected i.e. the number of fingers raised is preprocessed, based on the feature are extracted. After Feature extraction is done, by using conditionals statements in the program the feature are matched. As the features match, it automatically controls the media player and gives us the required results.

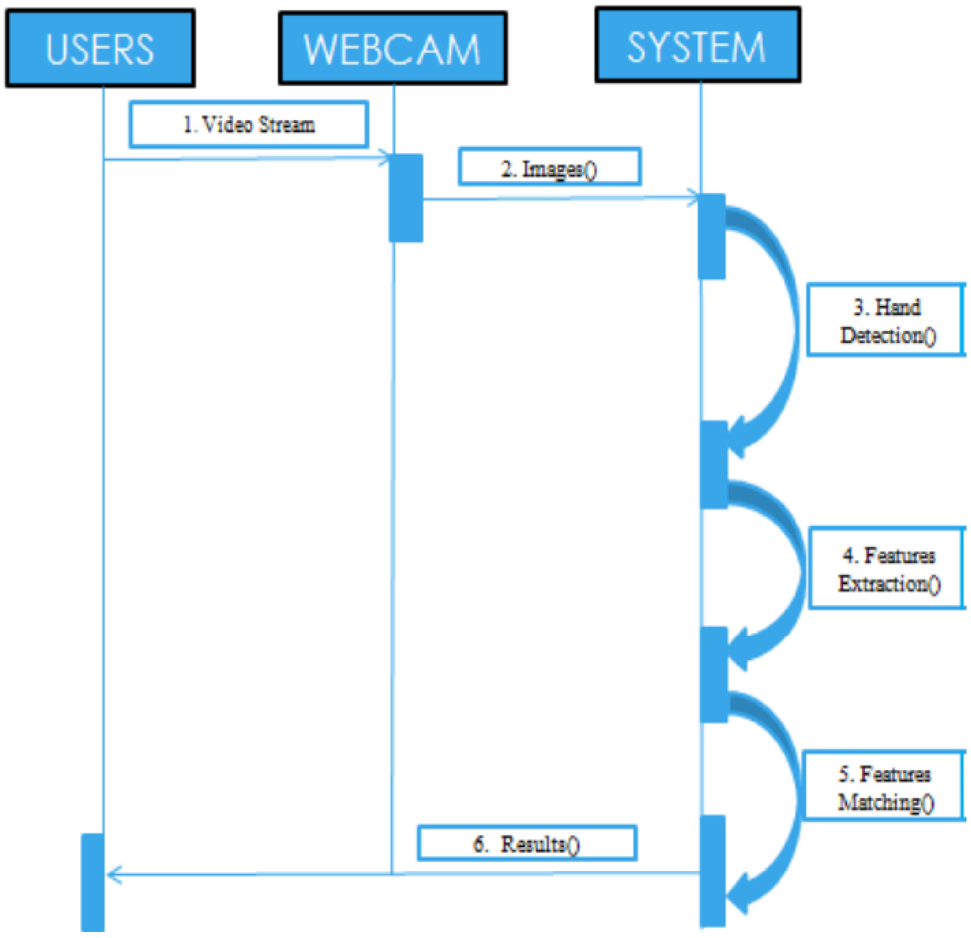


Fig 3.17 Sequence diagram

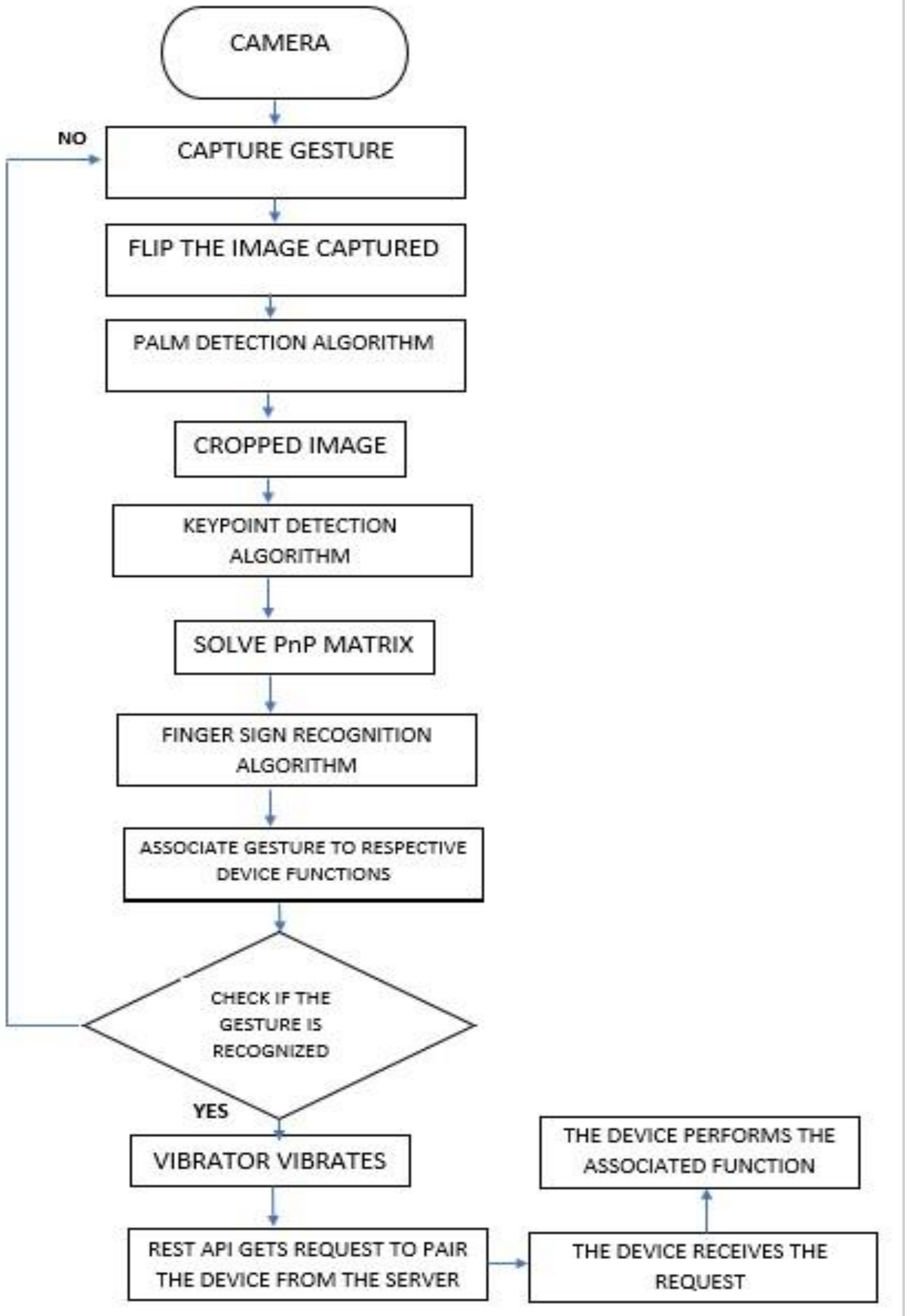

CAMERA
CAPTURE GESTURE
NO
FLIP THE IMAGE CAPTURED
PALM DETECTION ALGORITHM
CROPPED IMAGE
KEYPOINT DETECTION ALGORITHM
SOLVE PnP MATRIX
FINGER SIGN RECOGNITION ALGORITHM
ASSOCIATE GESTURE TO RESPECTIVE DEVICE FUNCTIONS
CHECK IF THE GESTURE IS RECOGNIZED
YES
VIBRATOR VIBRATES
REST API GETS REQUEST TO PAIR THE DEVICE FROM THE SERVER
THE DEVICE RECEIVES THE REQUEST
THE DEVICE PERFORMS THE ASSOCIATED FUNCTION

Fig 3.18-flowchart for the flow of data

ALGORITHM DESCRIPTION

The algorithms explained have been trained and tested for real-time detection and recognition of gestures. The output obtained is further used for appliance management using the cloud server relationship

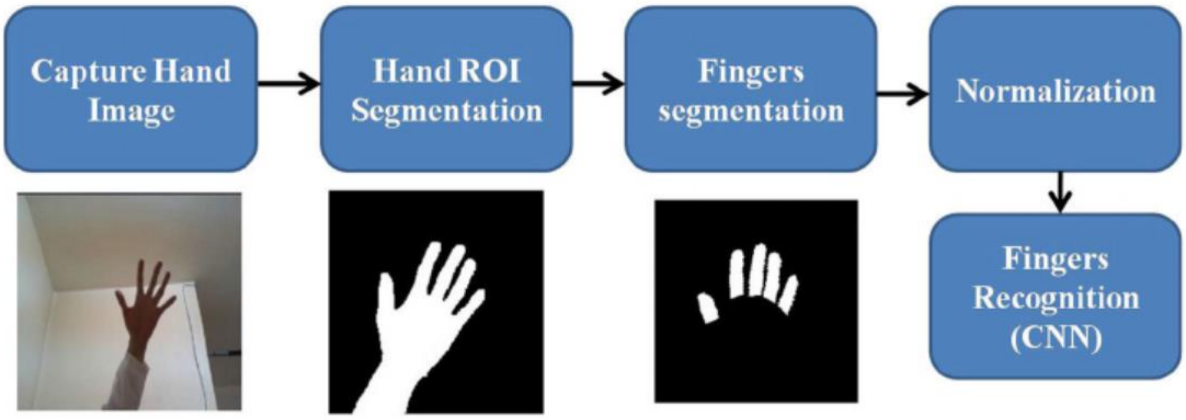


Fig 3.19 Overall Algorithmic output

PALM DETECTION ALGORITHM

The hand region of the image is segmented from the whole image using mask images available in open access dataset. The mask image is inverted, and it is convolved with the hand image which produces convolved image. The gray-level threshold is applied on the convolved image which produced hand region segmented image. The adaptive histogram equalization (AHE) method is used as enhancement method for improving the contrast of each pixel in an image. In the Histogram Equalization method, the entire image region is split into 3 * 3 patterns and the center pixel of each 3 * 3 pattern is contrast-enhanced by accumulating the maximum histogram count to the center pixel. In the case of the AHE method, the 3 * 3 patterns are overlapped with each other and then each center pixel of 3 * 3 patterns is contrast-enhanced.

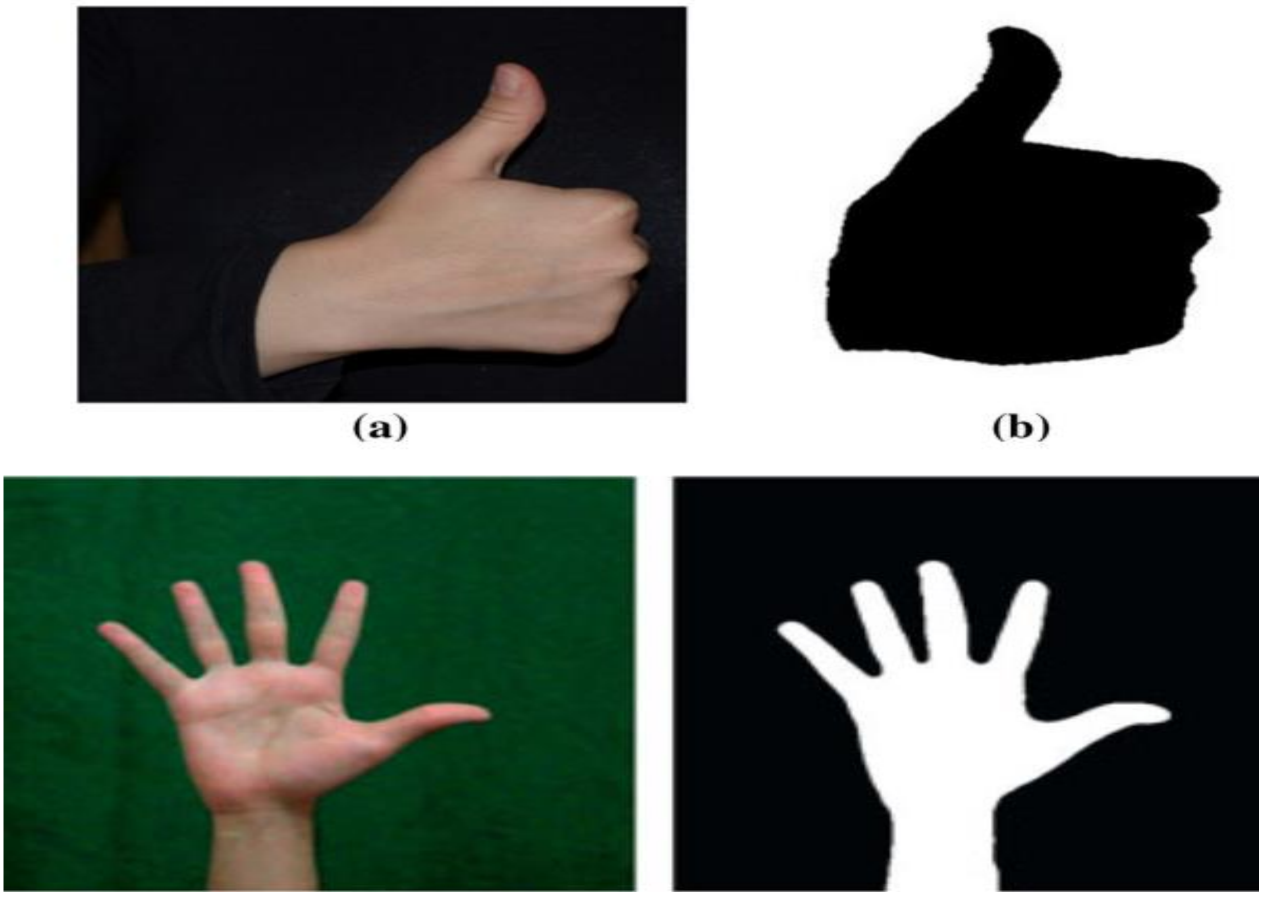


Fig 3.20 a)Hand image and b) mask image

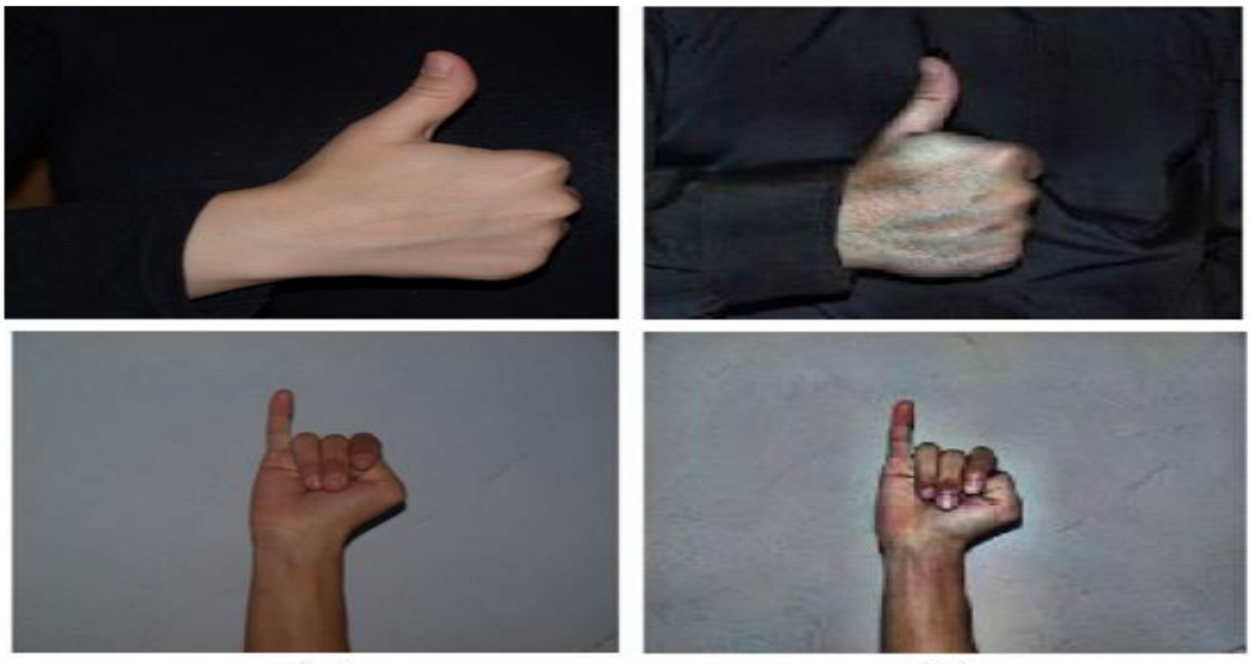


Fig 3.21 a) Source hand gesture images and b) enhanced hand gesture images

THE KEYPOINT DETECTION ALGORITHM:

The model is a convolutional neural network that helps in identifying key features in the palm once detection is finished. The model is trained using thirty-one thousand real-world images provides key-point localization and returns about 21 3-D points out of which the point number 0 is the reference point and the Euclidean distance from the reference point to the points in the fingers is calculated to recognize the gestures.

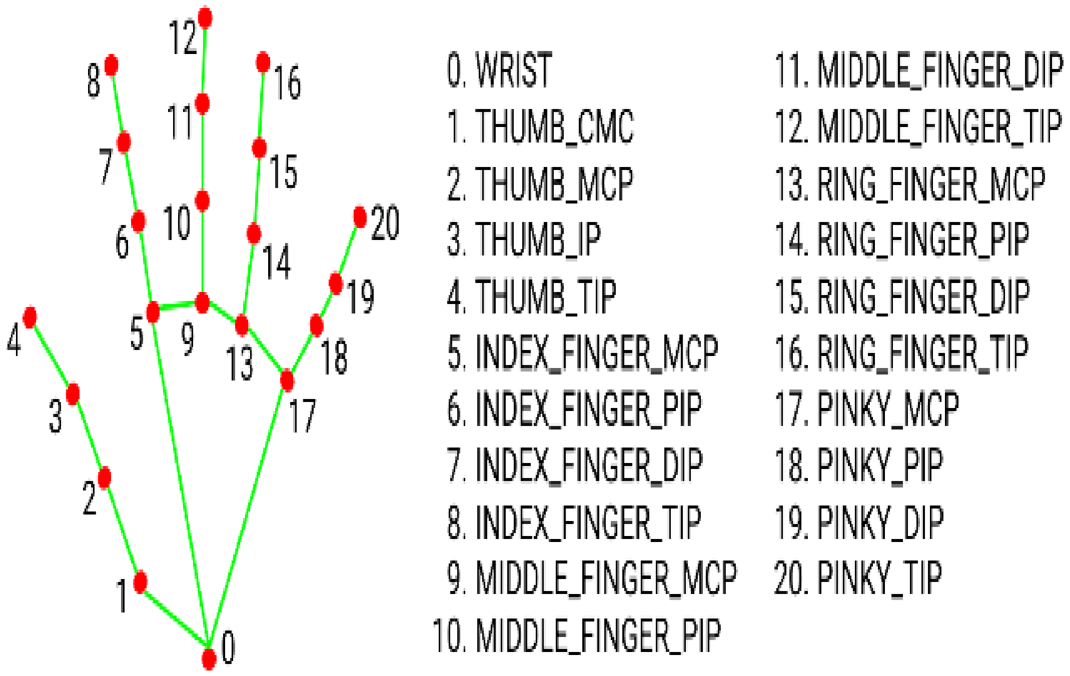


Fig 3.22: key features of the palm

FINGER RECOGNITION MODEL

To recognize the gesture shown the finger recognition model uses the Euclidean distance between the landmarks obtained. The Z axis is omitted to focus on major 2-D gestures

•

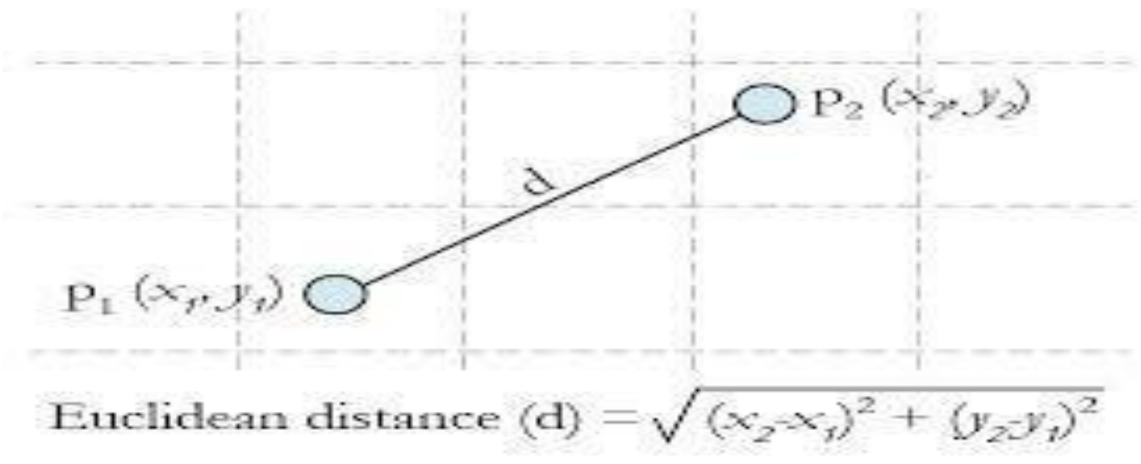


Fig 3.23 Distance mapping using Euclid's formula

For example, concerning fig 3., the distance between points 0-8 and 0-6 is calculated. If a finger is folded, the distance between 0-8 would be lesser than that between 0-6. This way the gesture is identified.

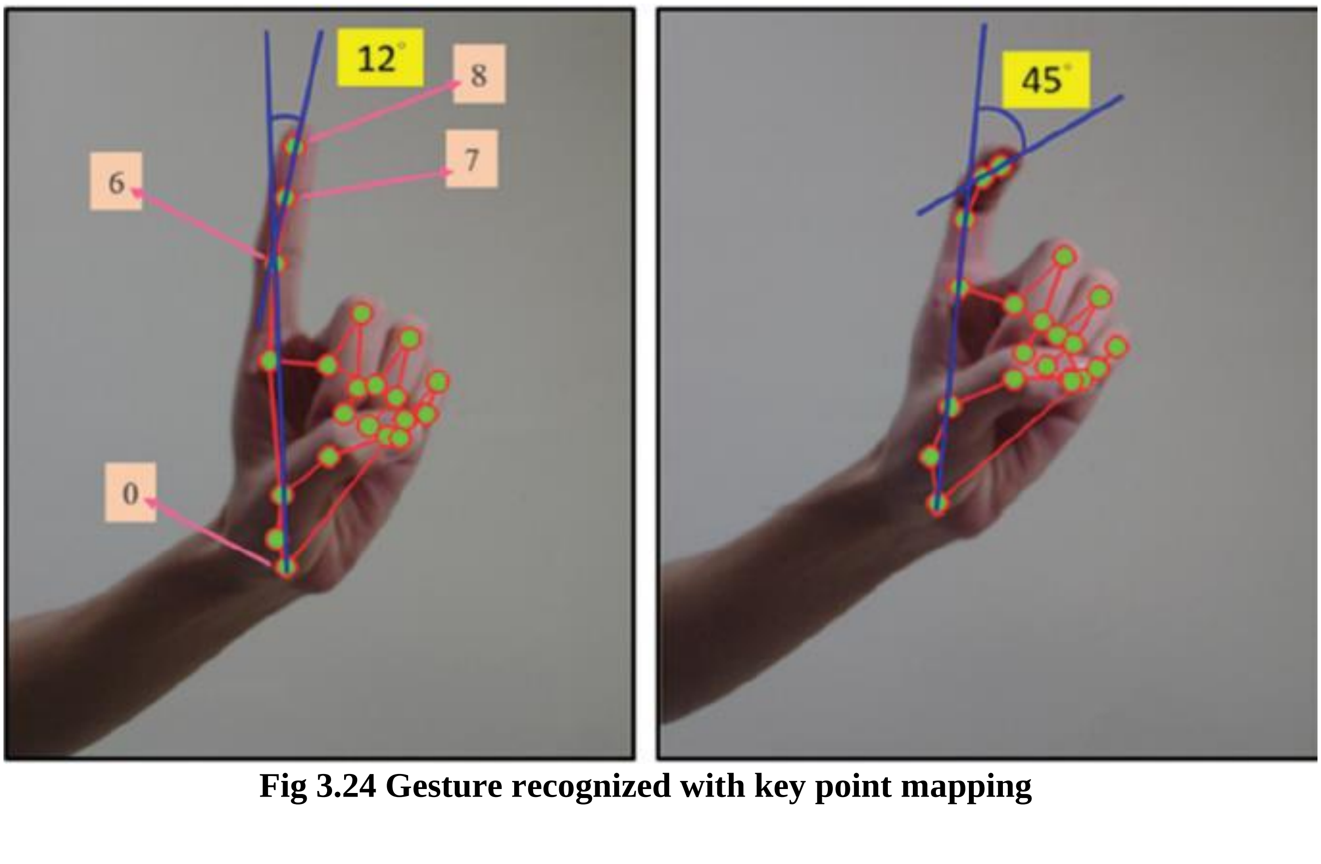


Fig 3.24 Gesture recognized with key point mapping

•

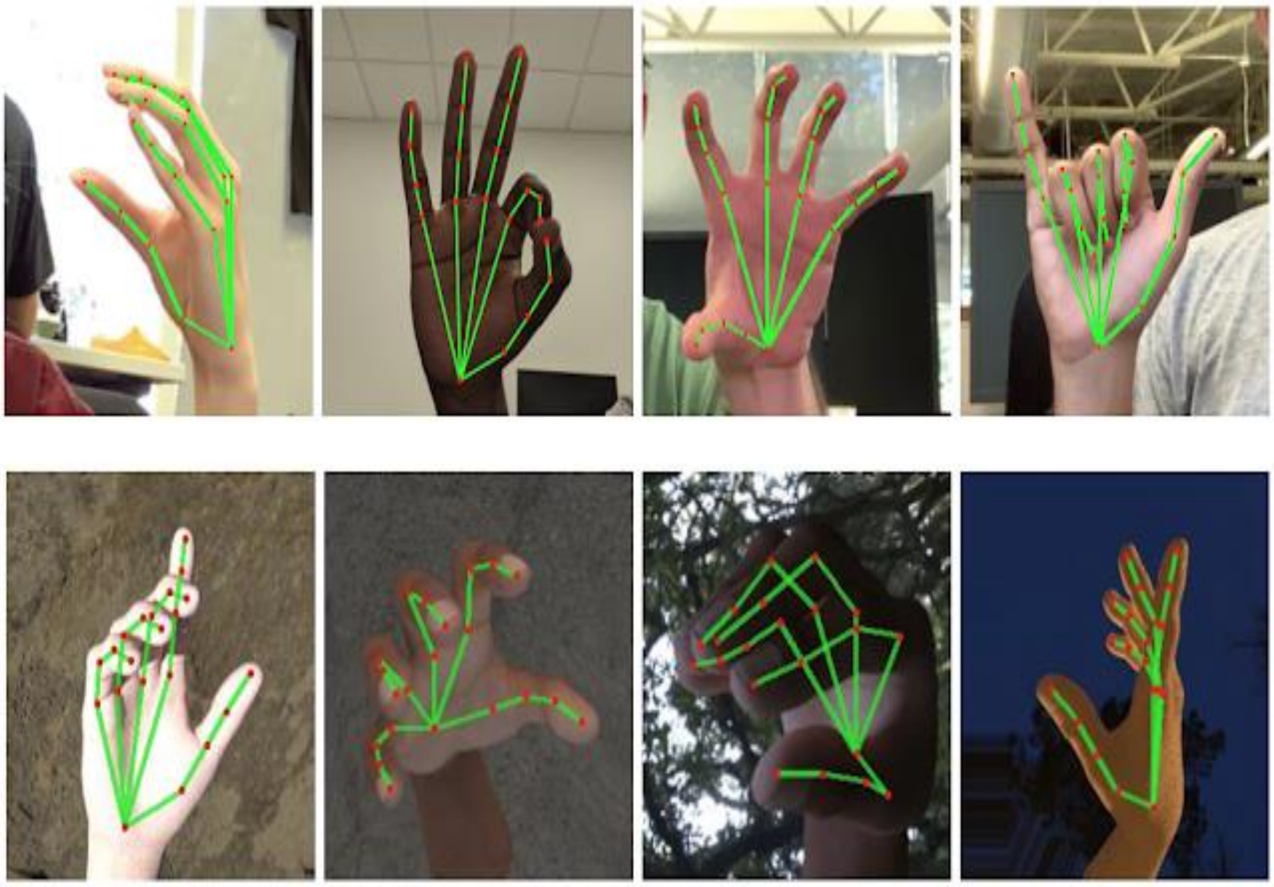


Fig 3.25 Varieties of gestures being marked for keypoint

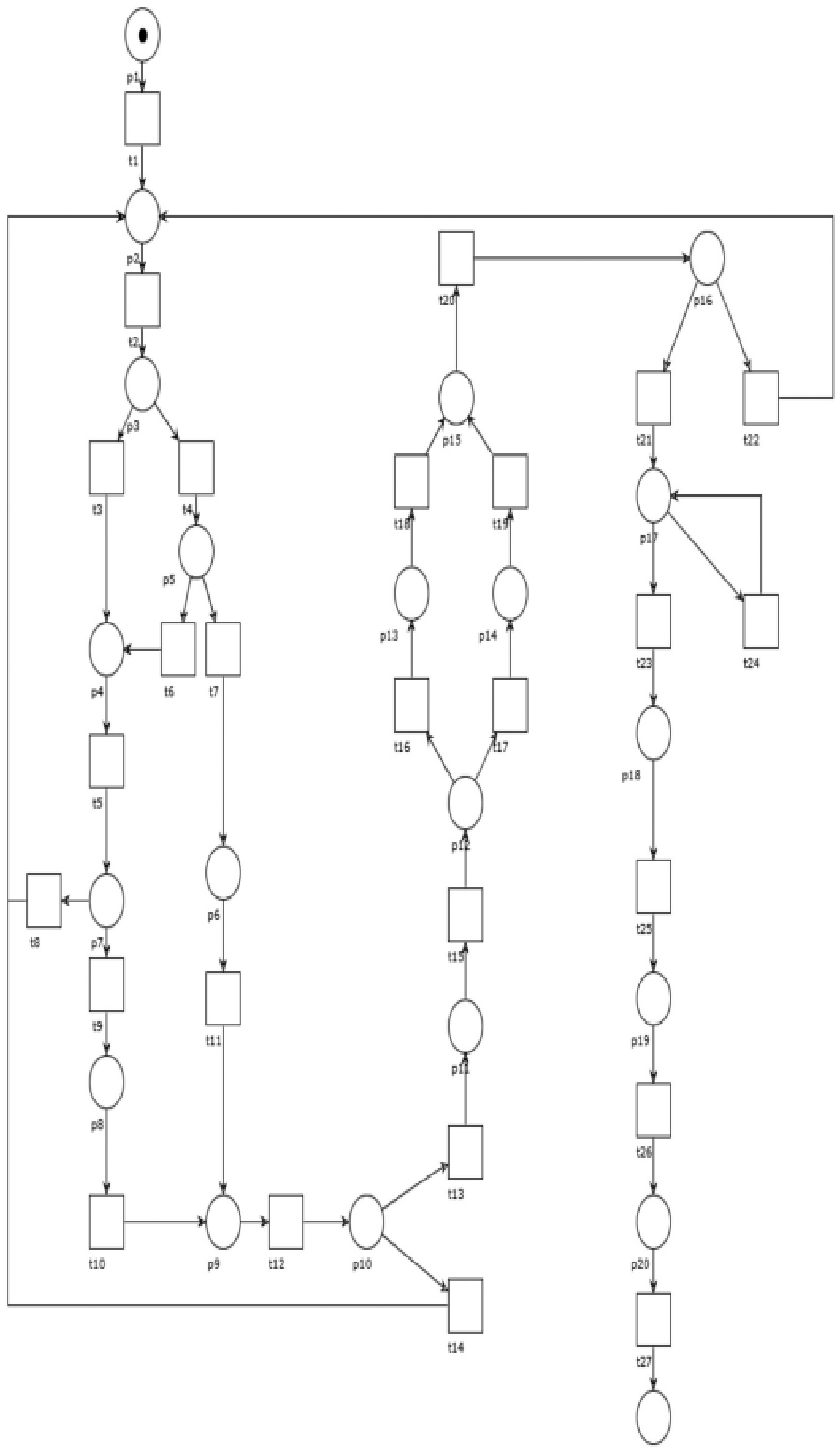


Fig 3.26 CNN layer

REST API

The proposed model uses a client-server architecture to communicate between the paired devices on the same network. The architecture is made up of clients and servers whose requests are managed through HTTP protocols. Data is transferred from Client to Server using the JSON format Modern internet utilises this architecture as it is easy to secure the transaction between the client and server. The paired devices act as servers searching for specific authenticated API keys. When the paired gesture is recognized the model sends a Client GET request to the server of the paired device. This request is received by the server and the device is activated. REST is a set of architectural constraints, not a protocol or a standard. API developers can implement REST in a variety of ways.

When a client request is made via a RESTful API, it transfers a representation of the state of the resource to the requester or endpoint. This information, or representation, is delivered in one of several formats via HTTP: JSON (Javascript Object Notation), HTML, XLT, Python, PHP, or plain text. JSON is the most generally popular file format to use because, despite its name, it’s language-agnostic, as well as readable by both humans and machines. Something else to keep in mind: Headers and parameters are also important in the HTTP methods of a RESTful API HTTP request, as they contain important identifier information as to the request's metadata, authorization, uniform resource identifier (URI), caching, cookies, and more. There are request headers and response headers, each with their own HTTP connection information and status codes.

For an API to be considered RESTful, it has to conform to these criteria:

- A client-server architecture made up of clients, servers, and resources, with requests managed through HTTP.
- Stateless client-server communication, meaning no client information is stored between get requests and each request is separate and unconnected.
- Cacheable data that streamlines client-server interactions.

A uniform interface between components so that information is transferred in a standard form. This requires that:

- resources requested are identifiable and separate from the representations sent to the client.

- resources can be manipulated by the client via the representation they receive because the representation contains enough information to do so.
- self-descriptive messages returned to the client have enough information to describe how the client should process it.
- hypertext/hypermedia is available, meaning that after accessing a resource the client should be able to use hyperlinks to find all other currently available actions they can take.

A layered system that organizes each type of server (those responsible for security, load-balancing, etc.) involved the retrieval of requested information into hierarchies, invisible to the client.

Code-on-demand (optional): the ability to send executable code from the server to the client when requested, extending client functionality.

Though the REST API has these criteria to conform to, it is still considered easier to use than a prescribed protocol like SOAP (Simple Object Access Protocol), which has specific requirements like XML messaging, and built-in security and transaction compliance that make it slower and heavier.

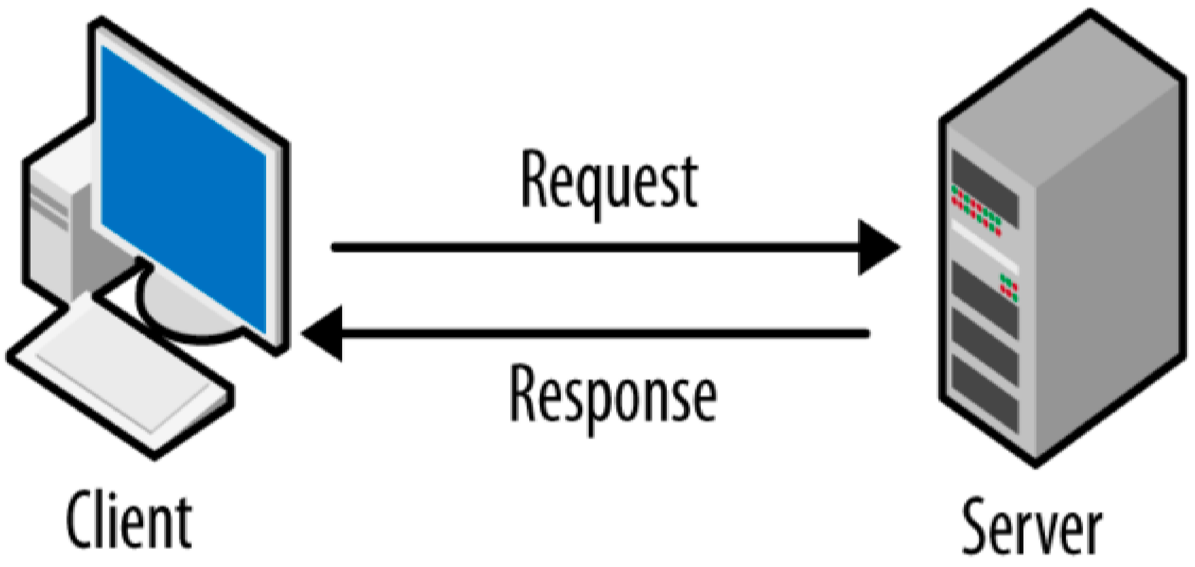


Fig 3.27: Client-Server architecture

CONCLUSION

This chapter shows a skeletal outline of the proposed model. It shows the data capturing, datamanipulation and output representation of the model. It further sheds light into the various hardware components and software used in the proposed model. It also goes into details aboutthe pin diagram and details about the different components that is being used. Furthermore, details about the intricate workings of the proposed model are elucidated. It shows how data isreceived and is manipulated to recognize the gesture and control the appropriate device accordingly.

CHAPTER – 4

RESULT AND ANALYSIS

4.1 GENERAL

This chapter depicts the results obtained which relate the various gestures with respect to theirfunctions and dependent parameters. Further merits and demerits are jotted down and discussedbriefly.

4.2 VERIFICATION OF OUTPUT

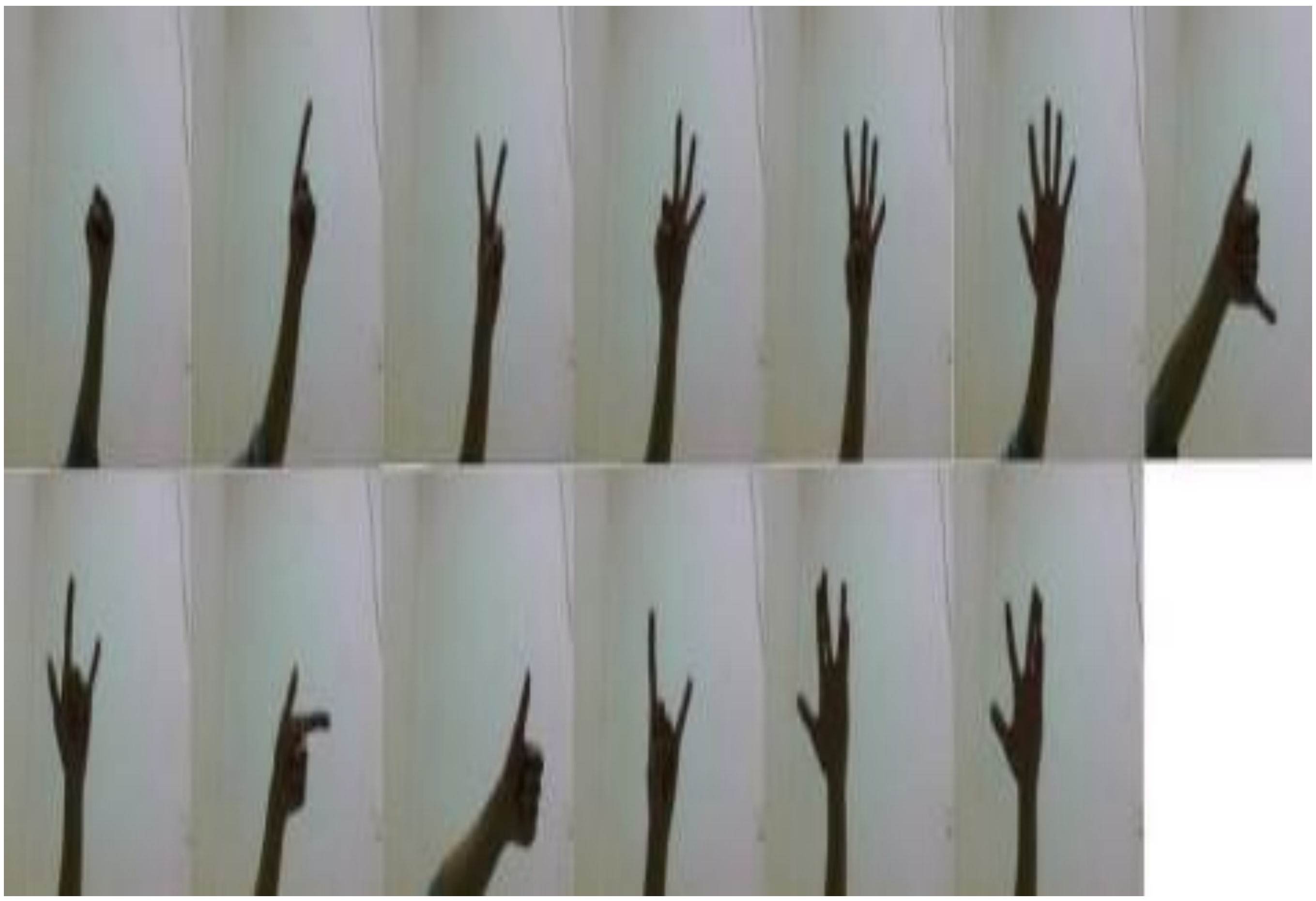


Fig 4.1 Gestures used for training

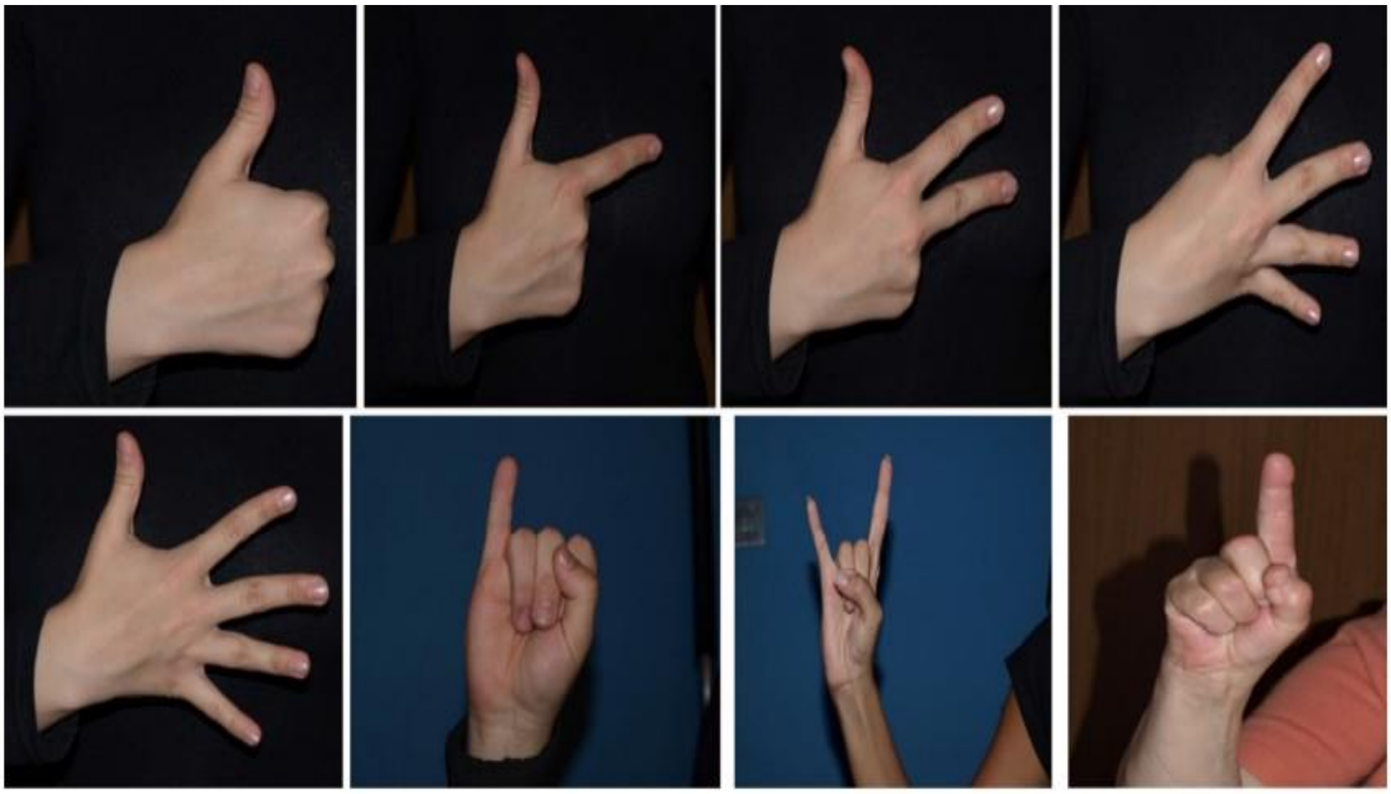


Fig 4.2 More training datasets

EXPERIMENTAL RESULTS

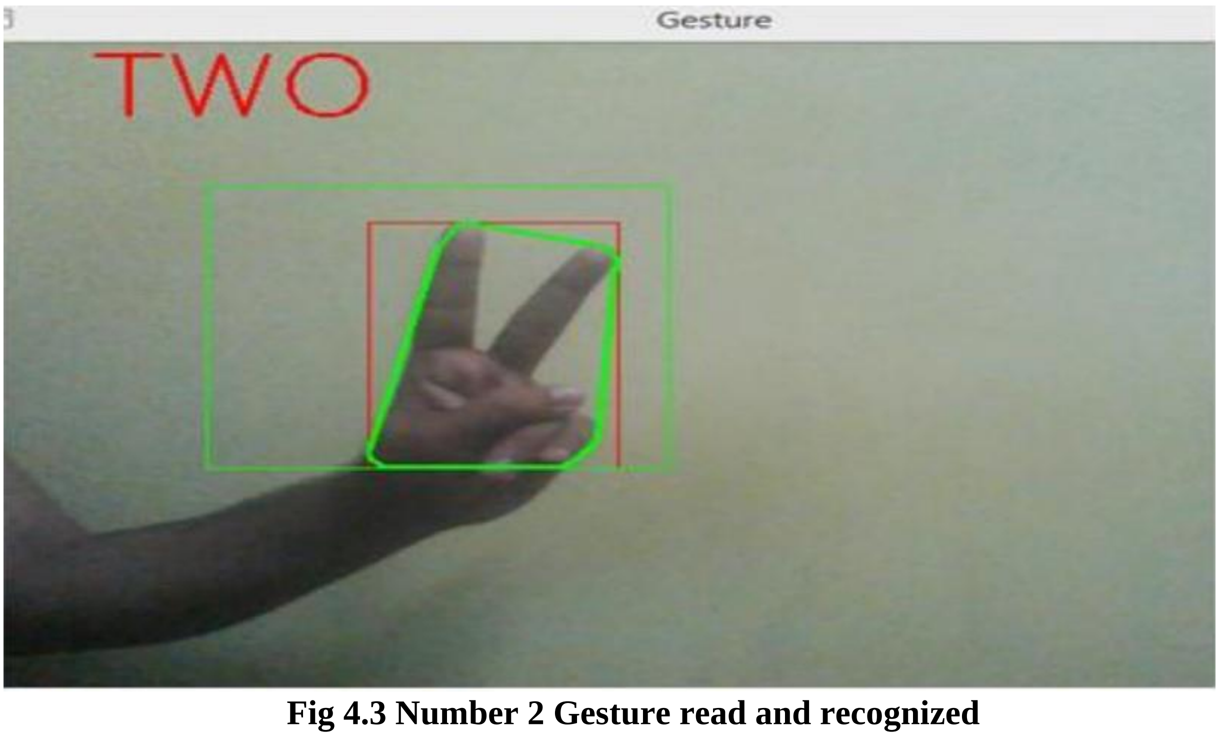


Fig 4.3 Number 2 Gesture read and recognized

The proposed model has been simulated to yield appropriate results in real-time applications. As a proof-of-concept evaluation of the source code is performed to use gestures to Switcha light source on and off and furthermore to control the audio input of another device.

The figures below show the verified output of a system being able to recognize a gesture posedand associate it with the controlling of a LED (a feasible light source)

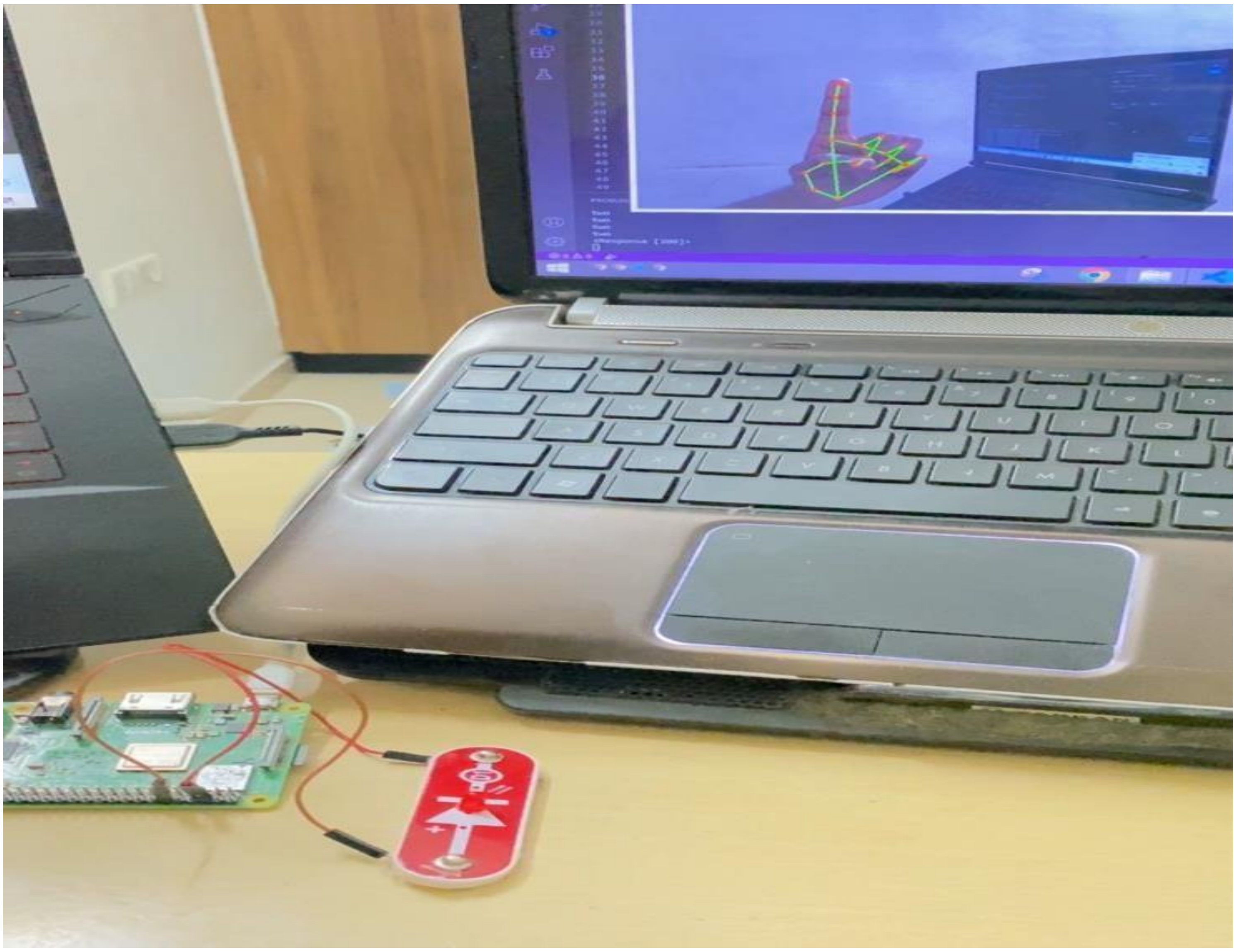


Fig 4.4- showing the gesture ONE to indicate the ON function of light source

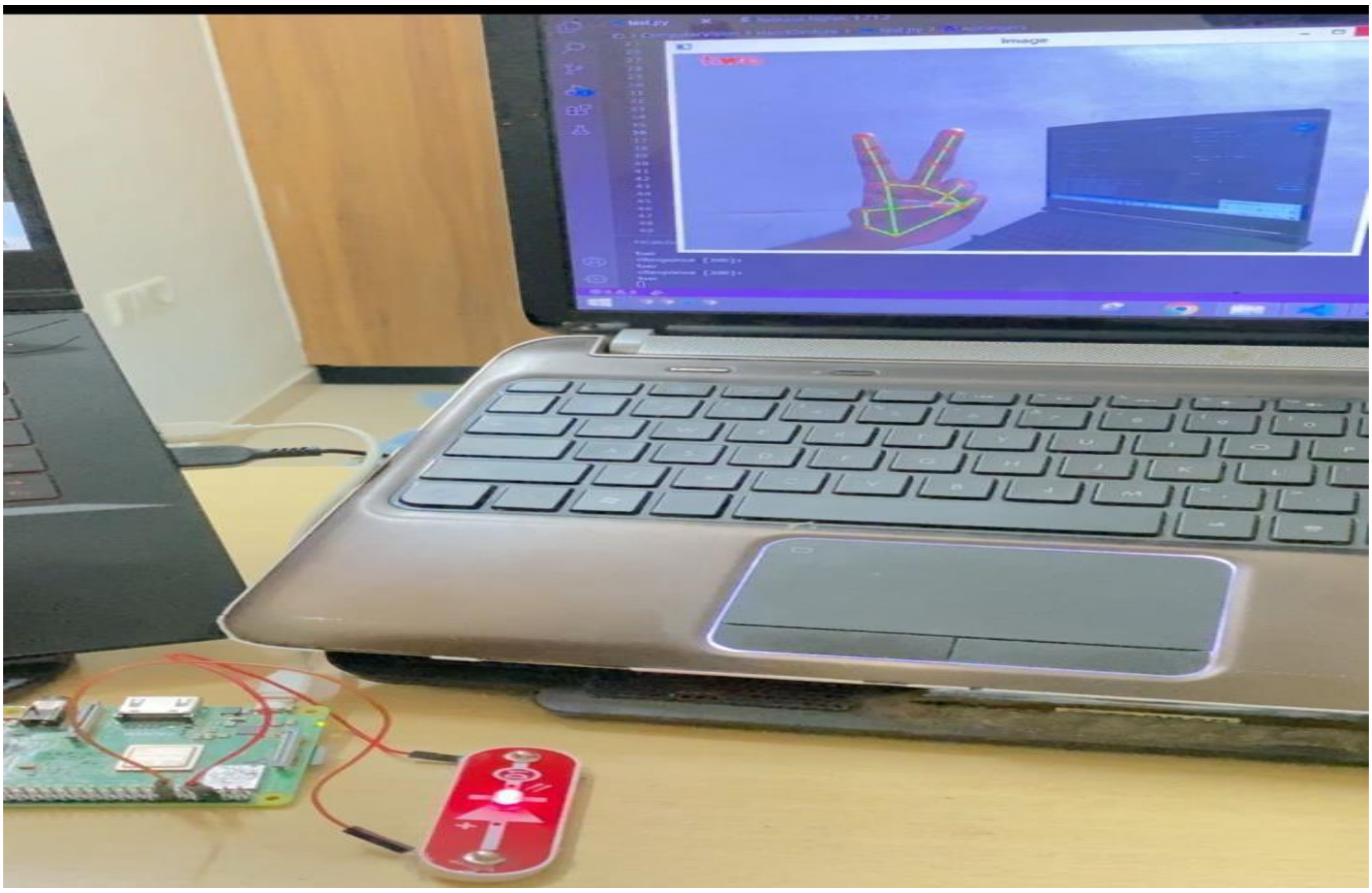


Fig 4.5- showing the gesture two to indicate the ON function of light

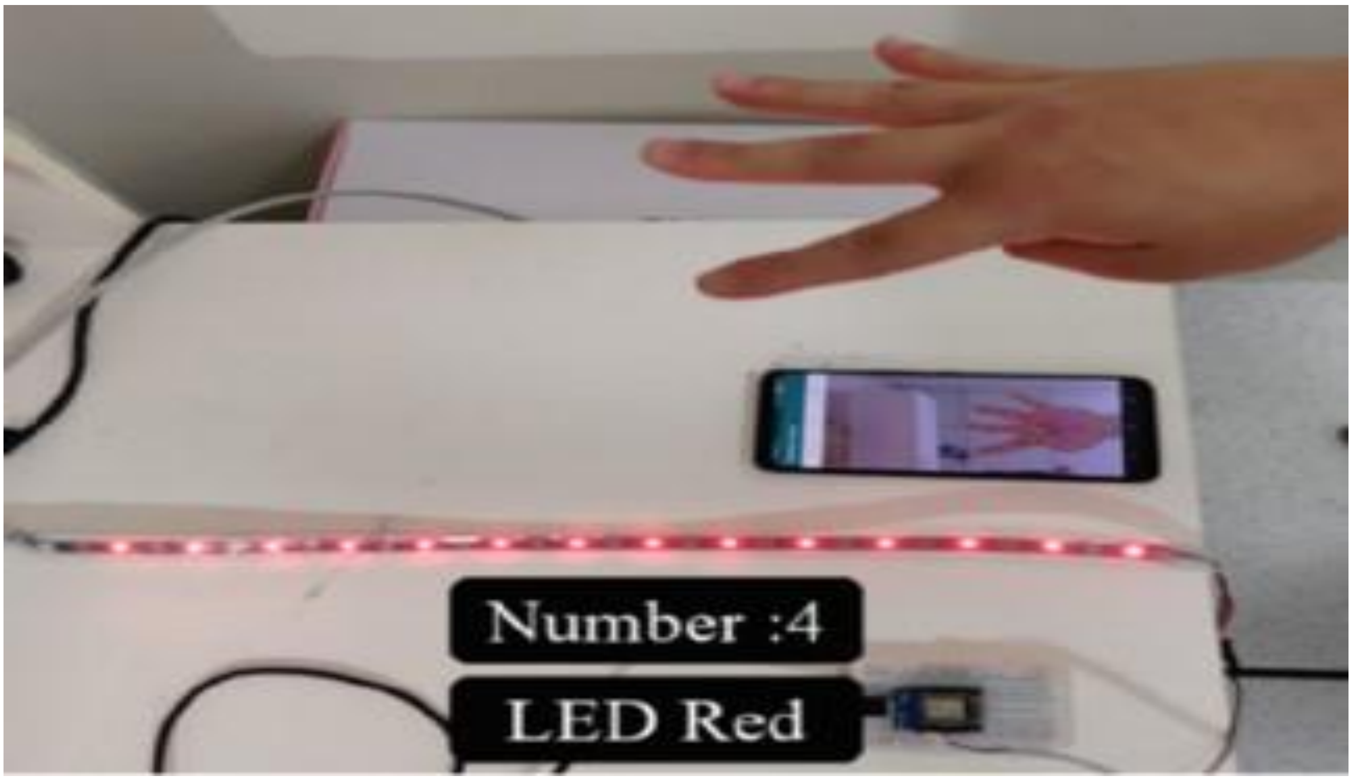


Fig 4.6 Showing gesture FOUR to turn the led strips to red

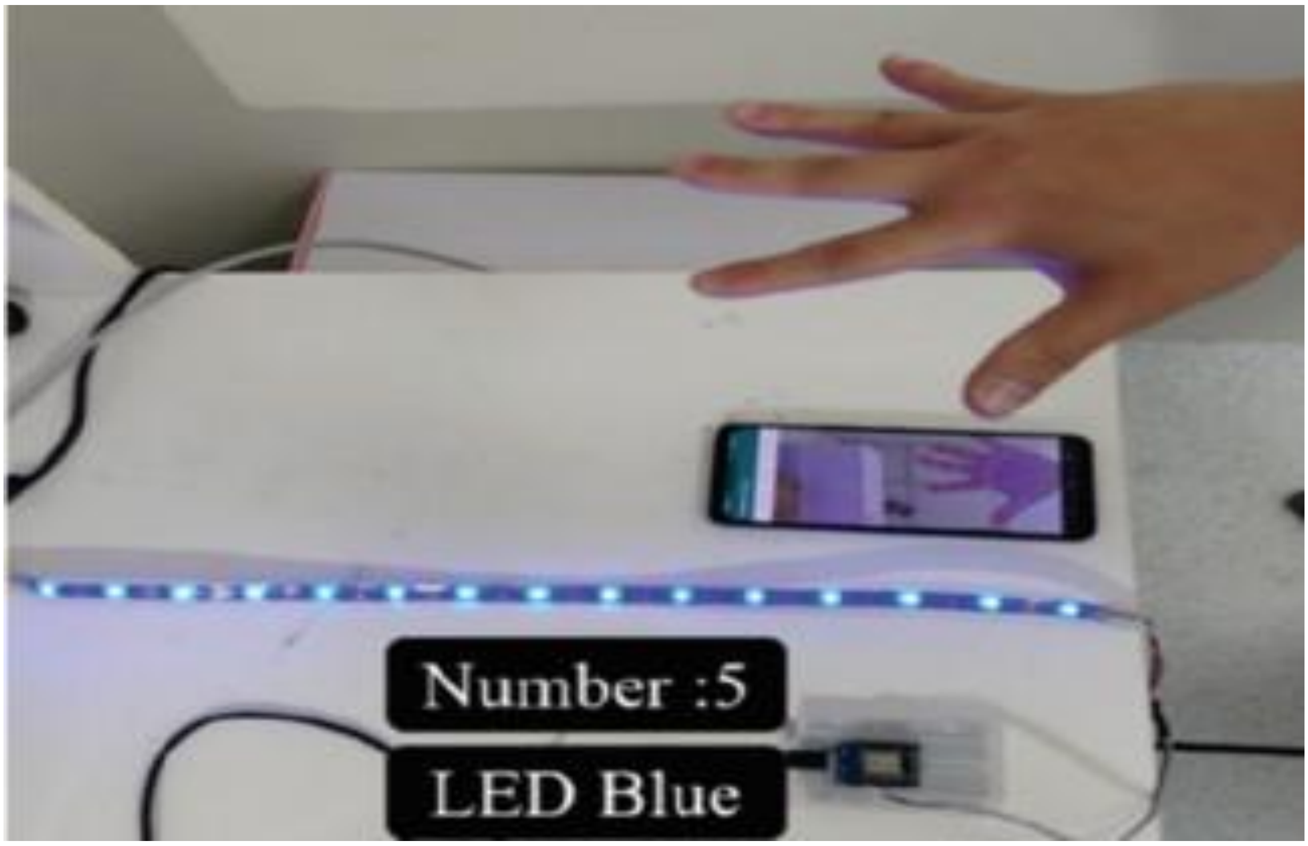


Fig 4.7 Showing gesture FIVE to turn led strips to blue color

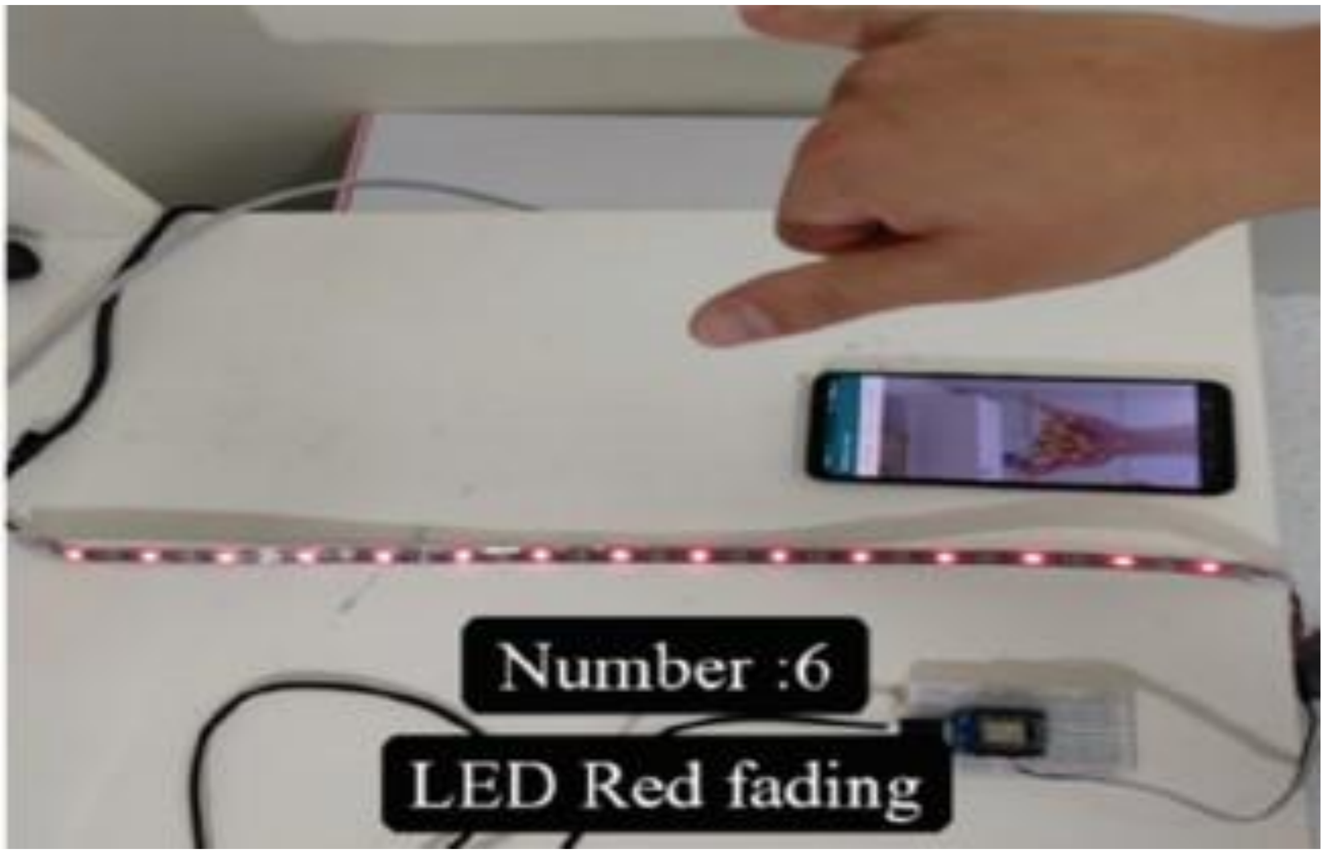


Fig 4.8 shows number SIX to fade the red LED

CHAPTER – 5
CONCLUSION AND DISCUSSIONS

The final chapter of the report consolidates all the findings of the executed module and putsforth the conclusion and discussions regarding the future advancements based on its final findings. Few demerits were reported on the previous chapter of the report for which theoretical solutions have been provided in the advancements part.

5.1 SUMMARY OF THE PRESENTED WORK

In this work, a model capable of recognizing and associating gestures with respective controlling functions of small-time appliances are consolidated. previously published papers and existing projects that utilizes gesture recognition using various technologies, along with their drawbacks has been briefly discussed about. The skeletal outline of the proposed model shows the data capturing , data manipulation and output representation. Further light is shed into the various hardware components and software used in the proposed model. Details about the pin diagram and details about the different components that is being used was thoroughly explained. Furthermore, details about the intricate workings of the proposed model are elucidated Gesture recognition is done so with the help of a convolutional neural network modelwhich was trained using python and embedded with raspberry pi and tensor flow. Due to shorttime, the module was executed for the controlling of an external light source(LED) and a soundsource (VLC MEDIA PLAYER). a deep insight into the various merits of the proposed modelwas spotted. It goes into detail about the various features and perks of this model. The same chapter also discusses about the demerits of the model and goes into details about the optimaldistance between the model and subject.

5.2 DISCUSSIONS

MERITS

The implemented model has shown to be a more effective and efficient than all the other models described in the literature survey from chapter 2. Some of the advantages observed that gives our model a front over the others are described below :

The model helps in unmonitored control of appliances for impaired people since it only uses simple gesture as a merit of controland it has shown to be highly cost effective as it uses the concept of IoT and needs only proper network connectivity for efficient managing of data. Due to the usage of local connectivity the model implements immediate and powerful interaction with high Speed and sufficient reliable for recognition system. With respect to capturing and recognizing of real time images the model shows good performance system with complex background and showcases fast and powerful results from the proposed algorithm

DEMERITS

Just like every other technologically accessible model, the implemented model along with its algorithm shows its own weaknesses. Most of the cons includes weak discoverability of the devices in the local network due to poor signal strength. Furthermore, Irrelevant object might overlap with the hand of the user that raises a gesture. Performance recognition algorithm decreases when the distance greater than 1.5 meters between the user and the camera is.

FUTURE ADVANCEMENTS

Using gesture recognition, good outcomes can be produced for various smart applications. Further scopes for this model include expanding this into a smart home control for visually andverbally impaired with a cloud server specific just for the home network and make the model as cost efficient as possible. More applications of gesture recognition can be embedded with this such as smart wheel chair and smart braille system which operate with respect to associationof a gesture made to its assorted function.

CHAPTER – 6

REFERENCES

[1] G. Jaramillo and M. E. Benalcázar, "Real-time hand gesture recognition with EMG using machine learning," 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), 2017, pp. 1-5, doi: 10.1109/ETCM.2017.8247487.

[2] Moin, A., Zhou, A., Rahimi, A. et al. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. Nat Electron **4**, 54–63 (2021). https://doi.org/10.1038/s41928-020-00510-8

[3] Qi, J., Jiang, G., Li, G. et al. Surface EMG hand gesture recognition system based on PCA and GRNN. Neural Comput & Applic **32**, 6343–6351 (2020). https://doi.org/10.1007/s00521-019-04142-8

[4] M. Al-Hammadi et al., "Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation," in IEEE Access, vol. 8, pp. 192527-192542, 2020, doi: 10.1109/ACCESS.2020.3032140.

[5] S. Skaria, A. Al-Hourani, M. Lech and R. J. Evans, "Hand-Gesture Recognition Using Two-Antenna Doppler Radar With Deep Convolutional Neural Networks," in IEEE Sensors Journal, vol. 19, no. 8, pp. 3041-3048, 15 April15, 2019, doi: 10.1109/JSEN.2019.2892073.

[6] Asif AR, Waris A, Gilani SO, Jamil M, Ashraf H, Shafique M, Niazi IK. Performance Evaluation of Convolutional Neural Network for Hand Gesture Recognition Using EMG. Sensors (Basel). 2020 Mar 15;20(6):1642. doi: 10.3390/s20061642. PMID: 32183473; PMCID: PMC7146563.

[7] Cheng-Ying Yang, Yi-Nan Lin, Sheng-Kuan Wang, Victor R.L. Shen, Yi-Chih Tung, Frank H.C. Shen & Chun-Hsiang Huang (2023) Smart Control of Home Appliances Using Hand Gesture Recognition in an IoT-Enabled System, Applied Artificial Intelligence, 37:1, 2176607, DOI: 10.1080/08839514.2023.2176607

[8] Fu J, Cao S, Cai L and Yang L (2021) Finger Gesture Recognition Using Sensing and Classification of Surface Electromyography Signals With High-Precision Wireless Surface Electromyography Sensors. Front. Comput. Neurosci. 15:770692. doi: 10.3389/fncom.2021.770692

[9] Lee, K.H.; Min, J.Y.; Byun, S. Electromyogram-Based Classification of Hand and Finger Gestures Using Artificial Neural Networks. Sensors 2022, 22, 225. https://doi.org/ 10.3390/s22010225

[10] GitHub android studio. [Online] Available: https://github.com/android. May 2022c. GitHub Google/mediapipe. [Online] Available: https://github.com/google/mediapipe. May 2022d. GitHub

[11] tfreytag/WoPeD. [Online] Available: https://github.com/tfreytag/WoPeD (Visited in 2022/05) GitHubThonny. [Online] Available: https://github.com/thonny/thonny. May 2022e.

[12] Gogineni, K., A. Chitreddy, A. Vattikuti, and N. Palaniappan. 2020. Gesture and speech recognizing helper bot. Applied Artificial Intelligence 34 (7):585–95. doi:10.1080/08839514. 2020.1740473.