```
                    cv2.FONT_HERSHEY_SIMPLEX, 0.3, (0, 255, 0), 1,
cv2.LINE_AA)

    # Stack both frames and show the image
    fgmask_3_f1 = cv2.cvtColor(fgmask_f1, cv2.COLOR_GRAY2BGR)
    fgmask_3_f2 = cv2.cvtColor(fgmask_f2, cv2.COLOR_GRAY2BGR)
    fgmask_3_f3 = cv2.cvtColor(fgmask_f3, cv2.COLOR_GRAY2BGR)

    stacked_frame = np.hstack((frame1, frame2, frame3))
    stacked_countours = np.hstack((fgmask_3_f1, fgmask_3_f2, fgmask_3_f3))
    stacked= np.vstack((stacked_frame, stacked_countours))
    cv2.imshow('Combined', cv2.resize(stacked, None, fx=0.90, fy=0.50))

    k = cv2.waitKey(40) & 0xff
    if k == ord('q'):
        break

# release video capture objects and close windows
cap1.release()
cap2.release()
cap3.release()
cv2.destroyAllWindows()
```

IntruderDetector.py

```
import cv2
import imutils
import numpy as np
import argparse


# model
HOGCV = cv2.HOGDescriptor()
HOGCV.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())


def detect(frame):
    bounding_box_cordinates, weights = HOGCV.detectMultiScale(
        frame, winStride=(4, 4), padding=(8, 8), scale=1.03)

    person = 1
    for x, y, w, h in bounding_box_cordinates:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```