# PROJECT REPORT

## Hand Gesture Switch Control

---

**1. Introduction**

TinyML (Tiny Machine Learning) refers to running lightweight machine learning or intelligent algorithms on low-power edge devices such as microcontrollers and embedded systems.

In this project, we developed a **real-time hand gesture recognition system** using pure OpenCV techniques without heavy deep learning models. The system detects hand gestures through contour detection and convex hull analysis.

The system can be deployed on:

- Raspberry Pi
- Edge AI devices
- Low-power embedded systems

---

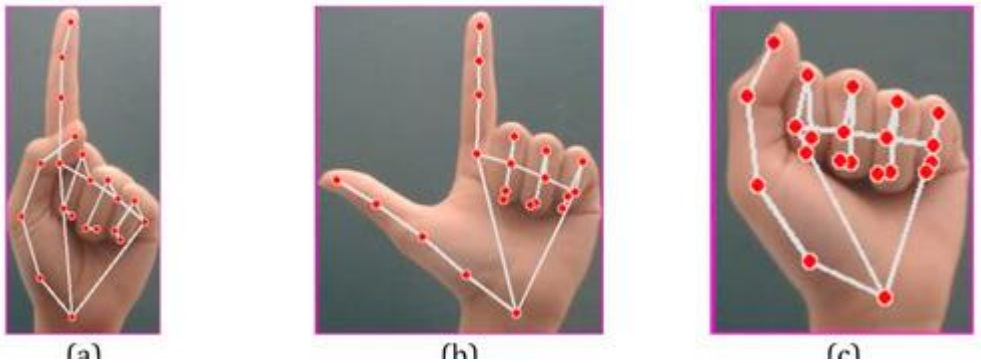
**2. Problem Statement**

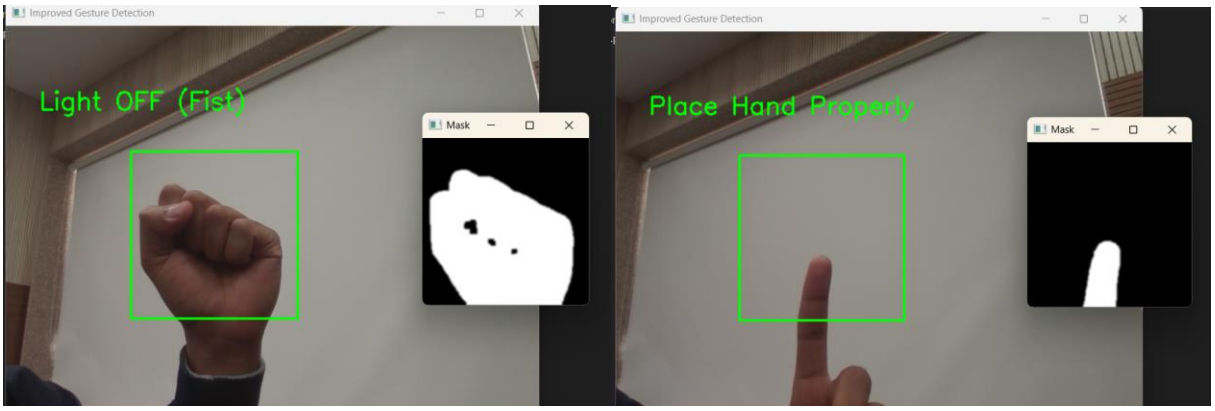
To design a lightweight, real-time gesture recognition system that:

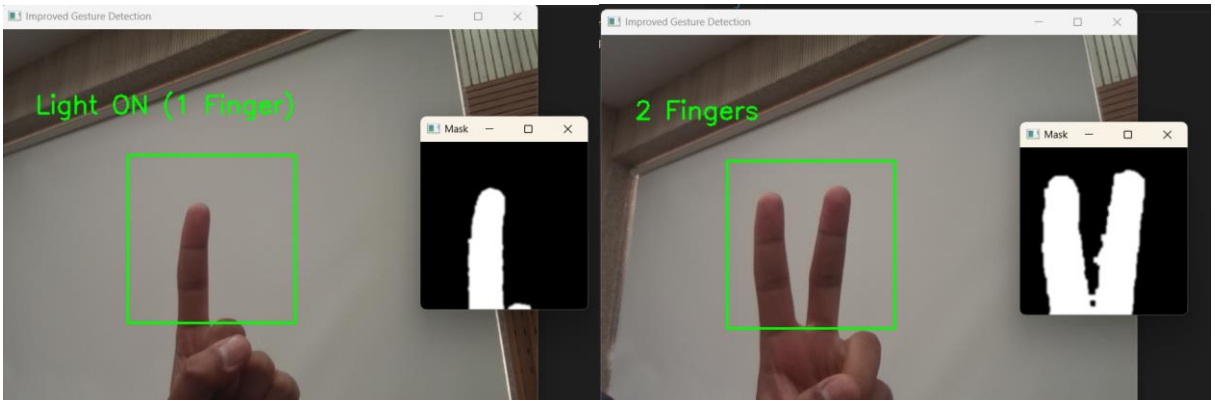
- Works without cloud processing
- Requires minimal computational resources
- Can be deployed on edge devices
- Controls virtual outputs (Light ON / OFF)

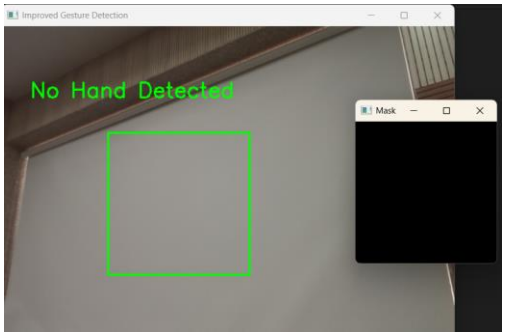
---

## 3. System Architecture



**4. System Flow:**

1. Capture live video
2. Define Region of Interest (ROI)
3. Convert to YCrCb color space
4. Skin detection
5. Noise filtering
6. Contour detection
7. Convex hull & defect analysis
8. Finger counting
9. Output decision

---

**5. Methodology**

**🔹 Step 1: ROI Selection**

A smaller green box is defined to reduce background noise and improve detection accuracy.

**🔹 Step 2: Skin Detection**

The image is converted to **YCrCb color space**, which provides better skin segmentation than HSV.

**🔹 Step 3: Noise Removal**

Morphological operations:

- Closing
- Opening
- Gaussian Blur

are applied to remove small noise particles.

**🔹 Step 4: Contour Detection**

The largest contour inside ROI is assumed to be the hand.

**🔹 Step 5: Convex Hull & Defects**

Convex hull is calculated.
Convexity defects between fingers are used to count finger gaps.

### 🔷 Step 6: Special Case Handling

Single finger detection is handled using bounding rectangle height/width ratio.

---

## 6. Algorithm

1. Capture frame
2. Extract ROI
3. Convert ROI to YCrCb
4. Apply skin threshold
5. Apply morphology operations
6. Detect contours
7. If area > threshold
    - Compute convex hull
    - Compute defects
    - Count fingers
8. Display result

---

## 7. Results

**Observed Outputs:**

| Gesture | Output |
|---|---|
| ✊ Fist | Light OFF |
| ☝ One Finger | Light ON |
| ✌ Two Fingers | 2 Fingers |
| ✋ Open Hand | 5 Fingers |

System works accurately under proper lighting and simple background conditions.

---

**8. Advantages**

- Lightweight (No Deep Learning required)
- Real-time performance
- Edge deployable
- Low computational cost
- Python 3.12 compatible

---

**9. Limitations**

- Sensitive to lighting conditions
- Skin tone variation may affect detection
- Background should be plain

---

**10. Applications**

- Smart home control
- Touchless interfaces
- Robotics control
- Assistive technology
- Industrial automation

---

11. **Future Scope**

- Convert to TensorFlow Lite model
- Deploy on ESP32-S3

- Add gesture classification using ML
- Add adaptive skin calibration

---