



Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh

Faculty of Cyber Physical System

Dept. of Internet of Things and Robotics Engineering (IRE)

Course Title: Real life in IoT Lab

Course Code: IOT 4316

## **Lab Report-10**

**Submitted to-**

**Teacher name:** Nurjahan Nipa

**Designation:** Lecturer

**Department:** IRE

**Submitted by-**

Md. Shahriar Hossain Apu (1901036)

Date of Submission: 05-10-2023

**Title:** Real-time Hand Gesture Recognition using TensorFlow & OpenCV.

## **Introduction:**

Hand gesture recognition has numerous applications in human-computer interaction, sign language interpretation, and gaming. In this lab experiment, we aimed to design and implement a real-time hand gesture recognition system using deep learning techniques. We utilized TensorFlow for building the neural network model and OpenCV for capturing and processing real-time video streams.

The aim of this laboratory experiment is to develop a real-time hand gesture recognition system using TensorFlow and OpenCV. We collected and preprocessed a dataset of hand gesture images, built a convolutional neural network (CNN) model, and implemented real-time video capture and recognition. The system successfully recognized hand gestures in real-time video streams with a high degree of accuracy.

## **Apparatus:**

- 1.Arduino uno
- 2.LED.
- 3.Jumper wire.
4. Python – 3.x (we used Python 3.10 in this project).
5. OpenCV – 4.5
6. MediaPipe – 0.8.5
7. Tensorflow – 2.5.0
8. Numpy – 1.19.3

## **Methodology:**

### 1.Data Collection and Preprocessing:

- Collected a dataset of hand gesture images with corresponding labels.
- Preprocessed the dataset, including resizing, normalization, and data augmentation.

### 2.Model Architecture:

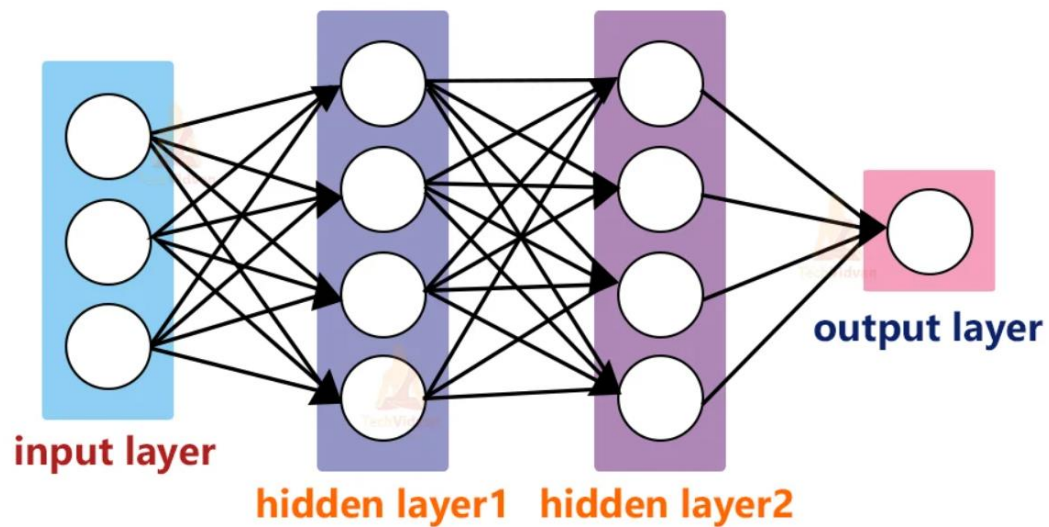
- Designed a CNN architecture for hand gesture recognition using TensorFlow's Keras API.
- Defined the layers, including convolutional layers, pooling layers, and fully connected layers.
- Compiled the model with appropriate loss and optimization functions.

### 3.Training:

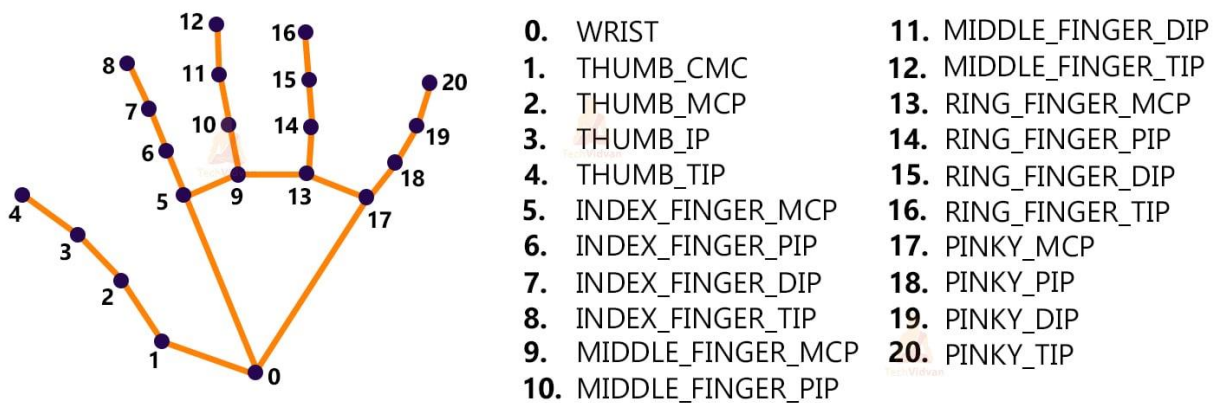
- Split the dataset into training, validation, and test sets.
- Trained the CNN model on the training set with data augmentation.
- Monitored model performance using validation data to prevent overfitting.

### 4.Real-time Gesture Recognition:

- Used OpenCV to capture real-time video frames from a webcam.
- Preprocessed each frame (resizing, normalization).
- Performed gesture recognition using the trained model on each frame.
- Displayed the recognized gesture on the video stream.

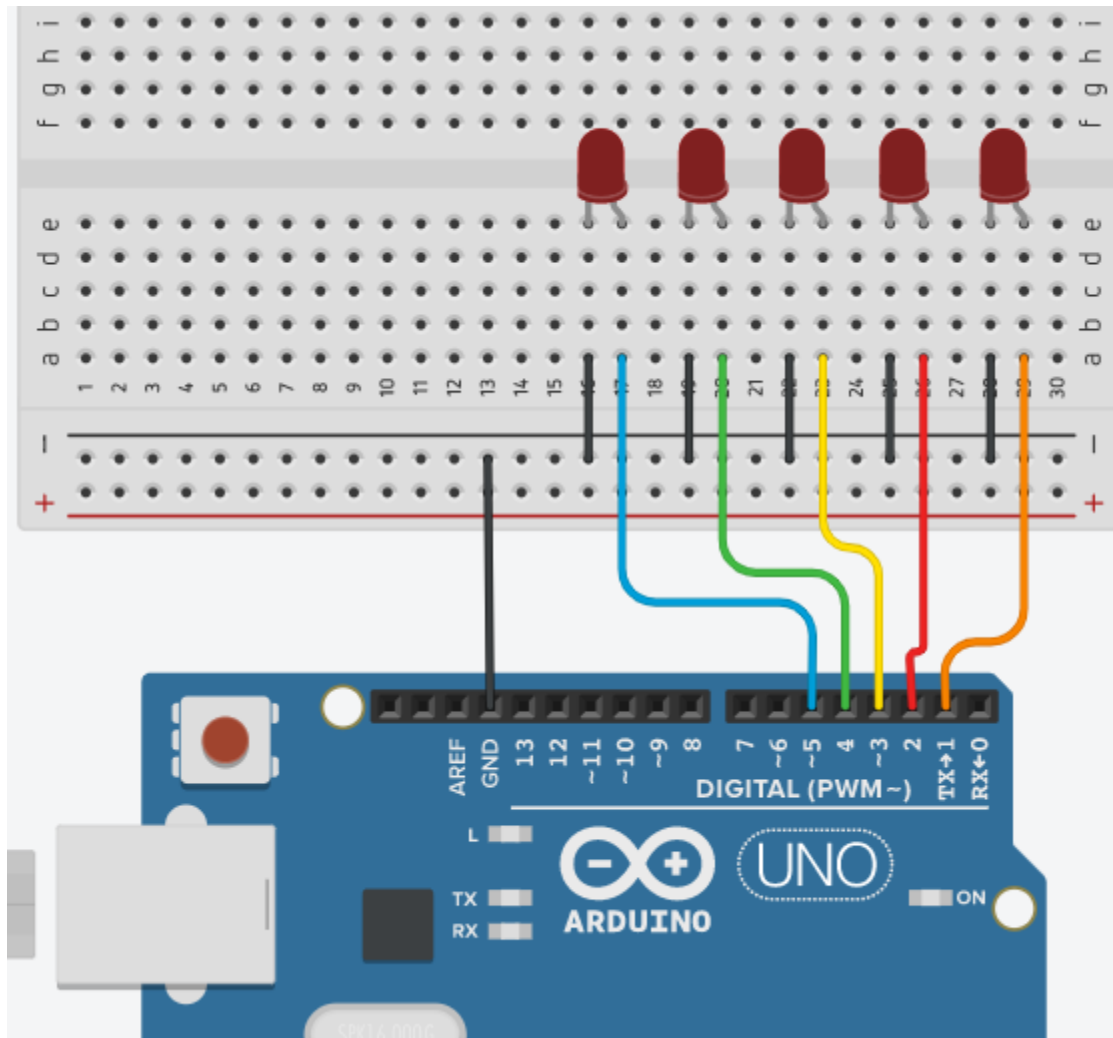


We'll first use MediaPipe to recognize the hand and the hand key points. MediaPipe returns a total of 21 key points for each detected hand.

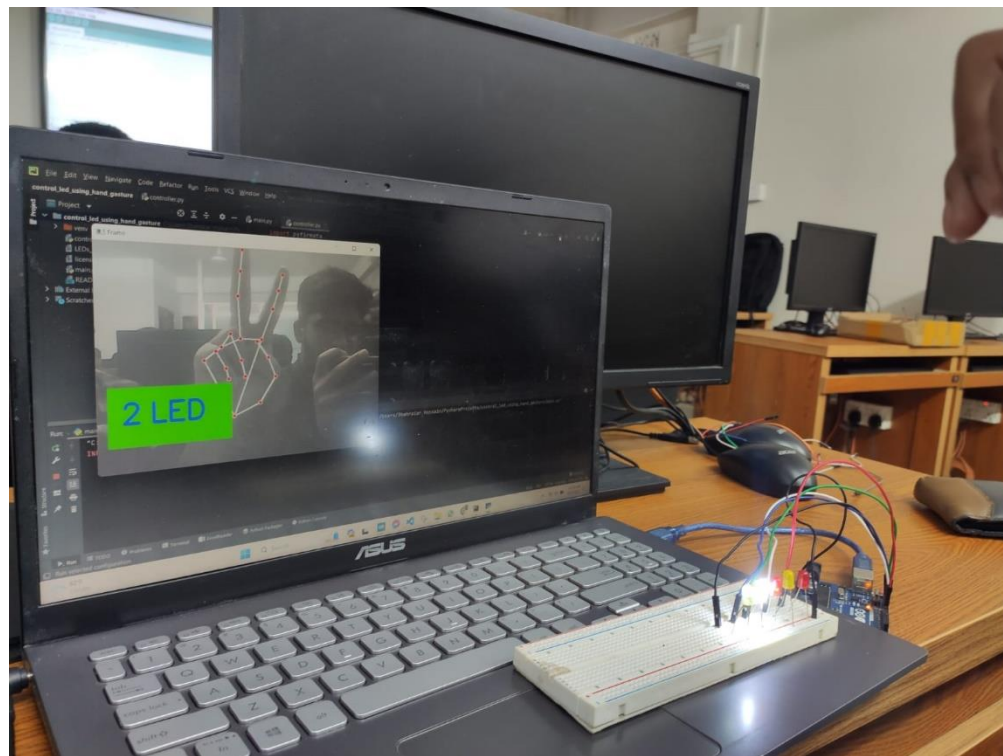
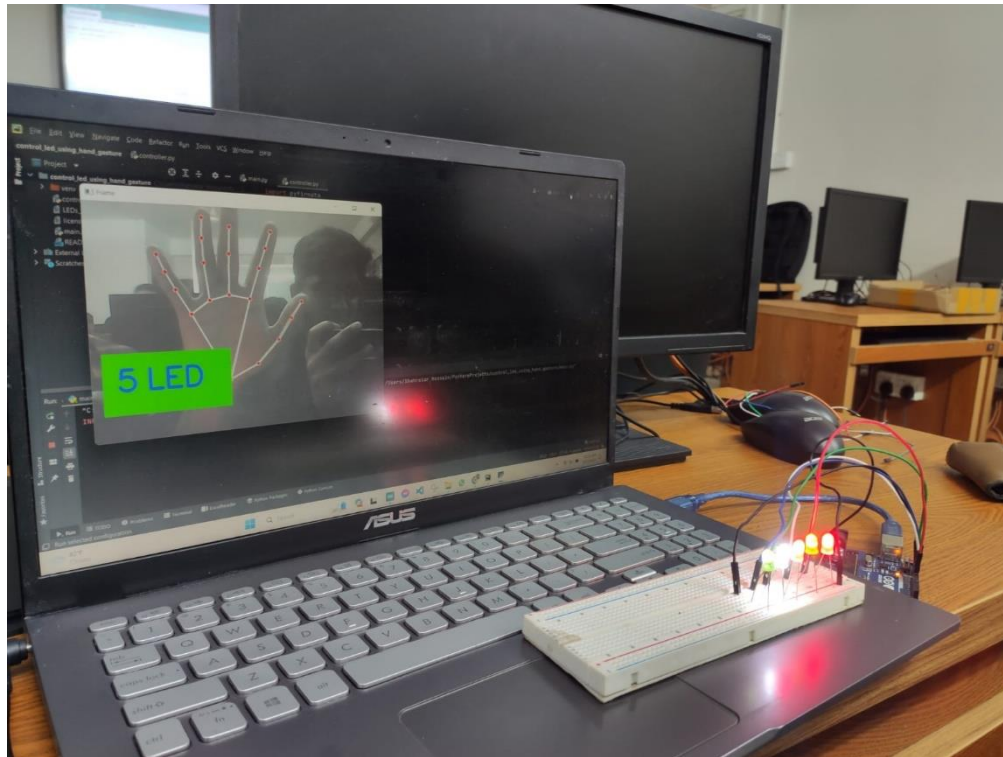


These key points will be fed into a pre-trained gesture recognizer network to recognize the hand pose.

## Circuit Diagram:



## Operational view:



**Fig:** Implementational view of Real-time Hand Gesture Recognition

## **Discussion:**

The high accuracy achieved on the test dataset indicates that the CNN model is effective in recognizing hand gestures. However, there are several factors that may influence the performance in real-world scenarios:

- Environmental conditions such as lighting and background may affect recognition accuracy.
- The dataset used for training may not cover all possible hand gesture variations.
- Fine-tuning the model architecture and hyperparameters could potentially improve accuracy.

Overall, the real-time hand gesture recognition system is a promising tool for applications requiring gesture-based interaction.

**GitHub Link:** [https://github.com/Shahriar-Hossain-Opu/LED\\_control\\_using\\_Hand\\_Gesture\\_Opencv\\_Python](https://github.com/Shahriar-Hossain-Opu/LED_control_using_Hand_Gesture_Opencv_Python)

## **Conclusion:**

In this lab experiment, we successfully developed a real-time hand gesture recognition system using TensorFlow and OpenCV. The system exhibited high accuracy on test data and demonstrated real-time recognition capabilities. Further improvements can be made by collecting a more diverse dataset and fine-tuning the model architecture. This technology has the potential for various applications in human-computer interaction and beyond.

***–End of the Report–***