# Gesture Control and Emotion Recognition System Report

**Rewaa Alaa 320210297**

**Sama Elqasaby 320210313**

**Nouran Galal 320210264**

## 1 Introduction

This report provides a comprehensive analysis of two computer vision-based systems: a **Gesture Control System** for hands-free slide navigation and an **Emotion Recognition System** for real-time facial emotion detection. Both systems leverage advanced computer vision techniques and are designed for intuitive human-computer interaction. The gesture control system enables presenters to control slides using hand gestures, while the emotion recognition system detects emotions from facial expressions, offering potential for audience engagement analysis. This report details the technical implementation, functionality, performance, and recommendations for both systems, with a focus on their integration potential.

## 2 Gesture Control System

### 2.1 Overview

The gesture control system allows presenters to navigate presentation slides using hand gestures captured via a webcam. It uses MediaPipe Hands for hand landmark detection and PyAutoGUI for simulating keyboard inputs, enabling seamless interaction with slide software (e.g., PowerPoint, Google Slides).

### 2.2 Technical Implementation

- **Libraries and Dependencies**:
  - OpenCV: For webcam capture and image processing.
  - MediaPipe Hands: For real-time hand landmark detection (21 landmarks per hand).
  - PyAutoGUI: For simulating keyboard presses (right, left, F5, Esc).
  - NumPy: For numerical computations (e.g., distance calculations).
- **Gesture Definitions**:
  - *Next Slide (Thumbs Up)*: Thumb extended upward, other fingers folded.
  - *Previous Slide (Thumbs Down)*: Thumb extended downward, other fingers folded.
  - *Start Presentation (Open Palm)*: All fingers spread.

- *End Presentation (Closed Hand)*: All fingers folded (fist).

- **Hand Detection and Gesture Recognition**:
  - MediaPipe detects 21 hand landmarks with a minimum detection confidence of 0.7 and tracking confidence of 0.5.
  - Gesture detection is heuristic-based:
    - *Thumbs Up/Down*: Checks thumb tip y-coordinate relative to thumb IP joint (THUMB_THRESHOLD = 0.05) and ensures other fingers are folded (tip-to-wrist distance less than PIP-to-wrist).
    - *Open Palm*: All finger tips far from wrist (DISTANCE_THRESHOLD = 0.1).
    - *Closed Hand*: All finger tips close to wrist.
  - Normalized distances ensure robustness to hand size and distance from the camera.

- **Slide Control**:
  - Gestures map to keyboard inputs:
    - Next Slide: Right arrow.
    - Previous Slide: Left arrow.
    - Start Presentation: F5.
    - End Presentation: Esc.
  - A 1-second cooldown (COOLDOWN = 1.0) prevents multiple triggers.

- **User Interface**:
  - Displays hand landmarks on the video feed for visual feedback.
  - Presenter info (e.g., “Presenter: rewaa | Status: Presenting”) in top-left corner.
  - Gesture feedback (e.g., “Gesture: Next Slide” or “Next Slide ”) in bottomcenter with semi-transparent background for legibility.

## 2.3 Functionality

- **Real-Time Operation**: Processes webcam feed at 15–30 FPS on a standard CPU, suitable for live presentations.
- **Intuitive Gestures**: Thumbs up/down, open palm, and fist are natural and easy to learn.
- **Robustness**: MediaPipes hand detection handles varying lighting and backgrounds effectively.
- **Feedback**: Visual confirmation of gestures and actions enhances user experience.

### 2.4 Performance Considerations

- **Strengths**:
  - Lightweight heuristic-based detection avoids the need for a trained model, ensuring low latency.
  - Single-hand detection (max_num_hands=1) optimizes performance.
  - Frame flipping ensures intuitive gesture orientation.
- **Limitations**:
  - Heuristic thresholds (THUMB_THRESHOLD, DISTANCE_THRESHOLD) may require tuning for different users or environments.
  - Limited to one hand; multi-hand gestures are not supported.
  - Sensitive to partial hand occlusion or extreme lighting conditions.

### 2.5 Recommendations

- Train a machine learning model (e.g., SVM or neural network) on landmark data to improve gesture classification accuracy.
- Add a confidence threshold (e.g., consistent detection over 3 frames) to reduce false positives.
- Implement histogram equalization for better performance in low-light conditions.
- Support multi-hand gestures for advanced controls (e.g., zoom with two hands).

## 3 Emotion Recognition System

### 3.1 Overview

The emotion recognition system detects facial emotions in real-time using a YOLOv7-tiny model for face detection and a custom emotion detection module. It processes webcam or video input, annotates faces with emotion labels, and supports saving outputs as images or videos.

### 3.2 Technical Implementation

- **Libraries and Dependencies**:
  - PyTorch: For model loading and inference.
  - OpenCV: For video processing and visualization.
  - YOLOv7: Tiny variant for face detection (yolov7-tiny.pt).
  - Custom emotion module: For emotion classification (assumed to be pretrained).

  - NumPy, Pathlib: For data handling.

- **Face Detection**:
  - YOLOv7-tiny detects faces with a confidence threshold (conf_thres = 0.5) and IoU threshold (iou_thres = 0.45).
  - Non-maximum suppression (NMS) filters overlapping detections.
  - Supports FP16 (half-precision) on CUDA for faster inference.

- **Emotion Detection**:
  - Crops detected face regions and passes them to the detect_emotion function.
  - Returns emotion labels (e.g., happy, sad) and associated colors for visualization.

- **Processing Pipeline**:
  - Input: Webcam (source='0') or video file.
  - Preprocessing: Images resized to 512x512 (img_size=512), normalized to [0,1].
  - Inference: YOLOv7 detects faces, followed by emotion classification.
  - Visualization: Bounding boxes with emotion labels plotted on faces.

- **Output Options**:
  - Live display with optional FPS counter (show_fps).
  - Save as video (e.g., MP4) or images (e.g., PNG, JPG).
  - Configurable output path (output_path).

## 3.3 Functionality

- **Real-Time Emotion Detection**: Identifies emotions on detected faces, suitable for audience analysis.
- **Flexible Input**: Supports webcam, video files, or streams (e.g., RTSP).
- **Visualization**: Color-coded bounding boxes (e.g., happy: green) enhance interpretability.
- **Output Saving**: Saves annotated videos or images for post-analysis.

## 3.4 Performance Considerations

- **Strengths**:
  - YOLOv7-tiny is lightweight, enabling real-time inference on GPUs.
  - Supports half-precision (FP16) for faster CUDA performance.

  - Flexible output formats (video, image, folder) suit various use cases.

- **Limitations**:
  - Requires a GPU for optimal performance; CPU inference is slower.
  - Emotion detection module details are not provided, potentially limiting accuracy.
  - No explicit handling of low-light or occluded faces.
  - High computational load for multiple faces in a single frame.

### 3.5 Recommendations

- Optimize the emotion detection model (e.g., use ONNX or TensorRT) for faster inference.
- Add preprocessing (e.g., face alignment, illumination normalization) to improve robustness.
- Implement multi-threading to handle multiple faces efficiently.
- Provide a confidence threshold for emotion labels to filter uncertain predictions.

## 4 Integration Potential

Given the presenters background in emotion recognition, integrating the gesture control and emotion recognition systems could create a powerful presentation tool:

- **Combined Feedback**: Display presenter gestures and audience emotions simultaneously (e.g., gesture feedback in bottom-center, audience emotions in top-right).
- **Context-Aware Control**: Use audience emotions to adjust presentation pace (e.g., pause if boredom is detected).
- **Unified Pipeline**: Share webcam input between systems, using MediaPipe for hand detection and YOLOv7 for face detection in parallel threads.
- **Data Logging**: Record gestures and emotions for post-presentation analysis (e.g., engagement metrics).

## 5 Conclusion

The gesture control system provides an intuitive, real-time solution for hands-free slide navigation, leveraging MediaPipes robust hand detection. The emotion recognition system offers valuable insights into audience emotions, powered by YOLOv7 and a custom emotion classifier. Both systems are well-suited for presentation environments but could benefit from machine learning enhancements and robustness improvements. Integrating the two systems could enable a next-generation presentation tool that combines presenter control with audience feedback, enhancing engagement and interactivity.