CGC ASSIGNMENT – 2

Name: RAHUL VARMA

Roll No: S20200010212

Motion Detection Assignment:

Task of this assignment:

For the previous assignment code add CLAHE (**Contrast-limited adaptive histogram equalization)** to improve the quality of the frames.

- ➔ Histogram Equalization for Enhancement
- ➔ Advanced Histogram Equalization
- ➔ CLAHE

Now, when the image captures many rectangles the send back the rectangle to the processing unit and the reject the smaller rectangles.

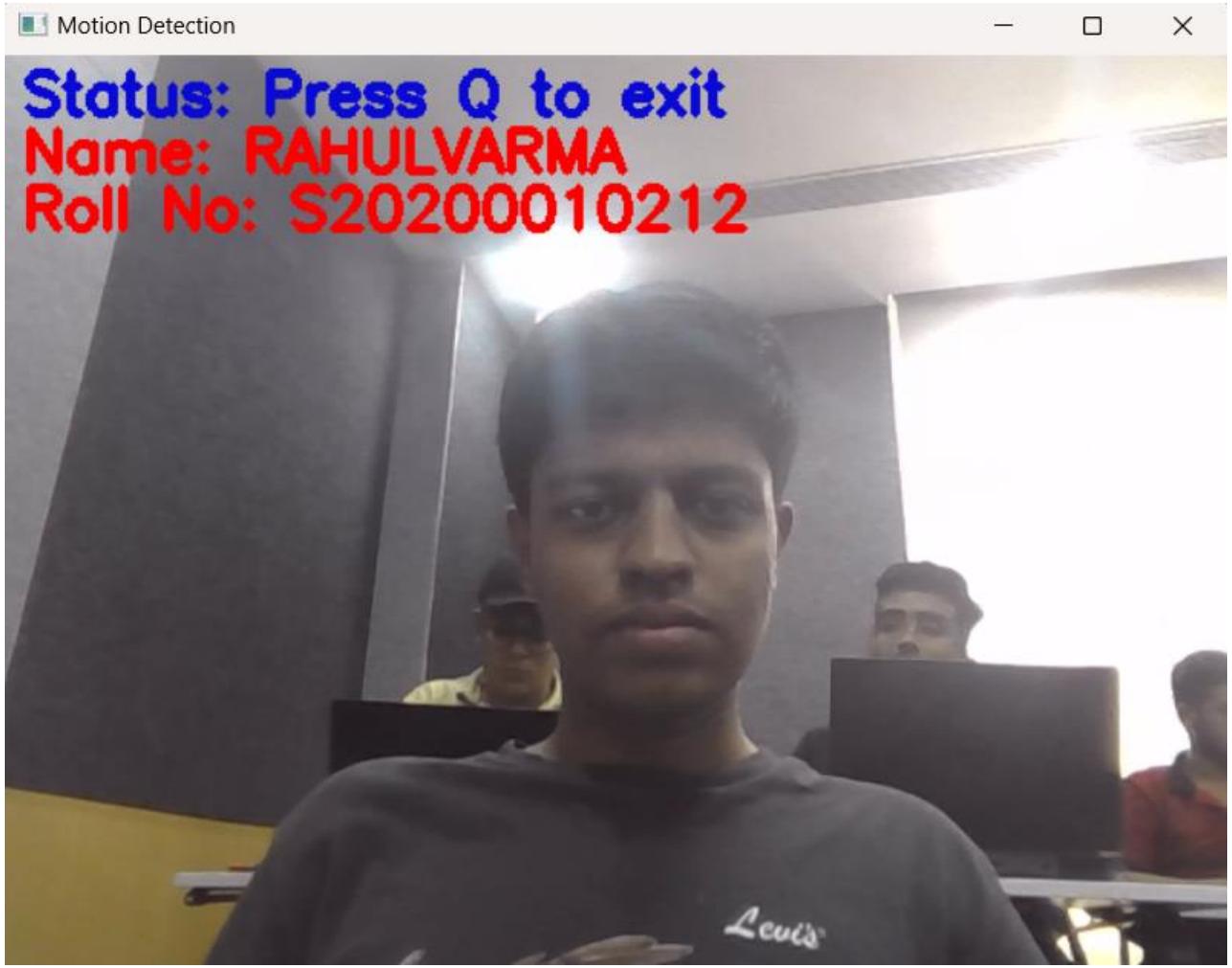
The Below code Implements a simple motion detection algorithm using computer vision techniques and the OpenCV library.

The algorithm captures video frames from the webcam, converts them to grayscale, applies a contrast limited adaptive histogram equalization (CLAHE) to improve the quality of the frames, calculates the difference between consecutive frames, performs a threshold operation to identify motion, dilates the thresholded image to fill in holes, finds contours in the dilated image, and finally draws rectangles around the contours to indicate motion.

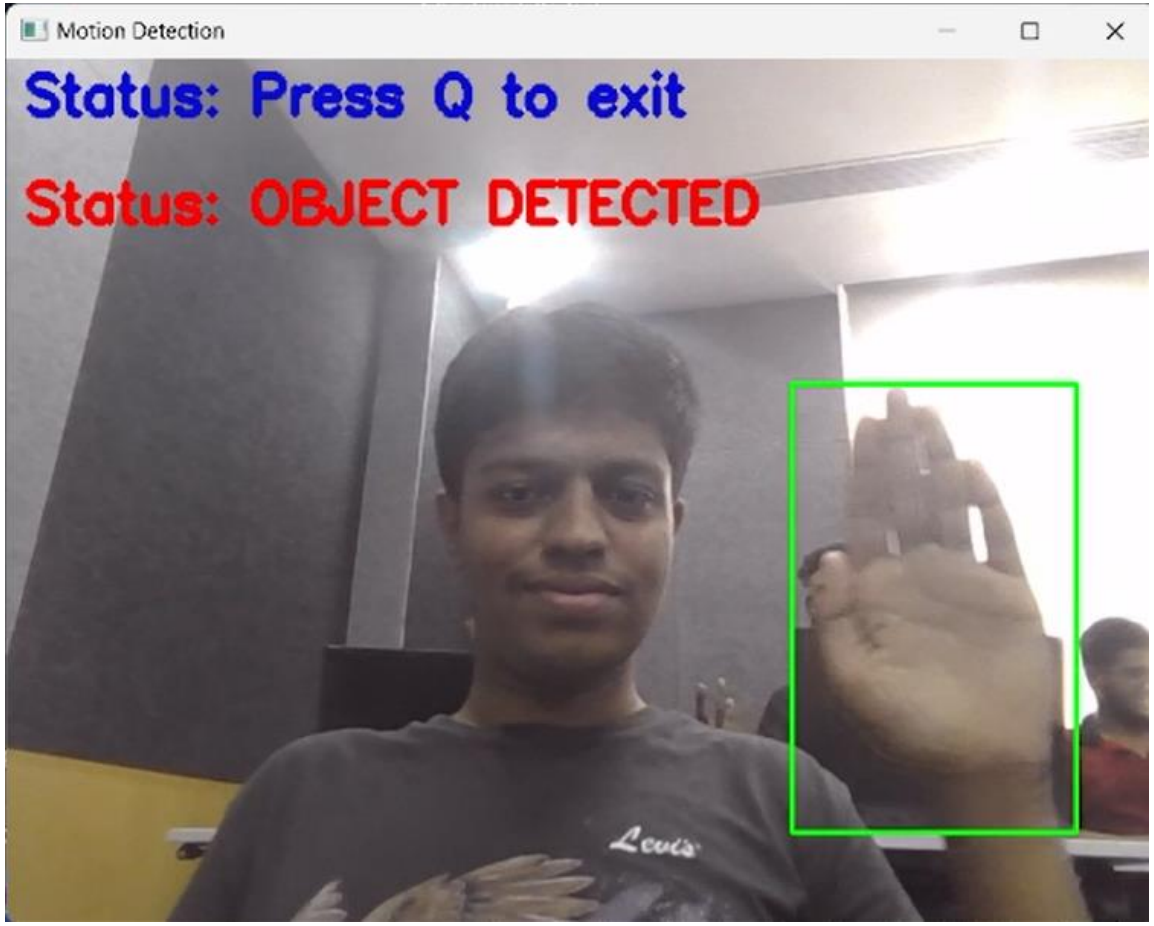
Rejecting the smaller rectangles and show only the large rectangle which is optimized

The motion detection is performed on the grayscale images to reduce computational complexity. The code continuously loops and updates the previous frame with the current frame, allowing for real-time motion detection.

Before Detection:



Motion Detected:



Code:

```python
import cv2

cap = cv2.VideoCapture(0)
# Applying CLAHE to improve contrast
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
min_contour_area = 2500

while True:
    ret, frame1 = cap.read()
    gray1 = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)

    gray1 = clahe.apply(gray1)

    ret, frame2 = cap.read()
    gray2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)
    gray2 = clahe.apply(gray2)
    diff = cv2.absdiff(gray1, gray2)
    thresh = cv2.threshold(diff, 25, 255, cv2.THRESH_BINARY)[1]

    dilated = cv2.dilate(thresh, None, iterations=2)
    cnts, _ = cv2.findContours(dilated, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    max_area = 0
    # Rejecting the smaller rectangles
    largest_rect = None
    for c in cnts:
        if cv2.contourArea(c) < min_contour_area:
            continue
        (x, y, w, h) = cv2.boundingRect(c)
        if cv2.contourArea(c) > max_area:
            max_area = cv2.contourArea(c)
            largest_rect = (x, y, w, h)
    # Detecting the object
    if largest_rect:
        (x, y, w, h) = largest_rect
        cv2.rectangle(frame1, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(frame1, "Status: {}".format('OBJECT DETECTED'), (10, 90), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
    else:
        cv2.putText(frame1, "Name: {}".format('RAHULVARMA'), (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)
        cv2.putText(frame1, "Roll No: {}".format('S20200010212'), (10, 90), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 3)

    cv2.putText
    cv2.putText(frame1, "Status: {}".format('Press Q to exit'), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (215, 10, 10), 3)
    cv2.imshow("Motion Detection", frame1)
    gray1 = gray2
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

```python
import cv2

cap = cv2.VideoCapture(0)
# Applying CLAHE to improve contrast
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
min_contour_area = 2500

while True:
    ret, frame1 = cap.read()
    gray1 = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)

    gray1 = clahe.apply(gray1)

    ret, frame2 = cap.read()
    gray2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)
    gray2 = clahe.apply(gray2)
    diff = cv2.absdiff(gray1, gray2)
    thresh = cv2.threshold(diff, 25, 255, cv2.THRESH_BINARY)[1]

    dilated = cv2.dilate(thresh, None, iterations=2)
    cnts, _ = cv2.findContours(dilated, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    max_area = 0
    # Rejecting the smaller rectangles
    largest_rect = None
    for c in cnts:
        if cv2.contourArea(c) < min_contour_area:
            continue
        (x, y, w, h) = cv2.boundingRect(c)
        if cv2.contourArea(c) > max_area:
            max_area = cv2.contourArea(c)
            largest_rect = (x, y, w, h)
```

```
    # Detecting the object
    if largest_rect:
        (x, y, w, h) = largest_rect
        cv2.rectangle(frame1, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(frame1, "Status: {}".format('OBJECT DETECTED'), (10, 90), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,
0, 255), 3)
    else:
        cv2.putText(frame1, "Name: {}".format('RAHULVARMA'), (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255),
3)
        cv2.putText(frame1, "Roll No: {}".format('S20200010212'), (10, 90), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0,
255), 3)

    cv2.putText
    cv2.putText(frame1, "Status: {}".format('Press Q to exit'), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (215, 10,
10), 3)
    cv2.imshow("Motion Detection", frame1)
    gray1 = gray2
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```