A Mini Project Report on

Hand Sign Language Detection using IoT

B.E. - I.T Engineering

Submitted By

Atharv Joshi 19104036

Anjali Singh 20204006

AbhayPratap Singh 19104037

Under The Guidance Of

Prof. Sonal Jain

DEPARTMENT OF INFORMATION TECHNOLOGY

A.P SHAH INSTITUTE OF TECHNOLOGY

G.B. Road, Kasarvadavali, Thane (W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic Year: 2022-23

CERTIFICATE

This to certify that the Mini Project report on Intrusion Detection System using PIR sensor and ESP32 Camera has been submitted by **Atharv Joshi** (19104036), **Anjali Singh** (20204006) and **AbhayPratap Singh** (19104037) who are bonafide students of A.P Shah Institute of Technology, Thane, Mumbai, as a partial fulfillment of the requirement for the degree in **Information Technology**, during the academic year **2022-2023** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Prof. Sonal Jain	Dr. Kiran Deshpande
Subject In-charge	Head Department of Information Technology

TABLE OF CONTENTS

Chapter 1

Introduction

In this world, there is nothing like equality but all we can do is try and make this place a little bit live-able and bearable. One of the things in our control is how we help other people around us. For this, we should start small but starting is the key for change. Normal human beings do not have much difficulty interacting with each other and can express themselves easily through speech, gestures, body language, reading, writing, speech being widely used among them. Deaf and mute people are perfectly capable of almost every task that a normal human can perform. To bring them on the level of the layman and also give them an opportunity to shine is one of the first steps in making this world an equal place. One of the ways in which they communicate through sign-language. This is something that has to be taught to them from a very young age. This is also something that is very exclusive to them rather a normal person is usually not proficient or even familiar with sign language. At times like these, it is really difficult to find a translator and there is only so much that universal signs can do. Hand signs are an effective form of human-to-human communication that has a number of possible applications.

Communication through signs has consistently been a significant way for communication among hearing and speech impaired humans, generally called deaf and dumb. It is the only mode of communicating for such individuals to pass on their messages to other human beings, and hence other humans need to comprehend their language. In this project, sign language detection or recognition web framework is proposed with the help of image processing. This application would help in recognizing Sign Language. The dataset used is the Indian Sign Language dataset. This application could be used in schools or any place, which would make the communication process easier between the impaired and non-impaired people. The proposed method can be used for the ease of recognition of sign language. The method used is Deep Learning for image recognition and the data is trained using Convolution Neural Network. Using this method, we would recognize the gesture and predict which sign is shown on the system

Chapter 2

Review of Literature

Sr. No	Research Paper	Finding from Paper
1.	Indian Sign Language recognition system using SURF with SVM and CNN.	We learned that sign language can be recognized using a webcam and different types of algorithms. For e.g., SVM - Support Vector Machine, CNN- Convolutional Neural Network and their accuracy while detecting objects on the webcam.
2.	ESP32 CAM WEB Server and Getting Started Guide	We learned how to use the ESP 32 cam and how to connect it to your PC to program it. This article used FTDI, this USB to Serial module. This Module is based around the FTDI FT232 chip. you can use other serial converters as well; wiring will remain the same for them as we only have to connect TX and RX with the ESP32.
3.	Sign Language Recognition Based on Machine Learning	This work showed us how machine learning can be applied in the process of Sign Language Recognition. This study proposed an online use of gesture-based communication acknowledgment utilizing American Sign Language (ASL). The proposed online application will assist with eliminating the correspondence hole by being an instructional exercise to learn and figure out the gesture-based communication. In this, we have utilized a dataset of 57,000 pictures for both testing and preparing.

		Calculations, for example, Na¨ive Bayes calculation, Support Vector Machine (SVM), k-Nearest Neighbors (KNN) and Convolutional Neural Network (CNN) are utilized for preparing the dataset and acquiring results.
4.	Real-Time Hand Detection using Convolutional Neural Networks for Costa Rican Sign Language Recognition	Used a video dataset to train their model and use it to detect the sign language in real-time. From this work we learned how to implement our hand sign language detection in real-time.

Chapter 3

Problem Statement

People affected by speech impairment rely only on sign language, which makes it more difficult for them to communicate with the remainder of the majority. This implies a requirement for sign language recognition system which can recognize and convert sign language into spoken or written language and vice versa. Such identifiers, however, are limited, costly, and cumbersome to use [1]. The problem statement goes as follows:" User wants to use sign language to converse with a deaf and mute person but the user is not familiar with the sign language." Proposed Solution: The solution we propose is making a real time hand sign language detection module using eps32 cam and a detection model hosted on the cloud which would be easier and cheaper to install and use

a. Motivation

Deaf and mute people cannot communicate in a way that physically abled people can. There are limited ways of communication for the deaf and mute, especially when they are young and/or are not capable of writing things but know sign language. We need a way for them to communicate and try to help them be more sociable. We want to incorporate everyone into our society but it is not possible when there are some physical limitations. The problem comes when the differently abled people have no way to communicate, especially when they are not literate in written language but know sign language. We plan on making a real-time hand sign detection system which will detect and display the Indian sign language on the screen as it is detected.

b. Objectives

These are the objectives that we hope to satisfy:

1. To detect hand sign language and display the letter on the screen. This will help a person who does not know how to sign to see what the deaf and mute person is trying to convey through the sign language.
2. To make use of leading machine learning algorithms to detect and recognize hand signs. This will be done through the use of CNN for building, training and testing the model which is faster and more efficient.
3. To teach young children how to sign and evaluate by checking their form. Children who want to learn how to sign can check their form against this model to check how they perform.
4. To use this anytime, anywhere. This will be done with the help of cloud hosting the model. Thus, it will give us the ability to use the same model for multiple devices/modules.

Chapter 4

System Architecture

a. State Diagram/Workflow

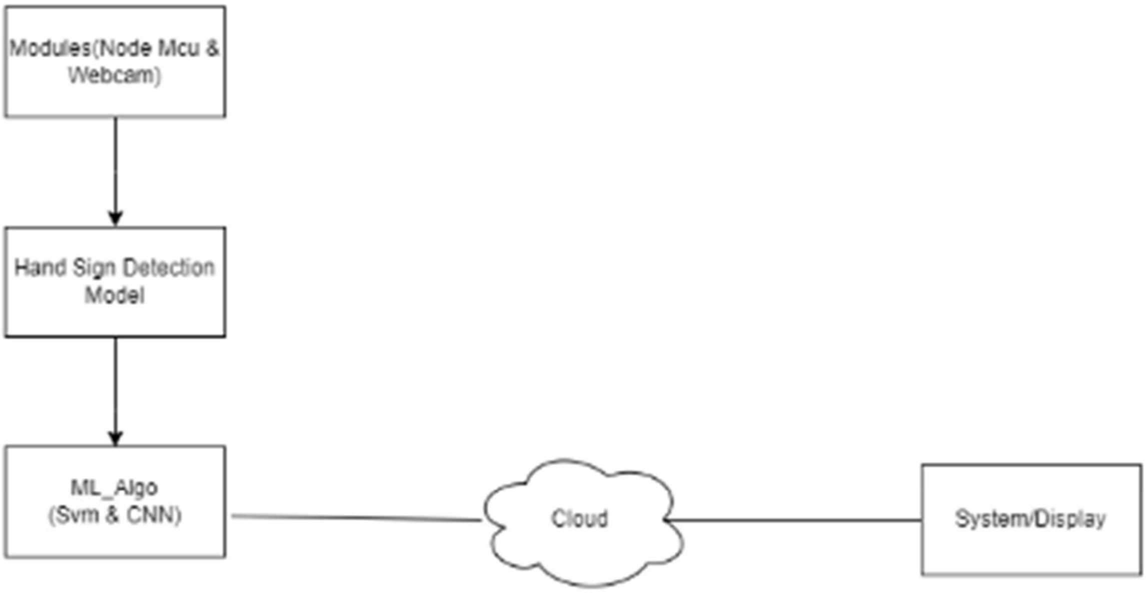

Figure 1: Proposed Working of the Project

b. Circuit Diagram

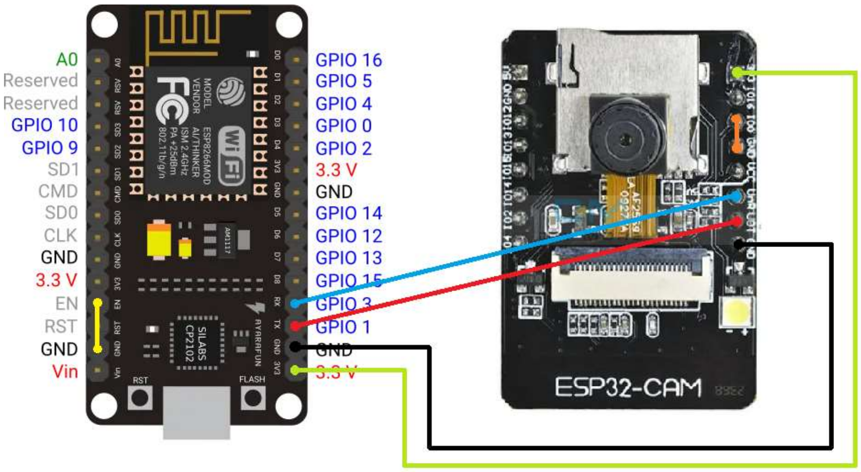

Figure 2 :- Circuit Diagram of Camera connection

Chapter 5

Project Timeline

Sr No.	Work to be Performed	Time Duration
1.	Literature Review and Documentation	1st August to 20th August
2.	Completion of Hardware	21st August to 10th September
3.	Working on model	11th September to 30th September
4.	Integration of IOT model (Arduino IDE)	1st October to 20th October

Chapter 6

Implementation

Code for Camera Web Server:

```
      #include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>

const char* WIFI_SSID = "Atharv's Galaxy M51";
const char* WIFI_PASS = "ok i wont";

WebServer server(80);


static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);
void serveJpg()
{
  auto frame = esp32cam::capture();
  if (frame == nullptr) {
    Serial.println("CAPTURE FAIL");
    server.send(503, "", "");
    return;
  }
  Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame-
>getHeight(),
                static_cast<int>(frame->size()));

  server.setContentLength(frame->size());
  server.send(200, "image/jpeg");
  WiFiClient client = server.client();
  frame->writeTo(client);
}

void handleJpgLo()
{
  if (!esp32cam::Camera.changeResolution(loRes)) {
    Serial.println("SET-LO-RES FAIL");
  }
  serveJpg();
}

void handleJpgHi()
{
  if (!esp32cam::Camera.changeResolution(hiRes)) {
    Serial.println("SET-HI-RES FAIL");
  }
  serveJpg();
}

void handleJpgMid()
{
  if (!esp32cam::Camera.changeResolution(midRes)) {
    Serial.println("SET-MID-RES FAIL");
  }
  serveJpg();
}


void  setup(){
  Serial.begin(115200);
  Serial.println();
  {
    using namespace esp32cam;
    Config cfg;
    cfg.setPins(pins::AiThinker);
    cfg.setResolution(hiRes);
    cfg.setBufferCount(2);
    cfg.setJpeg(80);

    bool ok = Camera.begin(cfg);
    Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
  }
  WiFi.persistent(false);
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  Serial.print("http://");
  Serial.println(WiFi.localIP());
  Serial.println("  /cam-lo.jpg");
  Serial.println("  /cam-hi.jpg");
  Serial.println("  /cam-mid.jpg");

  server.on("/cam-lo.jpg", handleJpgLo);
  server.on("/cam-hi.jpg", handleJpgHi);
  server.on("/cam-mid.jpg", handleJpgMid);

  server.begin();
}

void loop()
{
  server.handleClient();
}
```

Backend is done with Python.

a. Hardware Requirements: -

- Node MCU (ESP 8266).
- BreadBoard.
- Camera-ESP 32.
- Jumper Wires.

b. Software Requirements: -

Arduino IDE.

Python

OpenCV

c. Cost Estimation: -

Components	**Quantity**	**Price**
Node MCU	1	300/-
Breadboard	1	60/-
Camera	1	500/-
Jumper Wires	20	40/-
Total	-	900/-

d. Principle and working of project: -

1. ESP 32 Cam is used as the main sensor of the project. The camera picks up the image and a web server is programmed into it by using Arduino IDE and NodeMCU.
2. With the help of python program and a dataset from Kaggle, a model is created with the help of CNN which is used to train and test the data.
3. The camera picks up the live image and detects the hand sign that is shown on the camera. The input image is run through a mask that converts the image into a black and white in which the hand is white and the background is converted into black.
4. The image which is detected is recognized and the Letter or Number is displayed on the Screen.

Chapter 7

Conclusion

The ESP 32 camera is used to detect the image. It is programmed and then powered by NodeMCU (esp8266). A CNN model in the backend which is made in python is used to detect and recognize the image and the detected character is displayed on the screen.

Future Scope

The system can be used in hospitals and a portable model can be made. It can also be trained with different sign languages and even more phrases can be added to make it more efficient and usable.

References

[1] Katoch, S., Singh, V., Tiwary, U. S. (2022). Indian Sign Language recognition system using SURF with SVM and CNN. Array, 14, 100141. https://doi.org/10.1016/j.array.2022.100141

[2] https://www.hackster.io/ArnovS harmamakes/esp32−cam−web−server−and−getting−started − guide − f 1a04a

[3] Prof. Anjali M. Dalvi1, Shivam Sonawane2, Shruti Degaonkar3, Suraj Kulkarni4, Gauri Chavan5, “Sign Language Recognition Based on Machine Learning”, IJIRE-V3I03-137-143

[4] J. Zamora-Mora and M. Chac´on-Rivas,” Real-Time Hand Detection using Convolutional Neural Networks for Costa Rican Sign Language Recognition,” 2019 International Conference on Inclusive Technologies and Education (CONTIE), 2019, pp. 180-1806, doi: 10.1109/CONTIE49246.2019.00042.

[5] The dataset used to train the model: https://www.kaggle.com/datasets/vaishnaviasonawane/indian-sign-language-datase