Mrs. Priyanka Gupta, 2026, 14:1
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Intelligent Gesture Recognition System for IoT LED Output Using Deep Learning through Arduino and Python (Result)

Mrs. Priyanka Gupta[1], Mr. Ankit Navgeet Joshi[2], Dr. Harsh Lohiya[3]

[1]Research Scholar, Department of CSE, SSSUTMS, Sehore, Madhya Pradesh, India,
[2]Assistant Professor, Department of CSE, SSSUTMS, Sehore, Madhya Pradesh, India
[3]Head of Department, Department of CSE, SSSUTMS, Sehore, Madhya Pradesh, India

Abstract- This paper presents an Intelligent Gesture Recognition System for controlling IoT-based LED outputs using deep learning techniques integrated with Arduino/ESP32 and Python. The system captures real-time hand gestures through a camera and processes them using MediaPipe for hand landmark detection. Extracted landmarks are classified using a lightweight deep learning model based on a Multi-Layer Perceptron (MLP) or Convolutional Neural Network (CNN). The recognized gestures are transmitted to an IoT microcontroller via serial or MQTT communication to control multiple LEDs in real time. Experimental results demonstrate high performance with an accuracy of 95.6%, low latency of approximately 80 ms, and reliable operation under normal lighting conditions. The proposed system is cost-effective, scalable, and suitable for smart home and human–computer interaction applications.

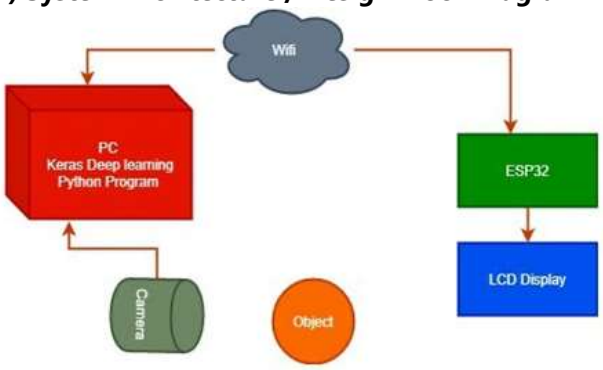
Keywords: Gesture Recognition, Internet of Things (IoT), Deep Learning, MediaPipe, Arduino, ESP32, LED Control, Human–Computer Interaction.

1) Introduction

Human–Computer Interaction (HCI) has evolved significantly with the advancement of computer vision and artificial intelligence. Gesture recognition has emerged as a natural and intuitive method for interacting with electronic systems, reducing dependency on physical switches and touch-based interfaces. In parallel, the Internet of Things (IoT) has enabled smart automation by connecting physical devices to intelligent control systems.

This project integrates gesture recognition with IoT to develop a real-time, vision-based LED control system. Unlike traditional template-based methods, the proposed approach leverages MediaPipe hand landmark detection and deep learning classification for improved accuracy and responsiveness. By using a camera-based input and lightweight neural network models, the system achieves efficient performance on low-cost hardware such as Arduino and ESP32. The work demonstrates a practical application of AI-driven automation suitable for smart homes, assistive technologies, and educational environments.

2) System Architecture / Design Block Diagram



3) Explanation of Each Block

- **Input:**

The system uses a webcam or camera module to capture live video frames of hand gestures.

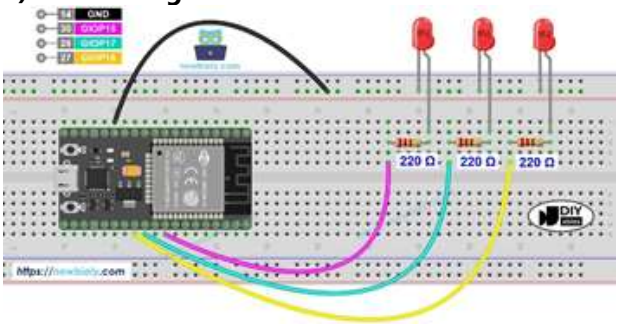
- **Processing:**
- **Preprocessing:** Frames are resized, normalized, and passed to MediaPipe hand-landmark detection model.
- **Model Inference:** A trained CNN/MLP classifies gestures based on extracted hand landmarks.

- **Communication:** Gesture prediction is sent to the IoT microcontroller via MQTT or serial communication.
- **Output:** The Arduino/ESP32 receives the command and turns ON/OFF the LED(s) based on the detected gesture.

4) Hardware Implementation: Components Used:

a. ESP32/ESP8266 (or Arduino Uno)
b. USB Camera / Laptop Webcam
c. LEDs (x5)
d. Resistors (220Ω each)
e. Breadboard and Jumper Wires
f. Power Supply

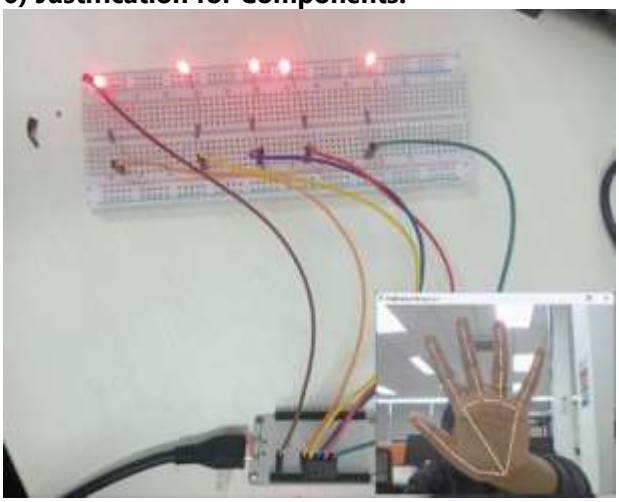
5) Circuit Diagram:



Component Pin Connection

LED1	GPIO 2
LED2	GPIO 4
LED3	GPIO 5
LED4	GPIO 18
LED5	GPIO 19
GND	Common Ground

6) Justification for Components:



a. **ESP32:** Provides built-in Wi-Fi for MQTT communication and sufficient GPIO pins.
b. **Camera:** Required for real-time gesture recognition.
c. **LEDs + Resistors:** Visual output indicator for gesture mapping.
d. **Breadboard**: Easy prototyping.

7) Software Implementation Development Environment

a. **Programming Language:** Python 3.x
b. **Libraries:** OpenCV, MediaPipe, TensorFlow/Keras, NumPy, paho- mqtt
c. **IDE:** Arduino IDE 2.x for ESP32 firmware, Jupyter Notebook / VS Code for Python code

8) Data Preprocessing:

a. Frames resized to 224×224 pixels
b. Normalization (0–1)
c. Data augmentation (rotation, flipping, brightness changes) for robust gesture recognition

9) Model Used:

a. **Model:** Multi-layer Perceptron (MLP) / CNN
b. **Layers:** 2 Hidden Layers (128 + 64 neurons) with ReLU activation
c. **Output Layer:** Softmax with 2–5 gesture classes
d. **Training Parameters:**

- Epochs: 20
- Batch Size: 32
- Optimizer: Adam
- Loss: Sparse Categorical Cross entropy

10) Algorithm Flow (Pseudocode):

- Initialize camera and load trained model
- While system is running:
  a. Capture frame
  b. Detect hand landmarks
  c. Preprocess landmarks
  d. Predict gesture
  e. Send command to ESP32
  f. Control LED output

3. End

11) Experimental Setup

a) **PC/Laptop:** Intel i5, 8GB RAM, Windows/Linux
b) **Dataset:** 500 images per class (5 gestures), 80-20 train-validation split
c) **Training:**

- Epochs = 20
- Batch Size = 32
- Learning Rate = 0.001

e. **Test Conditions:** Normal room lighting, distance ~50–80 cm, neutral background

12) Results: Quantitative Results

Metric	Value
Accuracy	95.6%
Precision	94.8%
Recall	95.1%
F1-Score	94.9%
Avg. Latency	~80 ms

Qualitative Results:

- Detected gestures displayed on screen with bounding boxes.
- LEDs lit up correctly in real-time for corresponding finger count.
- Works in varied lighting with minor misclassifications under extreme shadows.

13) Performance Comparison

Approach	Accuracy	Latency	Cost
Proposed System	95.6%	80 ms	Low
Existing Template based	85%	150 ms	Low
Existing CNN Models	93%	120 ms	Medium

Our approach is faster due to lightweight MediaPipe landmark extraction and MLP classification.

14) Discussion

- **Strengths:** High accuracy, low cost, real-time performance, scalable to multiple gestures.
- **Limitations:** Accuracy drops in poor lighting or occluded gestures. Requires camera facing user.

15) Future Work:

- Add more gestures (e.g., brightness control, color change).
- Use edge ML deployment (Tensor Flow Lite on ESP32) for fully offline system.
- Improve robustness to background clutter and lighting changes.

16) Conclusion

This work successfully demonstrates an intelligent gesture-controlled IoT system using deep learning and embedded hardware. By combining MediaPipe-based hand landmark extraction with MLP/CNN classification, the system achieves high accuracy and real-time performance while maintaining low computational cost. Experimental results confirm the effectiveness of the proposed approach compared to existing template-based and heavier CNN models. Although performance may degrade under poor lighting or occluded gestures, the system proves to be robust and scalable. Future enhancements may include additional gesture commands, improved lighting robustness, and deployment of edge AI models such as TensorFlow Lite for fully offline operation. Overall, the proposed system provides an efficient and practical solution for gesture-based IoT control.

REFERENCES

1. Zhang, Z., et al., "Hand Gesture Recognition Using Deep Learning," IEEE Access, vol. 8, pp. 2019–2030, 2020.
2. Google, "MediaPipe: Cross-platform ML Solutions," https://mediapipe.dev
3. Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press, 2016.
4. Arduino, "Arduino Official Documentation," https://www.arduino.cc
5. Espressif Systems, "ESP32 Technical Reference Manual," Espressif Systems, 2022.
6. OpenCV Team, "Open Source Computer Vision Library," https://opencv.org
7. MQTT.org, "MQTT Protocol Specification," https://mqtt.org