**Hand Gesture Recognition**

**1. Objective**

The primary goal of this project is to design and implement a Python application that uses a webcam to recognize a predefined set of static hand gestures in real-time. This demonstrates the ability to work with live video data, select appropriate computer vision tools, and apply logical reasoning to interpret visual information.

**2. Background**

Hand gesture recognition is a key problem in Human-Computer Interaction (HCI). It enables natural, touch-free interfaces and has applications in AR/VR, robotics, gaming, and accessibility technologies.

The process involves:

1. Detecting and tracking a hand.
2. Extracting key features (landmarks).
3. Classifying these into meaningful gestures.

For this project, we implement a robust solution using computer vision + rule-based logic to ensure real-time gesture recognition.

**3. Detailed Task Description**

The application performs the following tasks:

* **Environment Setup** → Project folder with src/, requirements.txt, and README.md for easy reproducibility.
* **Webcam Integration** → Captures live video feed as input.
* **Hand Detection & Tracking** → Uses **MediaPipe Hands** for landmark extraction.
* **Gesture Logic Implementation** → Rule-based conditions to classify gestures into the defined vocabulary.
* **Real-Time Display** → Shows bounding landmarks and overlays the recognized gesture name on screen.

**4. Core Requirements**

The final application meets all requirements:

✅ **Gesture Vocabulary**

* Open Palm
* Fist
* Peace Sign (V-sign)
* Thumbs Up
* *(Additional)* Pointing (index finger only)

✅ **Technology Choice**

* **OpenCV** → for webcam video capture and display.
* **MediaPipe** → for fast and accurate hand landmark detection.
* **NumPy** → for distance/angle calculations.
* **scikit-learn + joblib** → for saving optional models (extendable).

✅ **Performance**

* Runs smoothly in real time (~30 FPS).
* Optimized for live gesture recognition without lag.

✅ **User Feedback**

* Recognized gesture name displayed clearly on the video window.

✅ **Code Quality**

* Clean, modularized Python code with inline comments.
* Separate file gestures.py for gesture logic.

**5. Technology Justification**

* **OpenCV**: Lightweight, fast video processing library.
* **MediaPipe**: Provides 21 reliable hand landmarks with minimal computation cost, making it ideal for real-time tasks.
* **NumPy**: Handles vector/matrix operations efficiently for gesture geometry checks.
* **scikit-learn**: Enables ML-based extension if we add more gestures.

**6. Gesture Logic Explanation**

* **Open Palm** → All finger tips extended away from palm.
* **Fist** → All finger tips curled towards palm.
* **Peace Sign (V)** → Index & middle finger extended, others folded.
* **Thumbs Up** → Thumb extended upwards, all other fingers curled.
* **Pointing** *(extra)* → Only index finger extended, others curled.

This logic is rule-based, calculated using distances between landmarks and relative positions.

**7. Setup & Execution Instructions**

1. Clone the repository:
2. git clone https://github.com/yourusername/hand-gesture-recognition.git
3. cd hand-gesture-recognition/src
4. Install dependencies:
5. pip install opencv-python mediapipe numpy scikit-learn joblib
6. Run the application:
7. python main.py
8. Press **Q** to quit the application.

**8. Demonstration**

* Demo video: demo.mp4
* The application correctly recognizes:
  + Open Palm
  + Fist
  + Peace Sign
  + Thumbs Up
  + Pointing

**9. Project Structure**

hand-gesture-recognition/

│

├── src/

│ ├── main.py # Entry point

│ ├── gestures.py # Gesture detection logic

│ └── utils.py # Helper functions

│

├── requirements.txt # Dependencies

├── README.md # Documentation

└── demo.mp4 # Screen recording of demo

**10. Functionalities**

* Live hand tracking with MediaPipe.
* Real-time gesture recognition (≥30 FPS).
* Supports 5 gestures (Open Palm, Fist, Peace, Thumbs Up, Pointing).
* Modular code → easy to extend with more gestures.

**11. Conclusion**

The project successfully implements real-time static hand gesture recognition using computer vision.  
It meets all requirements: gesture vocabulary, technology justification, performance, user feedback, and code quality.