EXP:5 Multi-Object Tracking Using CSRT Tracker

Aim:

To track multiple objects in a video using OpenCV's `MultiTracker` with CSRT tracking algorithm.

Procedure:

1. **Load the Video File:**
   - Open a video file using `cv2.VideoCapture()`.
   - Read the first frame to initialize the tracking system.
2. **Prepare the Frame:**
   - Check if the frame is grayscale, and if so, convert it to BGR color format.
3. **Initialize Multi-Object Tracker:**
   - Use the CSRT tracker (`cv2.legacy.TrackerCSRT_create`).
   - Create a `MultiTracker` instance to handle multiple object tracking.
4. **Select Objects to Track:**
   - Manually select Regions of Interest (ROIs) from the first frame.
   - Each selection can be confirmed by ENTER or SPACE, and the selection process can be ended by pressing 'q'.
   - Assign random colors for each tracked object for visualization.

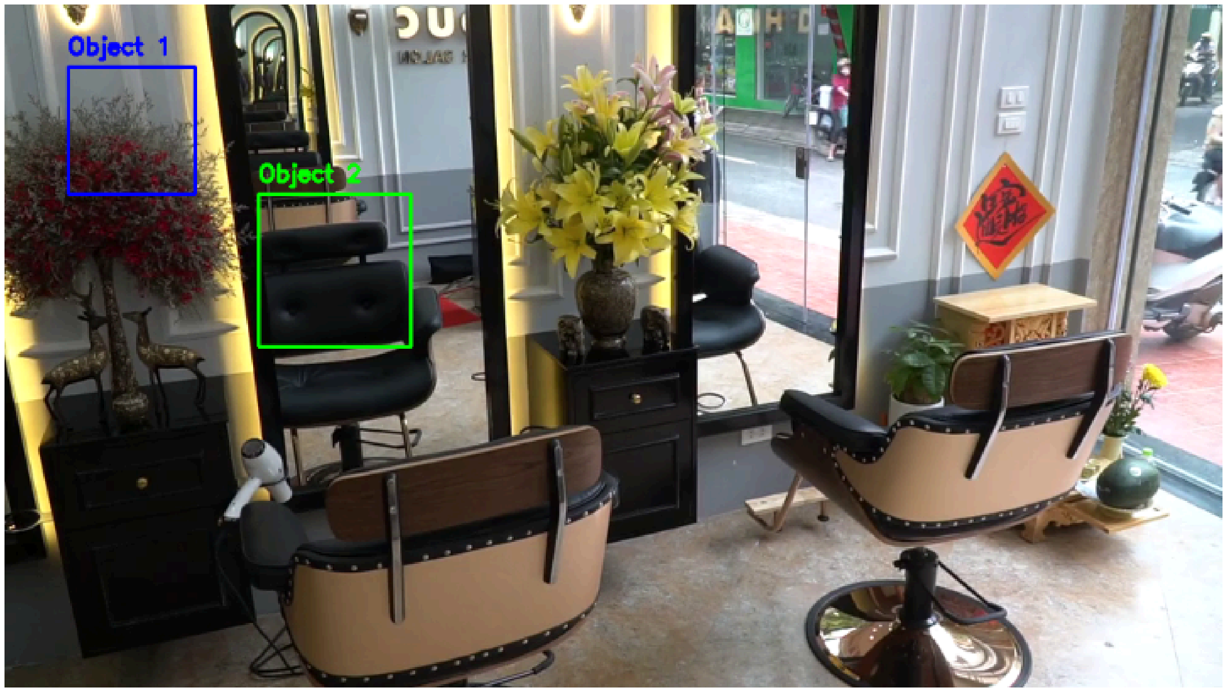
5. **Add Selected ROIs to MultiTracker:**
    - Add the manually selected bounding boxes to the `MultiTracker` along with individual tracker instances.
6. **Start Tracking Across Frames:**
    - Read frames sequentially from the video.
    - Update all trackers and retrieve the updated bounding box positions.
    - Draw rectangles and labels (e.g., Object 1, Object 2, etc.) around the tracked objects.
    - Show the output video frame-by-frame in a window.
7. **Exit Condition:**
    - Press 'q' at any time during video playback to stop tracking and exit the program.

Code:

```
import cv2

import numpy as np



# Load video

cap = cv2.VideoCapture("/content/29917-383980366_small.mp4")

if not cap.isOpened():

    print("Error: Could not open video.")

    exit()



# Read first frame
```

```
ret, frame = cap.read()

if not ret:

    print("Error: Failed to read the first frame.")

    exit()


print("Frame shape:", frame.shape)

print("Frame dtype:", frame.dtype)


# Convert grayscale to BGR if needed

if len(frame.shape) == 2:

    frame = cv2.cvtColor(frame, cv2.COLOR_GRAY2BGR)


# Initialize MultiTracker with CSRT

tracker = cv2.legacy.TrackerCSRT_create

multiTracker = cv2.legacy.MultiTracker_create()


# Select objects to track

print("Select the objects to track and press ENTER or SPACE. Press Q
to start tracking.")

bboxes = []

colors = []


while True:
    bbox = cv2.selectROI('MultiTracker', frame, fromCenter=False,
showCrosshair=True)

    bboxes.append(bbox)

    colors.append((np.random.randint(0, 255), np.random.randint(0,
255), np.random.randint(0, 255)))


    print("Press 'q' to quit selecting boxes and start tracking.")

    print("Press any other key to select another object.")


    k = cv2.waitKey(0) & 0xFF

    if k == ord('q'):

        break


cv2.destroyAllWindows()


# Add selected boxes to MultiTracker

for bbox in bboxes:

    print("Adding tracker for bbox:", bbox)

    multiTracker.add(tracker(), frame, bbox)


# Tracking loop

while cap.isOpened():

    ret, frame = cap.read()
    if not ret:

        print("End of video or failed to read frame.")

        break



    if len(frame.shape) == 2:

        frame = cv2.cvtColor(frame, cv2.COLOR_GRAY2BGR)



    success, boxes = multiTracker.update(frame)



    for i, newbox in enumerate(boxes):

        p1 = (int(newbox[0]), int(newbox[1]))

        p2 = (int(newbox[0] + newbox[2]), int(newbox[1] + newbox[3]))

        cv2.rectangle(frame, p1, p2, colors[i], 2, 1)

        cv2.putText(frame, f"Object {i+1}", (p1[0], p1[1] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.6, colors[i], 2)



    cv2.imshow('MultiTracker', frame)



    if cv2.waitKey(30) & 0xFF == ord('q'):

        break



cap.release()

cv2.destroyAllWindows()
```

Output:



Result:

Successfully implemented a multi-object tracking system using CSRT trackers with manual ROI selection. The system accurately tracks the selected objects through the video.