



อธิบายเทคนิค

1. ใช้ YOLOv2 เพื่อจำแนกวัตถุในภาพ

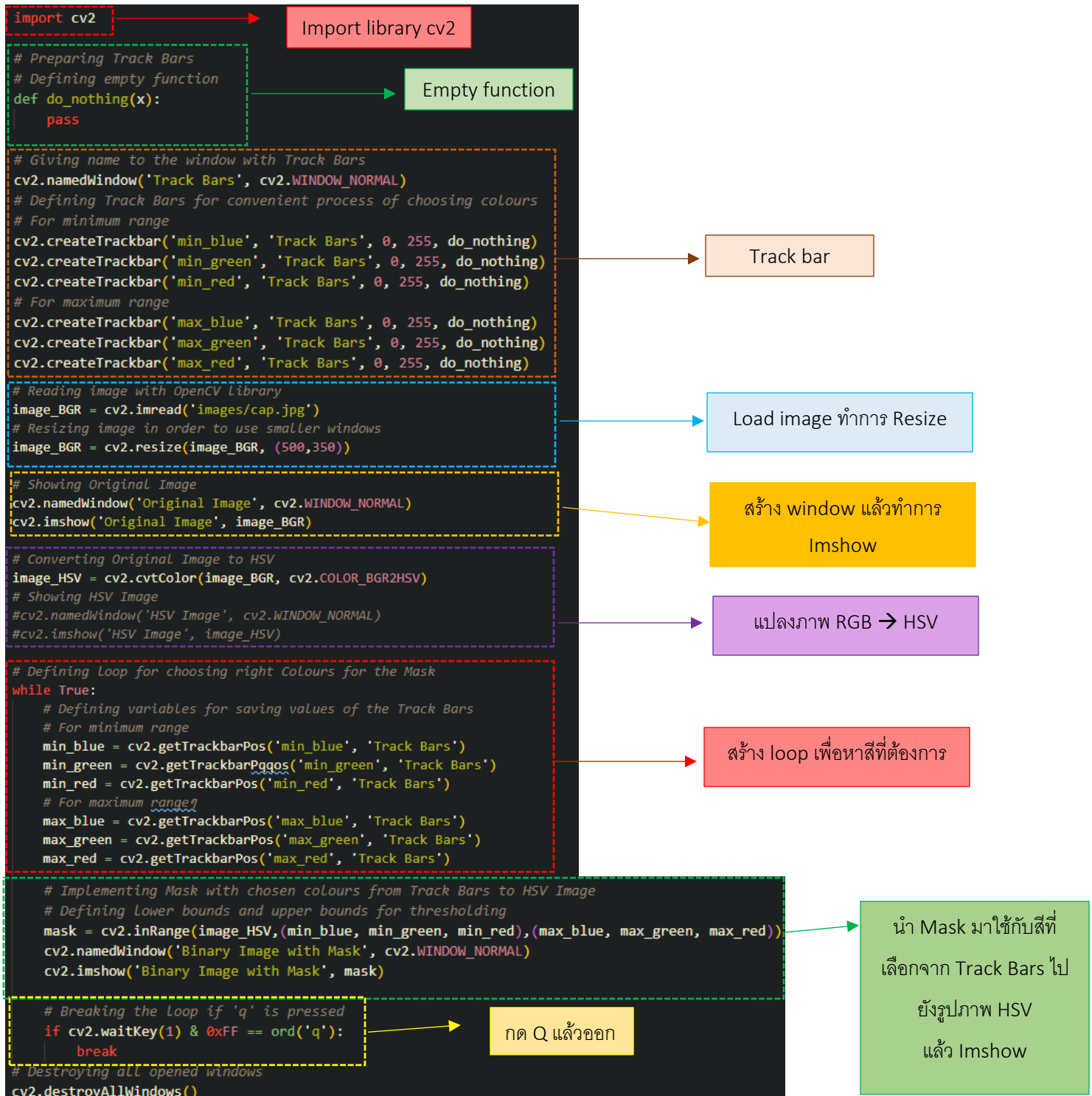
 yolov2.cfg	4/2/2020 11:30 AM	CFG File	3 KB
 yolov2.weights	4/2/2020 1:25 PM	WEIGHTS File	199,155 KB

2. ใช้โมเดลที่มีการเทรนไว้แล้ว เป็นดาต้าเซต COCO (80 objects)

 coco.names	11/2/2018 4:12 PM	NAMES File	1 KB
--	-------------------	------------	------

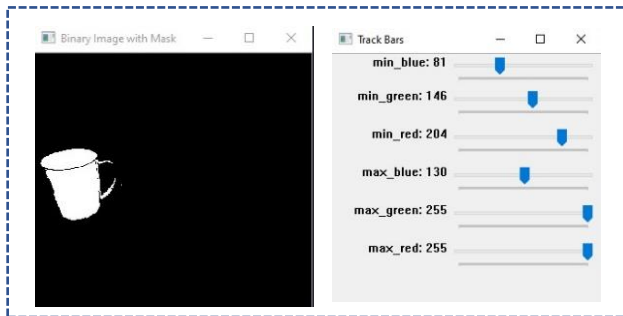
อธิบายการทำงานของโปรแกรม (Finding mark)

1. โค้ด (Code)

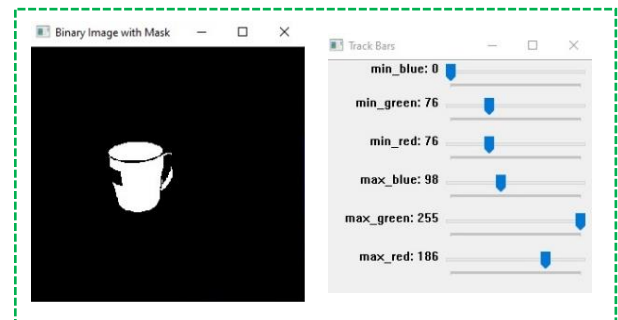


2. ผลลัพธ์

- Blue color



- Green color



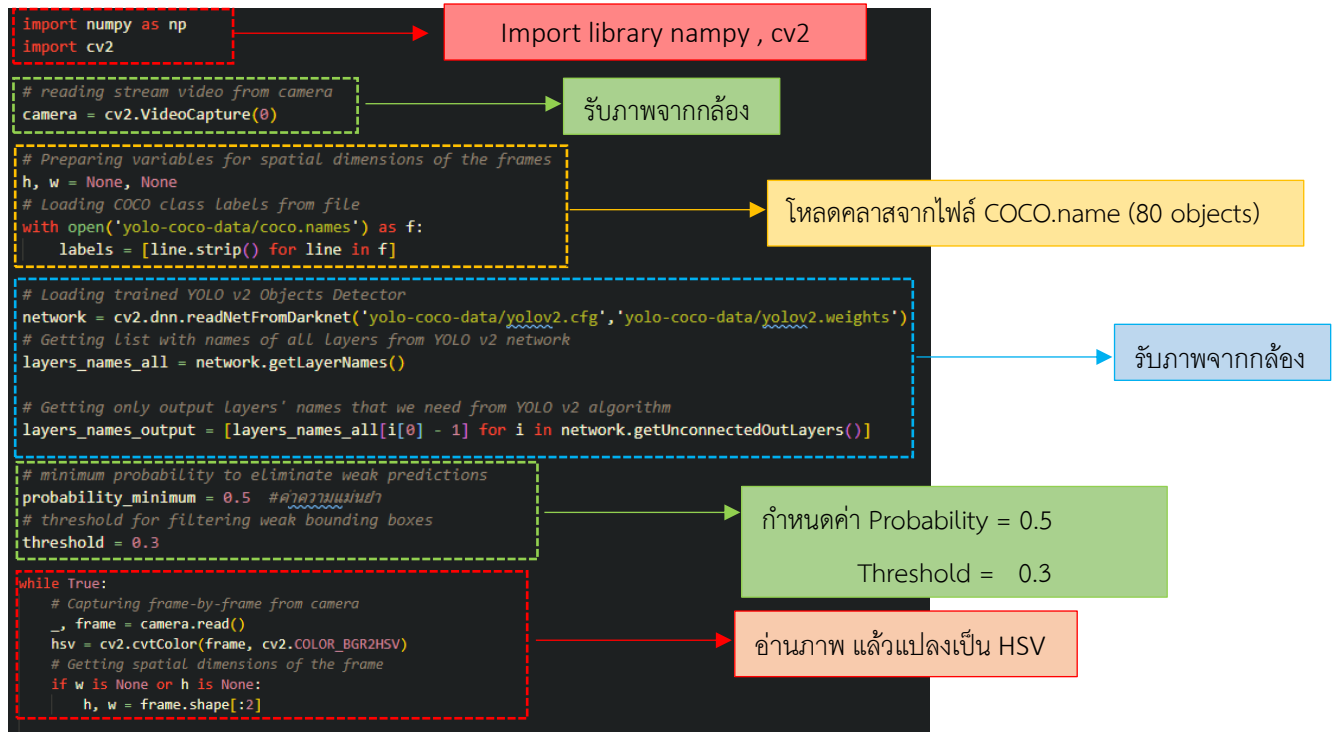
- Pink color



- Red color



อธิบายการทำงานของโปรแกรม (Yolo-v2-camera)



```
#defining the range of red color
red_lower=np.array([176,109,0],np.uint8)
red_upper=np.array([255,255,255],np.uint8)

#defining the Range of Blue color
blue_lower=np.array([81,146,204],np.uint8)
blue_upper=np.array([130,255,255],np.uint8)

#defining the Range of green color
green_lower=np.array([0,76,76],np.uint8)
green_upper=np.array([98,255,186],np.uint8)

#defining the Range of pink color
pink_lower=np.array([149,160,222],np.uint8)
pink_upper=np.array([162,255,255],np.uint8)
```

ค่าสูงสุด-ค่าต่ำสุดของสี ของวัตถุ ที่ต้องการหา
ซึ่งค่าตรงนี้ได้มาจากการ **finding mark**
จากขั้นตอนก่อนหน้า

```
#finding the range of red,blue and yellow color in the image
red = cv2.inRange(hsv, red_lower, red_upper)
blue = cv2.inRange(hsv,blue_lower,blue_upper)
green = cv2.inRange(hsv,green_lower,green_upper)
pink = cv2.inRange(hsv, pink_lower, pink_upper)
```

หา Range ของสีจากภาพที่รับมา

```
#Morphological transformation, Dilation
kernal = np.ones((5 ,5), "uint8")
#RED
red = cv2.dilate(red, kernal)
res = cv2.bitwise_and(frame, frame, mask = red)
#Blue
blue = cv2.dilate(blue,kernal)
res1 = cv2.bitwise_and(frame, frame, mask = blue)
#Green
green = cv2.dilate(green,kernal)
res2 = cv2.bitwise_and(frame, frame, mask = green)
#Pink
pink = cv2.dilate(pink, kernal)
res3 = cv2.bitwise_and(frame, frame, mask = pink)
```

ใช้ Morphological
transformation, Dilation

```
#สร้าง BLOB Detection
# blob = cv2.dnn.blobFromImage(image, scalefactor=1.0, size, mean, swapRB=True)
blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (416, 416),
                                swapRB=True, crop=False)

network.setInput(blob)
output_from_network = network.forward(layers_names_output)
```

สร้าง BLOB Detection
ทำการกำหนด scale factor
ตามที่ต้องการ

```
#...Getting bounding boxes...#
bounding_boxes = []
confidences = []
class_numbers = []
# Going through all output layers after feed forward pass
for result in output_from_network:
    for detected_objects in result:
        scores = detected_objects[5:]
        class_current = np.argmax(scores)
        confidence_current = scores[class_current]
        if confidence_current > probability_minimum:
            box_current = detected_objects[0:4] * np.array([w, h, w, h])
            x_center, y_center, box_width, box_height = box_current
            x_min = int(x_center - (box_width / 2))
            y_min = int(y_center - (box_height / 2))
            # Adding results into prepared lists
            bounding_boxes.append([x_min, y_min, int(box_width), int(box_height)])
            confidences.append(float(confidence_current))
            class_numbers.append(class_current)
```

สร้าง อาร์เรย์ว่าง 3 ตัวคือ
bounding_boxes,
confidences,
class_numbers

For loop ตรวจสอบวัตถุโดยที่ถ้าหาก
ค่า confidence มีค่ามากกว่า
probability (0.5)
ให้ทำการ detect object นั้น ๆ

```

#Tracking the Red Color
contours, hierarchy = cv2.findContours(red, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area>300):
        x,y,w,h = cv2.boundingRect(contour)
        frame = cv2.rectangle(frame, (x,y), (x+w,y+h), (0,0,255),2)
        cv2.putText(frame,"RED color",(x,y),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,0,255))
#Tracking the Blue Color
contours, hierarchy = cv2.findContours(blue, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area>300):
        x,y,w,h = cv2.boundingRect(contour)
        frame = cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0),2)
        cv2.putText(frame,"Blue color",(x,y),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255,0,0))
#Tracking the green Color
contours, hierarchy = cv2.findContours(green, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area>300):
        x,y,w,h = cv2.boundingRect(contour)
        frame = cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0),2)
        cv2.putText(frame," Green color",(x,y),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0,255,0))
#Tracking the pink Color
contours, hierarchy = cv2.findContours(pink, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area>300):
        x,y,w,h = cv2.boundingRect(contour)
        frame = cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,255),2)
        cv2.putText(frame,"Pink color",(x,y),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255,0,255))

```

ทำการ find Contours เช็ค
ว่าวัตถุมีสี (Color) อะไร
แล้วทำการวาดกรอบ
(Rectangle) กับ เขียน
ข้อความ (PutText)

```

#...Non-maximum suppression...#
results = cv2.dnn.NMSBoxes(bounding_boxes, confidences, probability_minimum, threshold)
# Checking if there is at least one detected object
if len(results) > 0:
    for i in results.flatten(): #ฟังก์ชันที่หาให้เรียงเป็นมิติเดียวคือ flatten()
        # Getting current bounding box coordinates,
        # its width and height
        x_min, y_min = bounding_boxes[i][0], bounding_boxes[i][1]
        box_width, box_height = bounding_boxes[i][2], bounding_boxes[i][3]
        colour_box_current = (155,0,155) #สีม่วง
        # Preparing text with label and confidence for current bounding box
        text_box_current = '{}'.format(labels[int(class_numbers[i])], confidences[i])
        # Putting text with label and confidence on the original image
        cv2.putText(frame, text_box_current,
            (x_min, y_min - 25), cv2.FONT_HERSHEY_COMPLEX, 0.7, colour_box_current, 2)

```

ตรวจจับวัตถุ แล้ว
PutText ชื่อวัตถุ
ขึ้น

```

# Showing current frame with detected objects
cv2.namedWindow('YOLO v2 Real Time Detections', cv2.WINDOW_NORMAL)
# Pay attention! 'cv2.imshow' takes images in BGR format
cv2.imshow('YOLO v2 Real Time Detections', frame)
# Breaking the loop if 'q' is pressed
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

```

สร้าง window
แล้ว Imshow
จะหยุดก็ต่อเมื่อกด Q

```

# Releasing camera
camera.release()
# Destroying all opened OpenCV windows
cv2.destroyAllWindows()

```

ผลลัพธ์

