Drown Detect

```
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np

# from picamera Import PiCamera
# camera = PiCamera
# camera.start_preview()
# open webcam
webcam = cv2.VideoCapture(0)

if not webcam.isOpened():
   print("Could not open webcam")
   exit()

t0 = time.time()  # gives time in seconds after 1970

# variable dcount stands for how many seconds the person has been standing still for
centre0 = np.zeros(2)
isDrowning = False

# this loop happens approximately every 1 second, so if a person doesn't move,
# or moves very little for 10seconds, we can say they are drowning

# loop through frames
```

```
while webcam.isOpened():

    # read frame from webcam
    status, frame = webcam.read()

    if not status:
        print("Could not read frame")
        exit()

    # apply object detection
    bbox, label, conf = cv.detect_common_objects(frame)
    # simplifying for only 1 person

    # s = (len(bbox), 2)

    if (len(bbox) > 0):
        bbox0 = bbox[0]
        # centre = np.zeros(s)
        centre = [0, 0]

        # for i in range(0, len(bbox)):
        # centre[i] =[(bbox[i][0]+bbox[i][2])/2,(bbox[i][1]+bbox[i][3])/2 ]

        centre = [(bbox0[0] + bbox0[2]) / 2, (bbox0[1] + bbox0[3]) / 2]

        # make vertical and horizontal movement variable
        hmov = abs(centre[0] - centre0[0])
        vmov = abs(centre[1] - centre0[1])
```

```
    # there is still need to tweek the threshold
    # this threshold is for checking how much the centre has moved

    x = time.time()

    threshold = 10
    if (hmov > threshold or vmov > threshold):
        print(x - t0, 's')
        t0 = time.time()
        isDrowning = False

    else:

        print(x - t0, 's')
        if ((time.time() - t0) > 10):
            isDrowning = True

    # print('bounding box: ', bbox, 'label: ' label ,'confidence: ' conf[0], 'centre: ', centre)
    # print(bbox,label ,conf, centre)
    print('bbox: ', bbox, 'centre:', centre, 'centre0:', centre0)
    print('Is he drowning: ', isDrowning)

    centre0 = centre
    # draw bounding box over detected objects

out = draw_bbox(frame, bbox, label, conf, isDrowning)

# print('Seconds since last epoch: ', time.time()-t0)
```

```
    # display output
    cv2.imshow("Real-time object detection", out)

    # press "Q" to stop
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# release resources
webcam.release()
cv2.destroyAllWindows()
```