**Problem 1: People Fall detection**

**~Shri Meenaakshi S**

**21PC34**

**Approach:**

**Check Fall Conditions:**

1. Condition 1: Check if the vertical distance between the left shoulder and left foot is less than the length factor, and if the left shoulder is above both the left foot and left hip.
2. Condition 2: Similarly, check if the vertical distance between the right shoulder and right foot is less than the length factor, and if the right shoulder is above both the right foot and right hip.
3. Condition 3: Check if the difference between the height and width of the bounding box is negative. This condition helps detect situations where a person's pose is stretched vertically, indicating a potential fall.

No Fall Detected: If none of the conditions are met for any pose, the algorithm returns False, indicating no fall has been detected.

Return Fall Detection: If any of the conditions are met for any of the poses, the algorithm returns True, indicating a fall has been detected. It also returns the bounding box coordinates of the detected pose.

**Code Explanation:**

Loading the YOLOv7-POSE Model: The script loads the pre-trained YOLOv7-POSE model, which is capable of detecting human poses in images. The model is loaded onto either the CPU or GPU based on availability.

Processing Video Frames: The script reads each frame from the input video file and preprocesses it for pose detection.

Pose Detection: For each frame, the script utilizes the YOLOv7-POSE model to detect human poses. It applies non-maximum suppression to filter out redundant pose detections and retrieves keypoint coordinates for each detected pose.

Fall Detection Algorithm: The core of the fall detection process is a custom algorithm implemented in the fall\_detection function. This algorithm analyzes the keypoint coordinates of detected poses to determine if a fall has occurred. It considers factors such as the relative positions of body parts (e.g., shoulders, feet) and the ratio of height to width of the bounding box.

Annotating Fall Detection: When a fall is detected, the script annotates the corresponding frame with a red bounding box around the fallen person and adds a text label indicating "Person Fell down".

Output Generation: Annotated frames are written to an output video file, preserving the original video content while highlighting instances of fall detection.

**How Fall detection algorithm works:**

Iterate Over Detected Poses: The algorithm loops through each detected pose obtained from the YOLOv7-POSE model's output.

Calculate Bounding Box Coordinates: For each pose, it calculates the bounding box coordinates (xmin, ymin, xmax, ymax). These coordinates define the smallest rectangle that encompasses the entire pose.

Extract Key Body Part Positions: From the detected pose, specific key body part positions are extracted. These typically include coordinates for shoulders, hips, and feet.

Calculate Length Factor: The algorithm computes a length factor based on the distance between key body parts, such as shoulders and hips. This length factor is used as a reference for detecting falls.

Calculate Width and Height of Bounding Box: Using the bounding box coordinates, the algorithm calculates the width (dx) and height (dy) of the bounding box.

Cases :

