IBM Gesture Trainer: Algorithm for AI Hand Gesture Trainer

Algorithm Details

🔹 1. Initialization and Setup

1.1. Import Libraries

- Load essential libraries like:
  - `cv2` for video capture and image processing.
  - `mediapipe` for hand landmark detection.
  - `sklearn` for training the Random Forest Classifier.
  - `tkinter` for GUI.
  - `pandas`, `numpy`, `joblib`, `threading` for data, modeling, and threading.

1.2. Set Constants

- Define file paths:
  - `gesture_data.csv` : stores collected feature data + labels.
  - `gesture_model.pkl` : saves trained model.

1.3. Initialize Mediapipe

- Set up the Mediapipe hand tracking system with:
  - 1 hand max.
  - Medium confidence thresholds.

🔹 2. Feature Extraction

2.1. Extract 3D Landmark Coordinates

- For each detected hand, extract:
  - 21 landmarks with x, y, z positions.
  - Flatten them into a 63-length feature vector `[x1, y1, z1, x2, y2, ..., x21, y21, z21]`.

🔹 3. Data Collection Logic

3.1. Start Webcam

- Open webcam using OpenCV.

3.2. Detect Hand Landmarks

- Flip and convert the frame to RGB.
- Use Mediapipe to detect hand landmarks.

3.3. Save Data

- If a hand is detected:
  - Draw landmarks on the frame (visual feedback).
  - Extract features using the helper.
  - Append label provided by user.
  - Save row to CSV (`gesture_data.csv`).

3.4. GUI Interaction

- User starts data collection from GUI → prompted to enter label.
- User presses "q" or clicks "Stop" to stop recording.

🔹 4. Model Training

4.1. Load Dataset

- Read `gesture_data.csv`.

4.2. Split into Features (X) and Labels (y)

- X = feature columns.
- y = target label.

4.3. Train Model

- Train a `RandomForestClassifier` (100 estimators).
- Save model as `gesture_model.pkl`.

🔹 5. Real-Time Prediction

5.1. Load Trained Model

- Load model from disk.

5.2. Start Webcam

- Begin live webcam feed.

5.3. Detect Hand and Predict

- For each frame:
  - Detect hand.
  - Extract features.
  - Predict label using the model.
  - Display prediction text on screen.

5.4. GUI Interaction

- Start/stop prediction using buttons.
- Press "q" to quit the prediction window.

🔹 6. GUI Controller (Tkinter)

6.1. Main Window

- Title: *AI Hand Gesture Trainer*.
- Buttons:
  - ✅ Start/Stop Data Collection
  - 📊 Train Model
  - 🤖 Start/Stop Live Prediction
  - ❌ Exit App

6.2. Actions Trigger Threads

- Long-running tasks like prediction and collection run in threads to keep GUI responsive.