EXPERIMENT 03 – BASIC MOTION DETECTION AND TRACKING

Aim: Create basic motion detection and tracking.

Algorithm:

1. Import Libraries: Import `torch` and `cv2`
2. Define `drawbox` Function: Create the `drawbox` function with `img` and `threshold` as inputs.
3. Predict Bounding Boxes: Use the model to predict bounding boxes on the input image.
4. Convert Predictions: Convert predictions to a pandas dataframe.
5. Loop Through Predictions: Iterate over each bounding box prediction.
6. Extract Coordinates: Extract `xmin`, `ymin`, `xmax`, `ymax`, `conf`, and `cls` for each box.
7. Apply Threshold: Check if `conf` is above the threshold.
8. Draw Box: Draw a rectangle on the image for valid predictions.
9. Annotate Image: Add text annotation with confidence and class label.
10. Capture and Display Video: Capture live video, process frames, and display annotated frames.

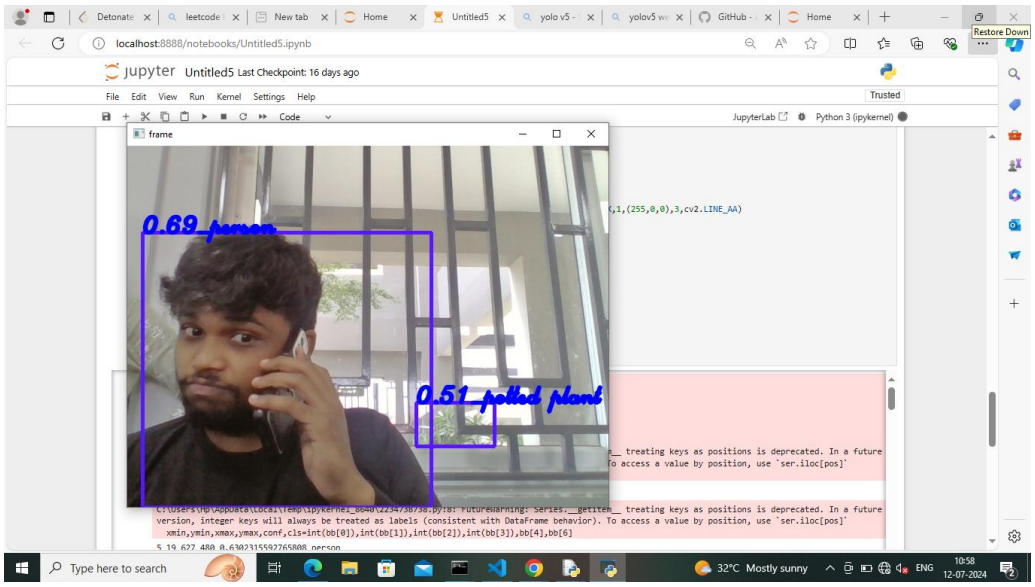
CODE:

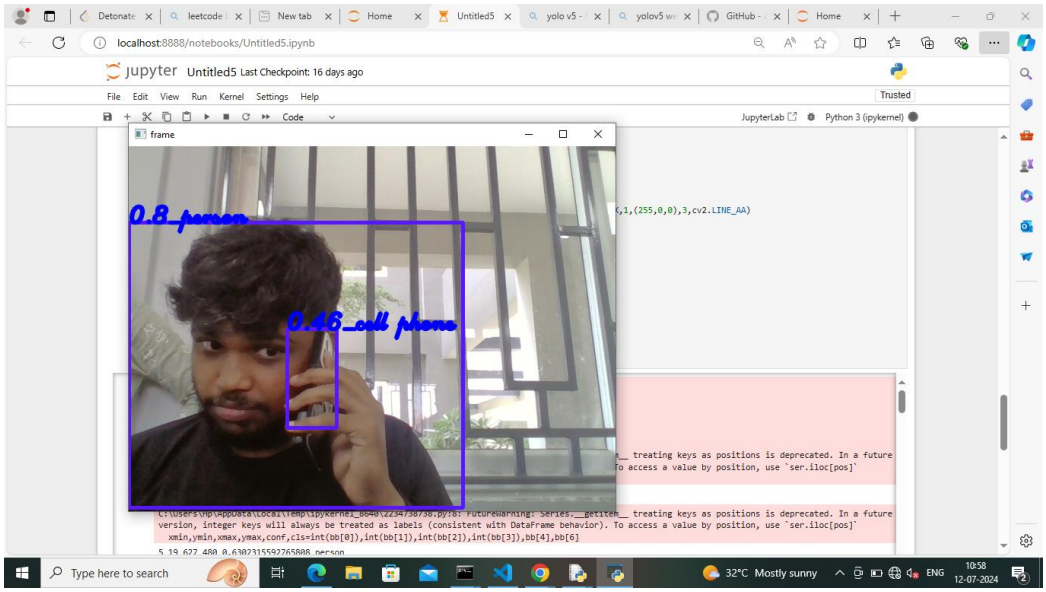
```
import torch
import cv2
def drawbox(img,threshold=0.4):
   q=model(img)
   df=q.pandas().xyxy[0]
   for i in df.index:
      bb=df.loc[i]
      xmin,ymin,xmax,ymax,conf,cls=int(bb[0]),int(bb[1]),int(bb[2]),int(bb[3]),bb[
4],bb[6]
      print(xmin,ymin,xmax,ymax,conf,cls)
      if threshold<=conf:
        cv2.rectangle(img,(xmin,ymin),(xmax,ymax),(255,23,89),4)
        cv2.putText(img,f'{round(conf,2)}_{cls}',(xmin,ymin),cv2.FONT_HERSHE
Y_SCRIPT_COMPLEX,1,(255,0,0),3,cv2.LINE_AA)
   return img,len(df)
model = torch.hub.load("ultralytics/yolov5", "yolov5s")
cap=cv2.VideoCapture(0)

#for live stream
```

```
while cap.isOpened():
    f=cap.read()[1]
    img,l=drawbox(f)
    cv2.imshow("frame",img)
    cv2.waitKey(1)
```

OUTPUT:





Result: The code has been executed successfully