```
# save.py

import cv2
import sys
import imutils
import time
import csv

(major_ver, minor_ver, subminor_ver) = (cv2.__version__).split('.')
data = []
number_of_trackers = 2

if __name__ == '__main__' :

    # Set up tracker.
    # Instead of CSRT, you can also use

    tracker_types = ['BOOSTING', 'MIL','KCF', 'TLD', 'MEDIANFLOW', 'GOTURN',
'MOSSE', 'CSRT']
    tracker_type = tracker_types[7]

    trackerslist = []
    for i in range(number_of_trackers):


        if int(minor_ver) < 3:
            tracker = cv2.Tracker_create(tracker_type)
            trackerslist.append(tracker)
        else:
            if tracker_type == 'BOOSTING':
                tracker = cv2.TrackerBoosting_create()
                trackerslist.append(tracker)
            elif tracker_type == 'MIL':
                tracker = cv2.TrackerMIL_create()
                trackerslist.append(tracker)
            elif tracker_type == 'KCF':
                tracker = cv2.TrackerKCF_create()
                trackerslist.append(tracker)
            elif tracker_type == 'TLD':
                tracker = cv2.TrackerTLD_create()
                trackerslist.append(tracker)
            elif tracker_type == 'MEDIANFLOW':
                tracker = cv2.TrackerMedianFlow_create()
                trackerslist.append(tracker)
            elif tracker_type == 'GOTURN':
                tracker = cv2.TrackerGOTURN_create()
                trackerslist.append(tracker)
            elif tracker_type == 'MOSSE':
                tracker = cv2.TrackerMOSSE_create()
                trackerslist.append(tracker)
            elif tracker_type == "CSRT":
                tracker = cv2.TrackerCSRT_create()
                trackerslist.append(tracker)

# Read video
video = cv2.VideoCapture(r"C:\Users\louis\Desktop\test.mp4")
# video = cv2.VideoCapture(0) # for using CAM

# Exit if video not opened.
if not video.isOpened():
  print("Could not open video")
  sys.exit()

# Read first frame.
ok, frame = video.read()
if not ok:
  print ('Cannot read video file')
  sys.exit()
```

```
066|
067| # Define an initial bounding box
068| bboxlist = []
069| for i in range(number_of_trackers):
070|     bbox = (287, 23, 86, 320)
071|     bboxlist.append(bbox)
072|
073| # Uncomment the line below to select a different bounding box
074| for i in range(number_of_trackers):
075|     bboxlist[i] = cv2.selectROI(frame, False)
076|
077|
078| # Initialize tracker with first frame and bounding box
079| oklist = []
080| for i in range(number_of_trackers):
081|     ok = trackerslist[i].init(frame, bboxlist[i])
082|     oklist.append(ok)
083|
084| while True:
085|      # Read a new frame
086|      ok, frame = video.read()
087|      if not ok:
088|          break
089|
090|      # Start timer
091|      timer = cv2.getTickCount()
092|
093|      # Update tracker
094|      for i in range(number_of_trackers):
095|         ok, bbox = trackerslist[i].update(frame)
096|         oklist[i] = ok
097|         bboxlist[i] = bbox
098|
099|      # Calculate Frames per second (FPS)
100|      fps = cv2.getTickFrequency() / (cv2.getTickCount() - timer);
101|
102|      # Draw bounding box
103|      if ok:
104|          # Tracking success
105|          positions_temp = []
106|          for i in range(number_of_trackers):
107|             p1 = (int(bboxlist[i][0]), int(bboxlist[i][1]))
108|             p2 = (int(bboxlist[i][0] + bboxlist[i][2]), int(bboxlist[i][1] +
bboxlist[i][3]))
109|             cv2.rectangle(frame, p1, p2, (255*(i),255*(i-1),255*(i-2)), 2, 1)
110|
111|             #Update boxes position list
112|             positions_temp.append(bboxlist[i][0] + bboxlist[i][2]/2)
113|             positions_temp.append(bboxlist[i][1] + bboxlist[i][3]/2)
114|          data.append(positions_temp)
115|
116|      else :
117|          # Tracking failure
118|          cv2.putText(frame, "Tracking failure detected", (100,80),
cv2.FONT_HERSHEY_SIMPLEX, 0.75,(0,0,255),2)
119|
120|      # Display tracker type on frame
121|      cv2.putText(frame, tracker_type + " Tracker", (100,20),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (50,170,50),2);
122|
123|      # Display FPS on frame
124|      cv2.putText(frame, "FPS : " + str(int(fps)), (100,50),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (50,170,50), 2);
125|      # Display result
126|      cv2.imshow("Tracking", frame)
127|
128|      # Exit if ESC pressed
129|      if cv2.waitKey(1) & 0xFF == ord('q'): # if press SPACE bar
```

```
        break

video.release()
with open(r'C:\Users\louis\Desktop\demo.csv', 'w', newline='') as file:
    writer = csv.writer(file, delimiter = ';', lineterminator = '\n')
    writer.writerows(data)
    file.close()
cv2.destroyAllWindows()
```