

```

if not ret:
    break

# dim = (1024, 720)
dim = (720, 480)
frame = cv2.resize(frame, dim, interpolation=cv2.INTER_AREA)

# vertices = np.array(
#     [[(0, 0), (0, 200), (200, 200), (200, 0)]], dtype=np.int32)
# vertices = np.array(
#     [[(50, 50), (50, 50+300), (50+300, 50+300), (50+300, 50)]],
dtype=np.int32)
vertices = np.array(
    [[(250, 50), (250, 50+300), (250+300, 50+300), (250+300, 50)]],
dtype=np.int32)
mask = np.zeros_like(frame)

# # cv2.fillPoly(mask, vertices, (255, 255, 255))
cv2.fillPoly(mask, vertices, (255, 255, 255)) # BGR
# cv2.imshow('mask', mask)
masked_frame= frame.copy()
frame = cv2.bitwise_and(frame, mask)
# frame = cv2.resize(frame, dim, interpolation=cv2.INTER_AREA)
frame= cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
frame= cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
# frame= cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
# frame= cv2.cvtColor(frame, cv2.COLOR_GRAY2RGB)

# Calculate the Average FPS
frame_counter += 1
fps = (frame_counter / (time.time() - start_time))

# Display the FPS
cv2.putText(frame, 'FPS: {:.2f}'.format(
    fps), (20, 20), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 1)

image= frame

net.setInput(cv2.dnn.blobFromImage(image, 0.00392,
    (416, 416), (0, 0, 0), True, crop=False))
outs = net.forward(output_layers)

```