PROJECT STATEMENT
(CS-474 Computer Vision)

Pose Estimation for Human-Computer Interaction

Objective:

Design and implement a robust pose estimation system capable of detecting and tracking the key points of the human body in real-time using a camera. The goal is to create a system that can estimate the 3D pose of a person, allowing for human-computer interaction, motion analysis, and gesture recognition. The aim is to delve deep into computer vision techniques, motion tracking, and 3D space mapping.

Project Description:

The core of this project is to build a pose estimation system using image or video data, identifying human key points (such as joints and limbs) and estimating their positions in space. The students will implement algorithms that can track human poses, identify specific gestures, and potentially even map them onto 3D space.

This project will allow to explore the following:

1. **Keypoint Detection**: Deep learning keypoint detection models like OpenPose or media-pipes, develop own using machine learning techniques and classical methods like SIFT & HOG..

2. **3D Pose Estimation**: Extend 2D pose estimation to 3D by calculating the depth of detected keypoints.

3. **Real-Time Processing**: Ensure the system runs in real-time by optimizing the performance for quick and efficient keypoint detection and tracking.

4. **Gesture Recognition (optional)**: Identify predefined gestures or activities, such as waving, pointing, sitting, or standing.

5. **Depth Estimation:** Measure the distance of the object from the camera using traditional methords or Deep Learning.

Input Specifications:

- Input data will be a stream of images or a video feed (e.g., from a webcam or pre-recorded video).
- The system should be capable of processing different poses and environmental settings (lighting, background, etc.).
- The pose estimation should work for single or multiple individuals in the frame.

Output Specifications:

1. **Keypoint Coordinates**:
   The output should provide the (x, y) coordinates for the detected keypoints (head, shoulders, elbows, wrists, knees, and ankles).

2. **3D Pose Estimation**:
   If extended to 3D, the output should include (x, y, z) coordinates for each keypoint.

3. **Real-Time Tracking**:
   The system should continuously track and update the keypoints for dynamic actions like walking, running, or interacting with objects.

4. **Gesture Recognition (Optional)**:
   Recognize specific gestures and return a label for each recognized gesture (e.g., "waving," "pointing").

5. **Visual Representation**:
   Visualize the output by overlaying the detected keypoints and the skeleton on the original image.

6. **Depth Map:**
   Display the depth map of the image demonstrating the distance of object from the camera.

Project Deliverables:

1. **Software Implementation:**
   - Detect Human body parts using classical keypoint detection methods (SIFT, HOG).
   - Implement pose estimation using Deep Learning models (e.g., OpenCV, TensorFlow, MediaPipe, OpenPose).
   - Compare Deep learning and Classical Methods.
   - Implement 3D pose estimation and real-time tracking.
   - Optionally, implement a gesture recognition system using machine learning models.
   - Create a user interface to display real-time feedback of pose tracking.
   - Display the depth and disparity map of the image.

2. **Theoretical Justification:**
   - Explain the algorithms used for keypoint detection and tracking.
   - Discuss the transformation from 2D to 3D pose estimation, including any challenges or assumptions.
   - Discuss performance optimization strategies for real-time processing, such as model quantization or edge processing.

3. **Testing and Evaluation:**
   - Evaluate the system's performance across various scenarios (lighting, multiple people, occlusion, etc.).

- Test the accuracy of keypoint detection and tracking with DL and Classical Methods..
- Optionally, test the gesture recognition system using a dataset of labeled gestures.

4. **Reporting:**
   - Write a detailed project report covering the design procedure, choice of algorithms, results, challenges faced, and solutions implemented.
   - Include a comparison of the system's accuracy for different scenarios.
   - Discuss any future work or improvements that could be made to the system.

Tools and Technologies:

- OpenCV for computer vision and image processing.
- MediaPipe or OpenPose for keypoint detection.
- TensorFlow or PyTorch for machine learning-based enhancements (if applicable).
- Monodepth2 for depth estimation
- Python or C++ for coding.
- (Optional) Unity or any 3D engine for visualizing 3D poses.

Note: This is a group project with a maximum of three members per group.