Assignment 4

1. Motion Detection Algorithm

For this project, I implemented a motion detection system using **Frame Differencing** combined with **Background Subtraction** to identify motion in real-time. Below is a summary of how the algorithm works:

- **Preprocessing**: Each frame from the video stream is first converted to **grayscale** to simplify processing by removing color information. A **Gaussian blur** is then applied to the grayscale frame to reduce noise and minimize small, irrelevant changes in the image.
- **Frame Differencing**: The motion detection algorithm compares consecutive frames using cv2.absdiff(). This function computes the absolute difference between the current frame and the previous frame, highlighting any regions where motion has occurred (i.e., where the pixel values have changed).
- **Thresholding**: After calculating the difference between frames, the resulting image is **thresholded** using cv2.threshold() to create a binary mask. This step eliminates small pixel variations that are not relevant to the motion detection and highlights the areas of significant motion.
- **Bounding Box Calculation (No Contours)**: Instead of using contours, which were disallowed by the assignment, the motion is localized by identifying the non-zero pixels in the binary image. A **bounding box** is then drawn around the area of motion using cv2.boundingRect() on the non-zero pixels. This bounding box visualizes the detected motion in the video feed.
- **Video Recording**: The detected motion is recorded using cv2.VideoWriter(), which saves the output video in MP4 format. This allows a clear demonstration of the motion detection algorithm in action.

2. Assumptions

Several assumptions were made during the development of this motion detector to simplify the problem and ensure its effectiveness:

- **Stationary Camera**: The algorithm assumes that the camera is stationary. Any movement of the camera itself would be interpreted as motion, leading to false

positives. This approach works best in environments where the camera remains fixed in position.

- **Consistent Lighting**: The algorithm assumes that the lighting conditions in the environment are relatively stable. Sudden changes in lighting (e.g., flickering lights or shadows) may be falsely interpreted as motion. The Gaussian blur applied during preprocessing helps mitigate some of these issues, but extreme changes in lighting may still cause problems.
- **Significant Motion Only**: The algorithm is designed to detect significant motion and ignores very small movements. For example, slight pixel changes or minor movements (e.g., leaves blowing in the wind) may not trigger the detection. This assumption simplifies the detection process by focusing on more substantial movements.
- **Low Noise in Video Feed**: The video stream is assumed to have minimal noise. While preprocessing steps like Gaussian blur help to smooth the image, a highly noisy video feed may still affect the accuracy of the motion detection.

Summary

- **Algorithm Used**: Frame Differencing and Background Subtraction combined with bounding box calculation (without contours).
- **Preprocessing**: Grayscale conversion and Gaussian blur are applied to reduce noise and simplify image processing.
- **Bounding Box Visualization**: Non-zero pixels from the thresholded image are used to compute bounding boxes that visualize detected motion, adhering to the requirement to avoid contours.
- **Assumptions**: The camera remains stationary, lighting is consistent, the algorithm focuses on significant motion, and the video feed has minimal noise.

This solution effectively detects and visualizes motion under the stated assumptions, and the recorded video output demonstrates the system in action.