HAND GESTURE TRANSLATOR

OUR MAIN IDEA

Communication is an essential process in our lives, and it's important to understand the needs of other people and help them.

Since we have the ability and the technology, we decided to make this project to bridge the gab between hearing and non hearing communities.

PHASES

01

AI model and computer vision

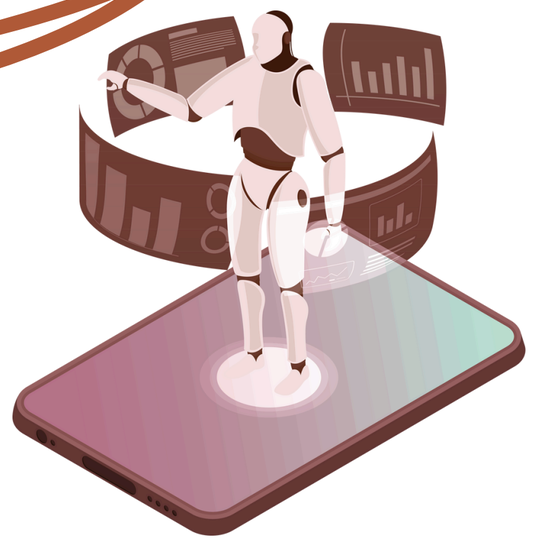
02

IOT part and integration with AI

03

MIT application

AI MODEL AND COMPUTER VISION



phases of AI

Hand detection
(computer vision phase)

feature extraction

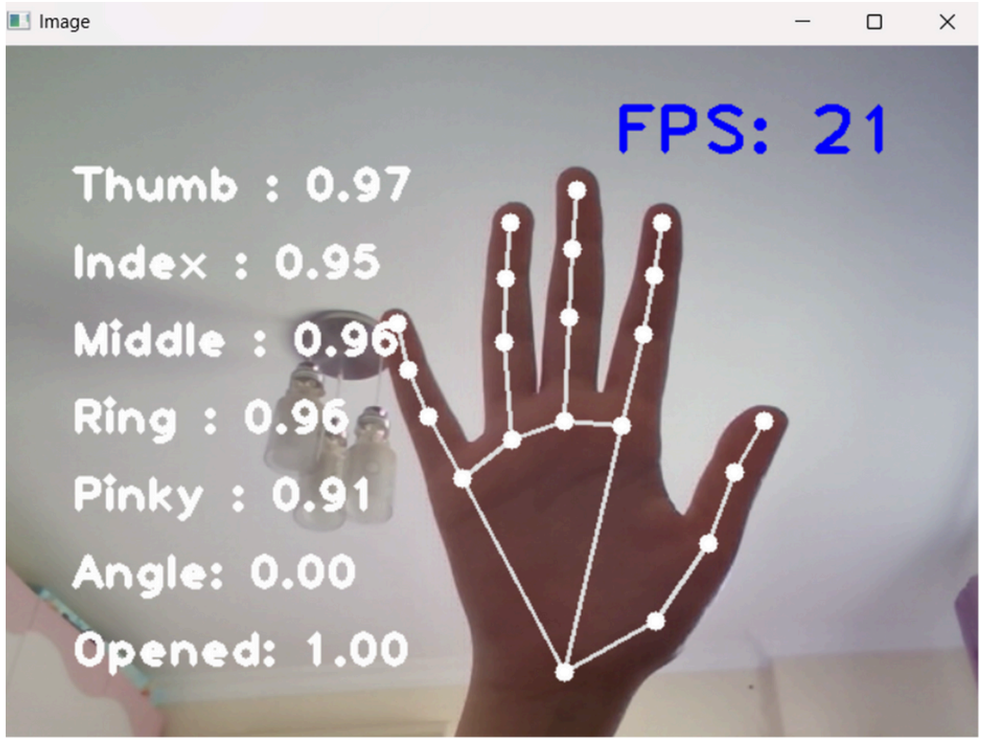
real time prediction

model training
(machine learnind
phase)

Hand Detection (Computer Vision Phase)

Description: We begin by capturing hand images using a webcam. Mediapipe’s hand detection model is used to locate and extract
21 key hand landmarks from image data for English , Arabic and Numbers

Goal: Isolate and track the hand for gesture
analysis.

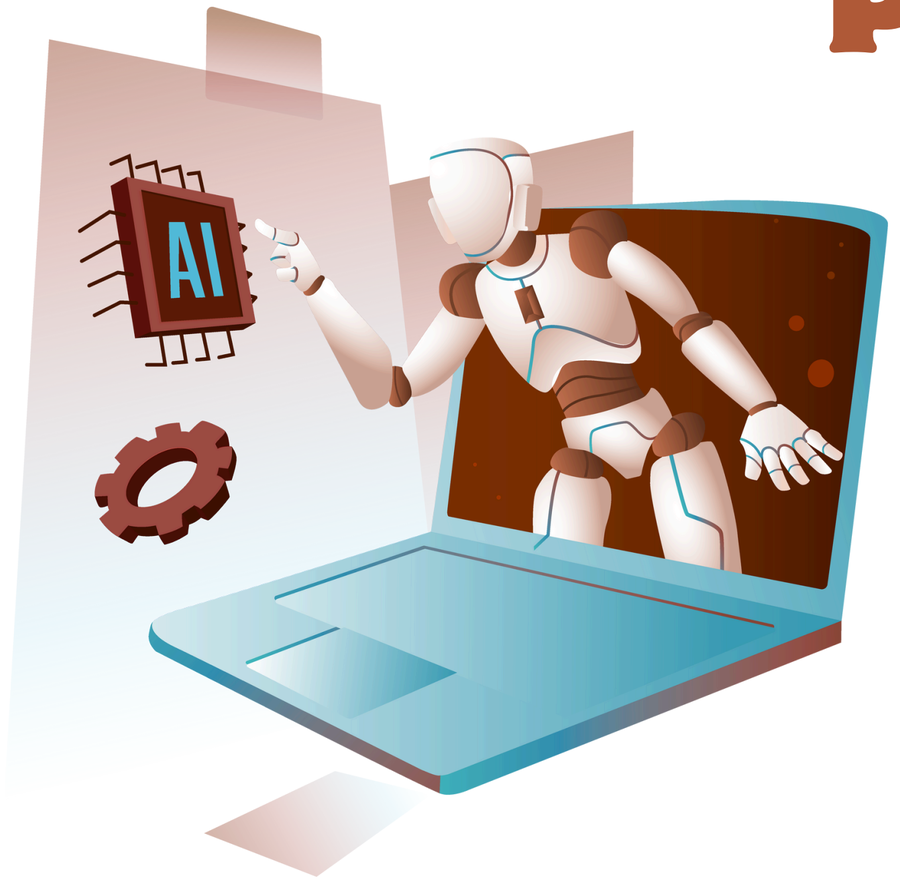


Feature Extraction

Description: For each detected hand, the 21 landmarks are converted into a 42-dimensional feature vector by normalizing x and y positions relative to the hand's position.

Goal: Standardize gesture data for accurate learning.

Model Training (Machine Learning Phase)



Description: The feature vectors are labeled and used to train a Random Forest Classifier. This model learns to distinguish between different letters in Arabic, English and Numbers based on hand shape and position.

Goal: Train the AI to recognize sign language accurately.

Real time prediction (Deployment phase)

Description:

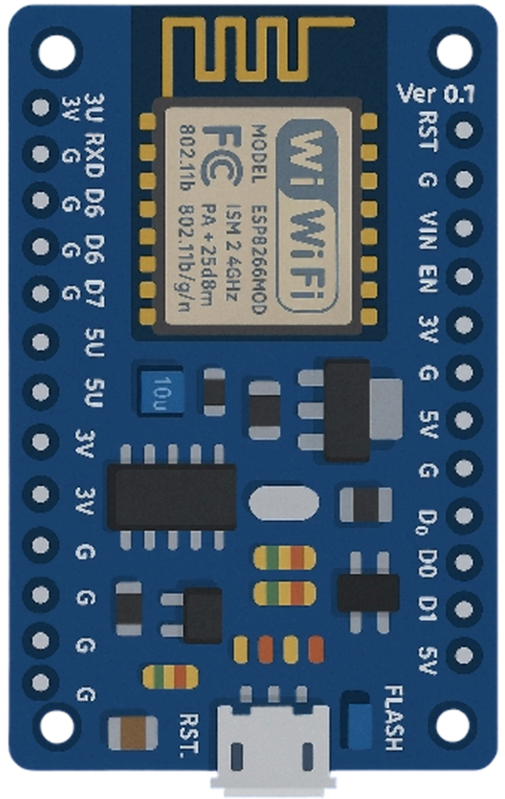
Goal: capturing hand images using a webcam. Mediapipe’s hand detection model to locate and extract 21 key hand landmarks from real video camera and process this landmarks then use the selected model to predict the letter (Arabic , English) or Numbers. and send it to the esp to display it on the LCD.

IOT and integration

ESP and integration with AI

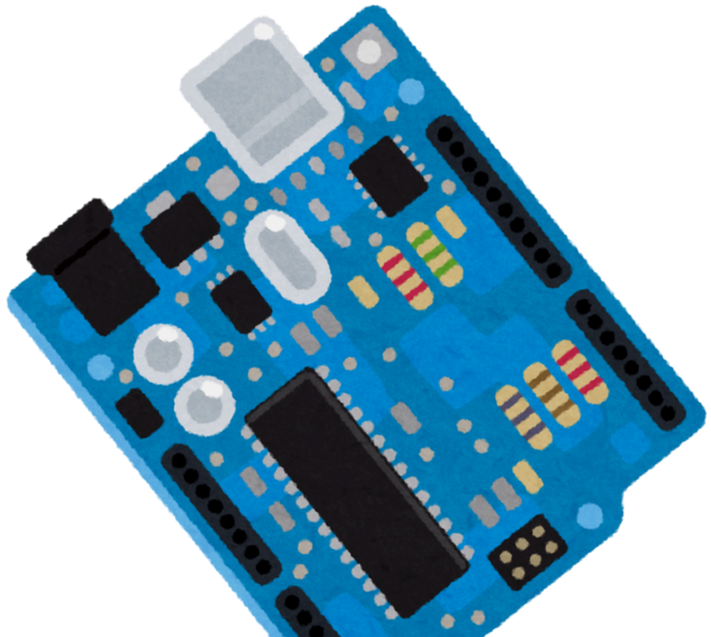
The integration between AI and IOT was implemented by using WebSocket protocol.

We used WebSocket to link the ESP to python wirelessly as It is a bidirectional protocol ,python acted as a WebSocket server and ESP acted as WebSocket client as the character translated is sent to the ESP .



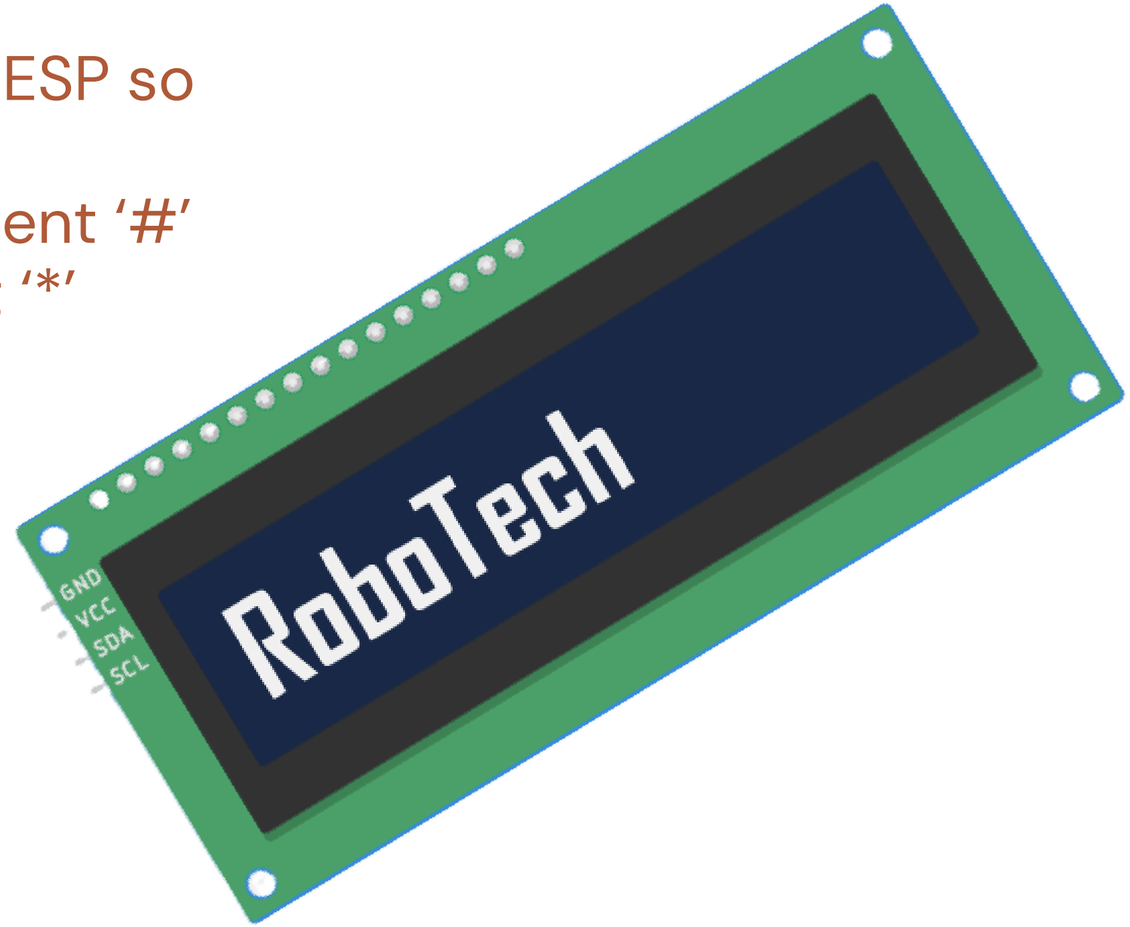
AURDUINO and ESP

The ESP and AURDUINO are linked by using UART protocol.we needed to connect them together as the components are on the AURDUINO , The character is sent by using Serial.print and is read by the aurduino by Serial.read



LCD

We wanted to show each character sent to the ESP so we used the LCD.
LCD shows also if a character is deleted if it is sent ‘#’ and shows that the word is reset if it is sent ‘*’

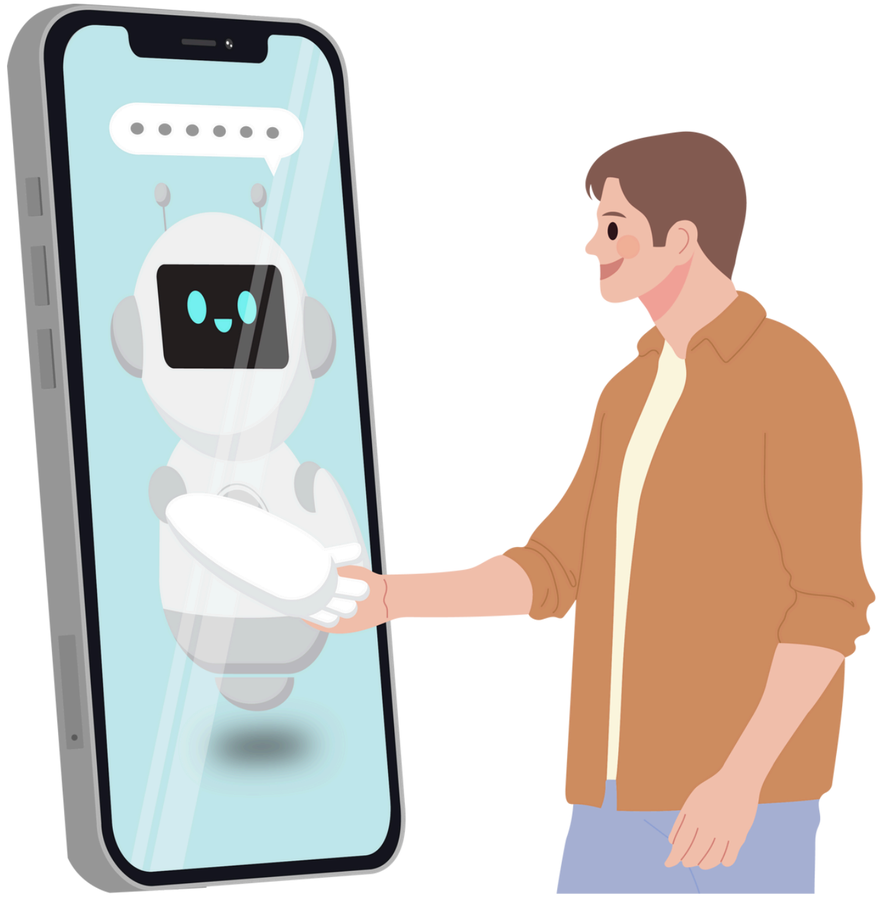


ESP and MIT

we needed a wireless communication to send the application the translated characters so we used WebSocket protocol again however in this time the ESP acted as a client and the application as a server

MIT APPLICATION

The application is our way to communicate with non hearing individuals.
The application displays the translated word and characters can be deleted from it .
It also displays pictures of sign language if you entered a word through it so it is a 2 way communication.



عِلْمٌ يُنْتَفَعُ بِهِ