



รายงาน

Assignment 2

กลุ่ม 4 Square

สมาชิกในกลุ่ม

6610110214 พิรณัฐ ปถมกุล (กัส)

6610110126 ธัญพิสิษฐ์ แสงส่อง (ปริน)

6610110492 ทักษลักษณ์ แยมมยาสุจริต (อุ)

6610110475 ณัฐดนัย ชูกุล (เกมส์)

รายงานฉบับนี้เป็นส่วนหนึ่งของการเรียนวิชา 241-251 AI for robot controlling module

สาขาวิชา ปัญญาประดิษฐ์ มหาวิทยาลัย สงขลานครินทร์ ภาคเรียนที่ 1 ปีการศึกษา 2567

## คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่ออธิบายกระบวนการตรวจจับและจำแนกวัตถุด้วยเทคโนโลยีกล้อง ซึ่งทำงานร่วมกับการโปรแกรมหุ่นยนต์อัจฉริยะ เพื่อให้หุ่นยนต์สามารถตรวจจับและจำแนกวัตถุได้อย่างมีประสิทธิภาพ การพัฒนาและทดสอบในครั้งนี้นับเป็นก้าวสำคัญของการประยุกต์ใช้เทคโนโลยี หุ่นยนต์ในการแก้ไขปัญหาจริงในสถานการณ์ที่ซับซ้อนและมีความท้าทาย

## สารบัญ

คำนำ	1
สารบัญ	2
บทนำ	3
วัตถุประสงค์	4
ที่มาและความสำคัญ	4
ขอบเขตการศึกษา	4
ขั้นตอนการดำเนินการ	4
เปิดใช้งานหุ่น	6
เปิดใช้งานกล้อง	7
สร้าง function ในการหา contour ที่มากที่สุดในการ detect ลูกโป่ง และ ขวดน้ำ	7
รับรูป	10
กำหนด range ของสีโป่ง และ ขวดน้ำ	11
สร้าง mask	11
ค้นหา Contour สำหรับโป่งและขวด	12
การตรวจสอบว่าเจอ Contour หรือไม่	13
Code ส่วนเดินรถ	13
แสดงผลลัพธ์	13
function ไว้สั่งหยุด	13
ปิดการทำงาน	14
หน้าที่ของสมาชิก	16

## บทนำ

การโปรแกรมหุ่นยนต์เพื่อตรวจจับและจำแนกวัตถุ เป็นกระบวนการที่ต้องอาศัยการจัดการข้อมูลอย่างมีระบบและรอบคอบ โดยอ้างอิงจากข้อมูลที่ได้รับจากสภาพแวดล้อมผ่านกล้องของหุ่นยนต์ ซึ่งช่วยให้หุ่นยนต์สามารถทำการจำแนกและแยกแยะวัตถุได้อย่างถูกต้องและมีประสิทธิภาพ

### วัตถุประสงค์

1. เพื่อศึกษาการทำงานของ OpenCV ในการประมวลผลภาพและการตรวจจับวัตถุ
2. เพื่อพัฒนาความสามารถของหุ่นยนต์ในการจำแนกและแยกแยะวัตถุจากภาพที่ได้จากกล้อง

### ที่มาและความสำคัญ

คณะผู้จัดต้องการที่จะศึกษาเกี่ยวกับ OpenCV นำมาประยุกต์ใช้กับหุ่น robomaster ในรายวิชา ARTIFICIAL INTELLIGENCE FOR ROBOT CONTROLLING MODULE เพื่อให้หุ่นยนต์จำแนกและแยกแยะวัตถุจากภาพได้ และสามารถนำไปต่อยอด เพื่อใช้ประโยชน์อื่นๆ ในอนาคตได้

### ขอบเขตการศึกษา

1. ใช้ กล้อง(DJI camera 720P) จากหุ่น robomaster
2. ใช้ sensor ที่ล้อ เพื่อใช้หาตำแหน่งที่หุ่นเดิน ด้วย sub\_position
3. ใช้หุ่น robomaster ในการศึกษาและทดลอง

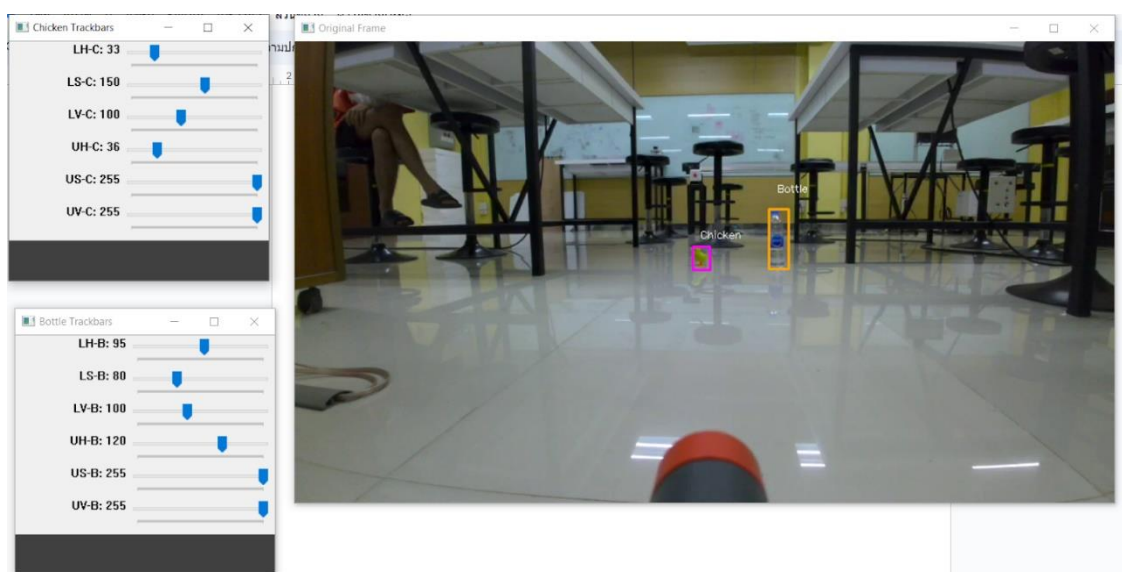
### ขั้นตอนการดำเนินการ

- 1.หาช่วงสีที่เหมาะสมกับ ลูกไก่ และ ขวดน้ำ
- 2.เลือกใช้ วิธีที่เหมาะสมในการ detect โดยจะเลือกใช้วิธี contour
- 3.เขียน code ตรวจจับไก่ และ ฉลากของขวดน้ำ ด้วยวิธี contour
- 4.ปรับขนาด bounding box ให้เหมาะสม กับแต่ละระยะ
- 5.ทดลองและแก้ไขข้อผิดพลาด

การปรับแต่งช่วงค่าสี HSV สำหรับการตรวจจับวัตถุด้วย Trackbars

สำหรับการจับภาพในโหมด HSV (Hue, Saturation, Value) จะใช้ Trackbars

เพื่อปรับค่า HSV แบบเรียลไทม์สำหรับการตรวจจับวัตถุสองประเภท ได้แก่ ไก่ (Chicken) และ ขวด (Bottle) (ณัฐดนัย ชูกุล (เกมส์))



ภาพแสดงปรับค่า HSV แบบเรียลไทม์สำหรับการตรวจจับวัตถุสองประเภท

1. สร้าง Trackbars สำหรับการปรับค่า HSV ใช้ฟังก์ชัน `cv2.createTrackbar()`

เพื่อสร้างแถบเลื่อน (Trackbars) สำหรับปรับค่า Hue (H), Saturation (S), และ Value (V) ในการตรวจจับวัตถุ โดยมีการสร้าง Trackbars แยกกันสำหรับ ไก่ และ ขวด เพื่อให้สามารถปรับค่าช่วงสีที่เหมาะสมสำหรับการตรวจจับแต่ละวัตถุ

# Create Trackbars for HSV adjustment for Chicken

```
cv2.namedWindow("Chicken Trackbars")
cv2.createTrackbar("LH-C", "Chicken Trackbars", 33, 179, nothing)
cv2.createTrackbar("LS-C", "Chicken Trackbars", 150, 255, nothing)
cv2.createTrackbar("LV-C", "Chicken Trackbars", 100, 255, nothing)
cv2.createTrackbar("UH-C", "Chicken Trackbars", 36, 179, nothing)
cv2.createTrackbar("US-C", "Chicken Trackbars", 255, 255, nothing)
cv2.createTrackbar("UV-C", "Chicken Trackbars", 255, 255, nothing)
```

# Create Trackbars for HSV adjustment for Bottle

```
cv2.namedWindow("Bottle Trackbars")
cv2.createTrackbar("LH-B", "Bottle Trackbars", 95, 179, nothing)
cv2.createTrackbar("LS-B", "Bottle Trackbars", 80, 255, nothing)
cv2.createTrackbar("LV-B", "Bottle Trackbars", 100, 255, nothing)
cv2.createTrackbar("UH-B", "Bottle Trackbars", 120, 179, nothing)
cv2.createTrackbar("US-B", "Bottle Trackbars", 255, 255, nothing)
cv2.createTrackbar("UV-B", "Bottle Trackbars", 255, 255, nothing)
```

2. การอ่านค่าจาก Trackbars เพื่อกำหนดช่วงค่า HSV ในรูปแบบของโปรแกรมจะใช้ฟังก์ชัน `cv2.getTrackbarPos()` เพื่อดึงค่าปัจจุบันของแต่ละ Trackbar ที่ปรับเปลี่ยนแล้ว จากนั้นนำค่าที่ได้ไปสร้างเป็นช่วงค่าของ HSV ที่ใช้ในการตรวจจับไก่และขวด

# Get HSV values for Chicken from trackbars

```
lh_c = cv2.getTrackbarPos("LH-C", "Chicken Trackbars")
ls_c = cv2.getTrackbarPos("LS-C", "Chicken Trackbars")
lv_c = cv2.getTrackbarPos("LV-C", "Chicken Trackbars")
uh_c = cv2.getTrackbarPos("UH-C", "Chicken Trackbars")
us_c = cv2.getTrackbarPos("US-C", "Chicken Trackbars")
uv_c = cv2.getTrackbarPos("UV-C", "Chicken Trackbars")
```

# Get HSV values for Bottle from trackbars

```
lh_b = cv2.getTrackbarPos("LH-B", "Bottle Trackbars")
ls_b = cv2.getTrackbarPos("LS-B", "Bottle Trackbars")
lv_b = cv2.getTrackbarPos("LV-B", "Bottle Trackbars")
uh_b = cv2.getTrackbarPos("UH-B", "Bottle Trackbars")
us_b = cv2.getTrackbarPos("US-B", "Bottle Trackbars")
uv_b = cv2.getTrackbarPos("UV-B", "Bottle Trackbars")
```

การ detect วัตถุ (พีรณัฐ ปถมกุล (กัส))

Contour Detection เป็นเทคนิคสำคัญในการประมวลผลภาพดิจิทัล (Digital Image Processing) โดยใช้ในการตรวจจับและติดตามขอบเขตของวัตถุในภาพ

โดยใน code จะมีหลักการทำงานดังนี้

1.เปิดใช้งานหุ่น

```
if __name__ == "__main__":
    ep_robot = robot.Robot()
    ep_robot.initialize(conn_type="ap")
    ep_camera = ep_robot.camera
    ep_gimbal = ep_robot.gimbal
    ep_chassis = ep_robot.chassis
```



ภาพ แสดงการเชื่อมต่อของหุ่น robomaster sdk

## 2.เปิดใช้งานกล้อง

```
ep_gimbal.recenter(pitch_speed=200, yaw_speed=200).wait_for_completed()
ep_camera.start_video_stream(display=False, resolution=camera.STREAM_720P)
ep_gimbal.moveto(pitch=-10, yaw=0).wait_for_completed()
```



ภาพ แสดงตำแหน่งของกล้อง

- recenter ของ gimbal ให้หันไปด้านหน้า
- ใช้รูปที่ได้จาก กล้องด้วยความละเอียด 720p
- หันกล้องลงเล็กน้อย เพื่อให้เวลาเข้าไปใกล้แล้วไม่หลุดเฟรม

## 3. สร้าง function ในการหา contour ที่มากที่สุดในการ detect ลูกไก่ และ ขวดน้ำ

### 3.1 Detect ลูกไก่

- การค้นหา Contour ที่ใหญ่ที่สุด

ฟังก์ชัน `max()` ถูกใช้เพื่อเลือก contour ที่มีพื้นที่มากที่สุดจากลิสต์ `chick_contours` ซึ่งเก็บ contours ที่ตรวจจับได้ในภาพ

```
max_chick_contour = max(chick_contours, key=cv2.contourArea)
```

- การคำนวณขนาดและตำแหน่งของกรอบสี่เหลี่ยมที่ล้อมรอบ Contour

ฟังก์ชัน `cv2.boundingRect()` จะคำนวณตำแหน่ง (x, y) และขนาด (width, height) ของกรอบสี่เหลี่ยมที่ล้อมรอบ contour ที่ใหญ่ที่สุด

```
(x_chick, y_chick, w_chick, h_chick) = cv2.boundingRect(max_chick_contour)
```



-การปรับขนาดกรอบสี่เหลี่ยมตามความกว้าง

หากความกว้างของกรอบมากกว่า 65 พิกเซล จะคำนวณค่าปรับขนาดที่แตกต่างจากกรณีอื่น โดยเพิ่มค่าของ w,h เพื่อให้กรอบขยายใหญ่ขึ้นและปรับตำแหน่งของ Y ใหม่

```
if w_chick > 65:
    adj_w_chick = int(w_chick * 0.3)
    adj_h_chick = int(h_chick * 0.6)
    new_y_chick = y_chick - adj_h_chick // 2 + 20
else:
    adj_w_chick = int(w_chick * 0.3)
    adj_h_chick = int(h_chick * 0.5)
    new_y_chick = (y_chick - adj_h_chick // 2 + 20) - 16
```

```
new_x_chick = x_chick - adj_w_chick // 2
new_w_chick = w_chick + adj_w_chick
new_h_chick = h_chick + adj_h_chick
```

-การวาดกรอบสี่เหลี่ยมบนภาพ

```
cv2.rectangle(frame, (new_x_chick, new_y_chick), (new_x_chick + new_w_chick, new_y_chick + new_h_chick), (255, 0, 255), 2)
```

ใช้ cv2.rectangle วาดกรอบสี่เหลี่ยมที่ล้อมรอบไก่ในภาพ โดยใช้สีชมพูและความหนาของเส้น 2 พิกเซล

-การใส่ข้อความ "Chicken" บนภาพ

```
cv2.putText(frame, f"Chicken", (x_chick, y_chick - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (250, 250, 250), 1)
```

ใช้ cv2.putText ใส่ข้อความ "Chicken" ใกล้กับตำแหน่งของกรอบ โดยใช้สีขาวและขนาดตัวอักษร 0.4

### 3.2 Detect ขวดน้ำ

ฟังก์ชัน detect\_bottle() ถูกออกแบบมาเพื่อการตรวจจับและทำเครื่องหมายตำแหน่งของ "ฉลากขวดน้ำ"

-การค้นหา Contour ที่ใหญ่ที่สุด

```
max_bottle_contour = max(bottle_contours, key=cv2.contourArea)
```

ฟังก์ชัน max() ถูกใช้เพื่อเลือก contour ที่มีพื้นที่มากที่สุดจากลิสต์ bottle\_contours ซึ่งเก็บ contours ที่ตรวจจับได้ในภาพ

-การคำนวณขนาดและตำแหน่งของกรอบสี่เหลี่ยมที่ล้อมรอบ Contour

```
(x_bottle, y_bottle, w_bottle, h_bottle) = cv2.boundingRect(max_bottle_contour)
```

ฟังก์ชัน cv2.boundingRect() คำนวณตำแหน่ง (x, y) และขนาด (width, height) ของกรอบสี่เหลี่ยมที่ล้อมรอบ contour ที่ใหญ่ที่สุด

-การปรับขนาดกรอบสี่เหลี่ยมตามความกว้างของกรอบ

```
if w_bottle > 85:
    adj_w_bottle = int(w_bottle * 0.25)
    adj_h_bottle = int(h_bottle * 2.5)
    new_y_bottle = y_bottle - adj_h_bottle // 2 - 10

elif w_bottle > 50:
    adj_w_bottle = int(w_bottle * 0.25)
    adj_h_bottle = int(h_bottle * 2.9)
    new_y_bottle = y_bottle - adj_h_bottle // 2 - 11

elif w_bottle > 35:
    adj_w_bottle = int(w_bottle * 0.3)
    adj_h_bottle = int(h_bottle * 3.2)
    new_y_bottle = y_bottle - adj_h_bottle // 2 - 8

else:
    adj_w_bottle = int(w_bottle * 0.5)
    adj_h_bottle = int(h_bottle * 3.2)
    new_y_bottle = y_bottle - adj_h_bottle // 2 - 3
```

ฟังก์ชันใช้เงื่อนไขที่แตกต่างกันเพื่อตรวจสอบความกว้างของกรอบ (w\_bottle) และคำนวณค่าปรับขนาดกรอบตามระดับความกว้าง

หากความกว้างมากกว่า 85 พิกเซล:

adj\_w\_bottle คำนวณค่าปรับความกว้างเป็น 27% ของความกว้างเดิม (w\_bottle).

adj\_h\_bottle คำนวณค่าปรับความสูงเป็น 250% ของความสูงเดิม (h\_bottle).

new\_y\_bottle ปรับตำแหน่ง Y ใหม่โดยการลดตำแหน่ง Y ลงเพื่อให้กรอบขยายขึ้น (ลดตำแหน่ง Y) โดยใช้ค่าปรับความสูงและลดค่า 10.

หากความกว้างมากกว่า 50 พิกเซล แต่ไม่เกิน 85 พิกเซล:

adj\_w\_bottle คำนวณค่าปรับความกว้างเป็น 27% ของความกว้างเดิม.

adj\_h\_bottle คำนวณค่าปรับความสูงเป็น 290% ของความสูงเดิม.

new\_y\_bottle ปรับตำแหน่ง Y ใหม่โดยการลดตำแหน่ง Y ลงและลดค่า 11.

หากความกว้างมากกว่า 35 พิกเซล แต่ไม่เกิน 50 พิกเซล:

adj\_w\_bottle คำนวณค่าปรับความกว้างเป็น 28% ของความกว้างเดิม.

adj\_h\_bottle คำนวณค่าปรับความสูงเป็น 340% ของความสูงเดิม.

new\_y\_bottle ปรับตำแหน่ง Y ใหม่โดยการลดตำแหน่ง Y ลงและลดค่า 8.

หากความกว้างไม่เกิน 35 พิกเซล:

adj\_w\_bottle คำนวณค่าปรับความกว้างเป็น 47% ของความกว้างเดิม.

adj\_h\_bottle คำนวณค่าปรับความสูงเป็น 340% ของความสูงเดิม.

new\_y\_bottle ปรับตำแหน่ง Y ใหม่โดยการลดตำแหน่ง Y ลงและลดค่า 3.

-การคำนวณพิกัดและขนาดใหม่ของกรอบ

```
new_x_bottle = x_bottle - adj_w_bottle // 2
```

```
new_w_bottle = w_bottle + adj_w_bottle
```

```
new_h_bottle = h_bottle + adj_h_bottle
```

-พิกัด x ใหม่ (new\_x\_bottle):

คำนวณตำแหน่ง X ใหม่โดยการขยายกรอบไปทางซ้าย ( $x\_bottle - adj\_w\_bottle // 2$ ).

-ความกว้างใหม่ (new\_w\_bottle):

ความกว้างของกรอบใหม่จะเป็นความกว้างเดิมบวกกับค่าปรับความกว้าง ( $w\_bottle + adj\_w\_bottle$ ).

-ความสูงใหม่ (new\_h\_bottle):

ความสูงของกรอบใหม่จะเป็นความสูงเดิมบวกกับค่าปรับความสูง ( $h\_bottle + adj\_h\_bottle$ ).

-การวาดกรอบสี่เหลี่ยมบนภาพ

```
cv2.rectangle(image, (new_x_bottle, new_y_bottle), (new_x_bottle + new_w_bottle, new_y_bottle + new_h_bottle),
(0, 165, 255), 2)
```

วาดกรอบสี่เหลี่ยมที่ล้อมรอบขวดในภาพ โดยใช้สีส้ม (สีรหัส (0, 165, 255)) และความหนาของเส้น 2 พิกเซล

-การใส่ข้อความ "Bottle" บนภาพ

```
cv2.putText(image, f" Bottle ", (x_bottle, y_bottle-50), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (250, 250, 250), 1)
```

ใส่ข้อความ "Bottle" บนภาพใกล้กับตำแหน่งของกรอบ โดยใช้สีขาวและขนาดตัวอักษร 0.4

ใน loop หลัก while True:

4.รับรูป

```
image = ep_camera.read_cv2_image(strategy="newest", timeout=0.5)
```

```
hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
```



ภาพแสดง ภาพของไม้และขวดน้ำที่แปลงเป็น hsv

HSV ถูกใช้งานเพื่อแยกแยะสีได้อย่างชัดเจนกว่าระบบสี RGB เพราะใน RGB สีถูกกำหนดจากค่าของ Red, Green, และ Blue ที่ผสมกัน แต่ใน HSV สีถูกแยกออกจากกันอย่างชัดเจน และสามารถปรับความอิ่มตัวหรือความสว่างของสีได้ง่ายขึ้น ดังนั้น การแปลงเป็น hsv จะดึงค่าจะเห็นส่วนของไก่ และ ฉลากขวดน้ำชัดเจน

5.กำหนด range ของสีไก่ และ ขวดน้ำ

```
lower_chick = np.array([33, 150, 100])
```

```
upper_chick = np.array([36, 255, 255])
```

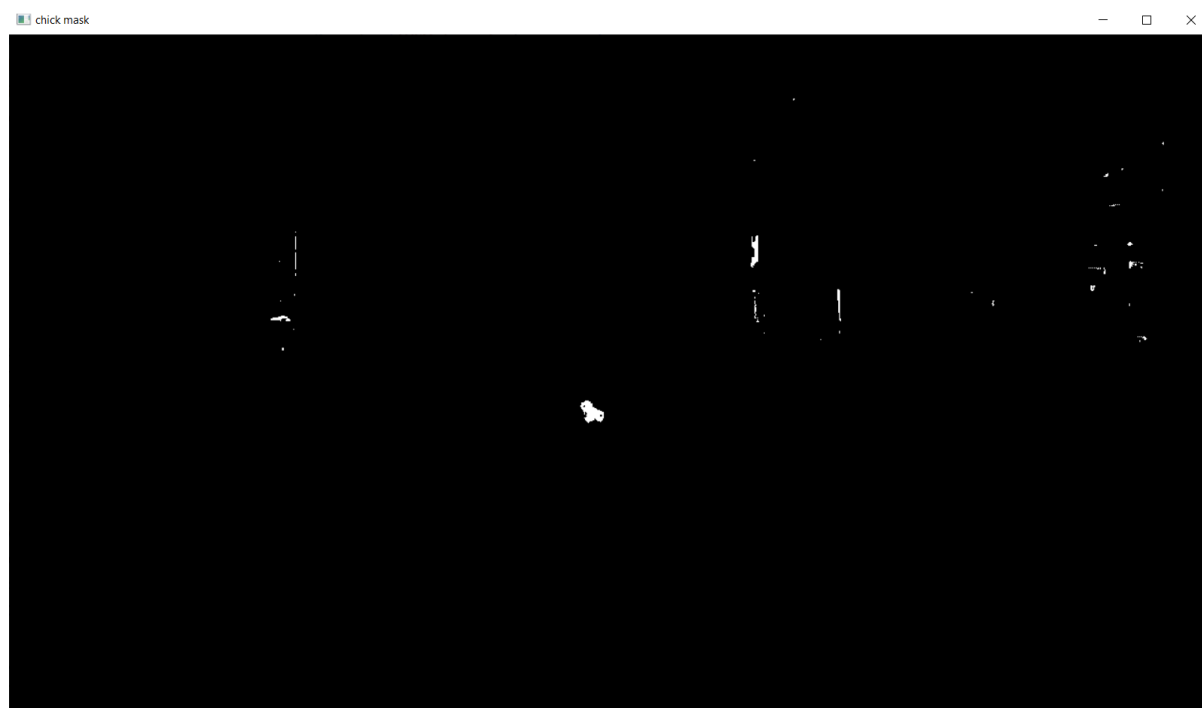
```
lower_bottle = np.array([95, 80, 100])
```

```
upper_bottle = np.array([120, 255, 255])
```

6.สร้าง mask

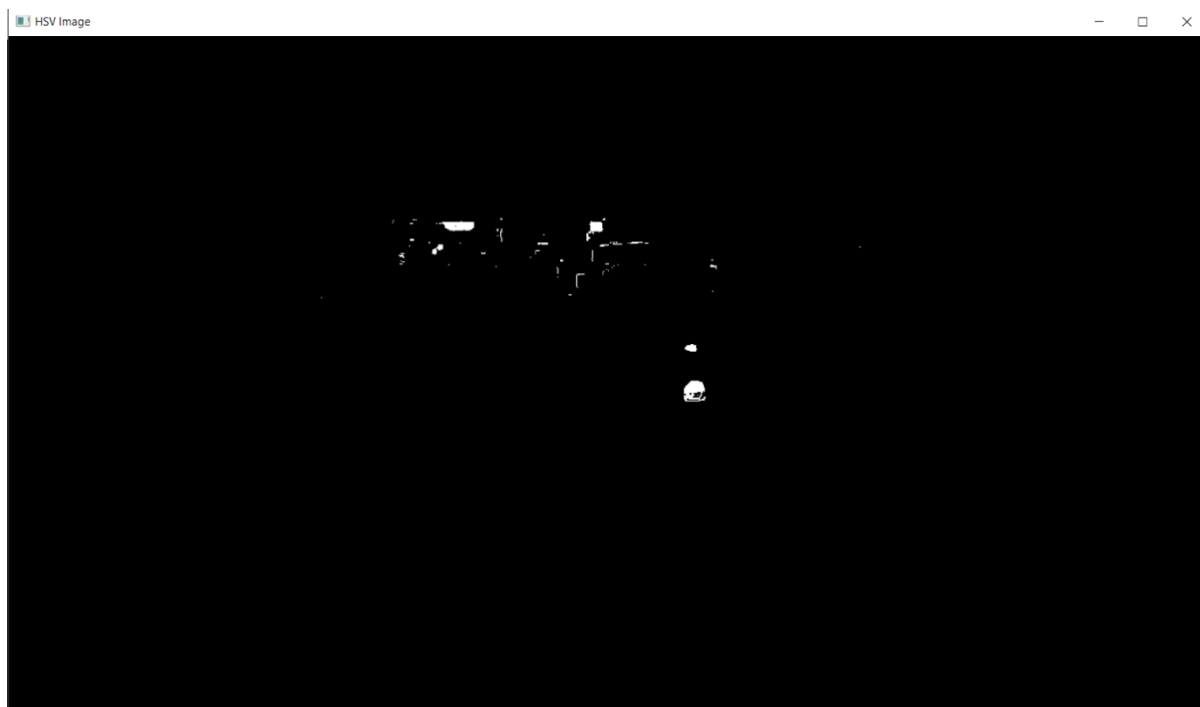
```
mask_chick = cv2.inRange(hsv_image, lower_chick, upper_chick)
```

```
mask_bottle = cv2.inRange(hsv_image, lower_bottle, upper_bottle)
```



ภาพ แสดงการทำ mask ของลูกไก่

โดยจะทำให้ช่วงที่เป็นสีเหลืองเป็น สีขาว และให้ที่เหลือเป็นสีดำ



ภาพ แสดงการทำ mask ของขวดน้ำ

โดยจะทำให้ช่วงที่เป็นสีน้ำเงินเป็น สีขาว และให้ที่เหลือเป็นสีดำ

7. ค้นหา Contour สำหรับไก่และขวด

```
chick_contours, _ = cv2.findContours(mask_chick.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
bottle_contours, _ = cv2.findContours(mask_bottle.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

คำสั่ง `cv2.findContours()` ทำการค้นหาและดึงเอา contour (เส้นรอบขอบวัตถุ) จากภาพในรูปแบบของ mask

mask\_chick: ภาพหรือแมสก์ที่สร้างขึ้นมาสำหรับการตรวจจับไก่

mask\_bottle: ภาพหรือแมสก์ที่สร้างขึ้นมาสำหรับการตรวจจับขวด

ตัวเลือก `cv2.RETR_EXTERNAL` จะดึงเอาเฉพาะ contour ด้านนอกสุดของวัตถุ

ตัวเลือก `cv2.CHAIN_APPROX_SIMPLE` จะลดจำนวนจุดใน contour เพื่อประหยัดพื้นที่ในการเก็บข้อมูล

ผลลัพธ์จะเป็นลิสต์ `chick_contours` สำหรับไก่ และ `bottle_contours` สำหรับขวด ซึ่งเก็บข้อมูล contour ของวัตถุที่ตรวจพบ

8.การตรวจสอบว่าเจอ Contour หรือไม่

```
if chick_contours:
```

```
    detect_chicken()
```

```
if bottle_contours:
```

```
    detect_bottle()
```

โค้ดนี้ทำการค้นหา contour ของวัตถุสองประเภทในภาพ คือ "ไก่" และ "ขวด" โดยใช้ฟังก์ชัน cv2.findContours() หากพบ contour ของวัตถุใดๆ ในภาพ จะทำการเรียกใช้ฟังก์ชัน detect\_chicken() หรือ detect\_bottle() เพื่อวาดกรอบและระบุตำแหน่งของวัตถุในภาพ

9. Code ส่วนเดินรถ

```
x = 0.0
```

```
target_distance = 1.2 # เป้าหมายระยะทางที่ต้องการ
```

```
if target_distance - x > 0:
```

```
    speed = 30 # ความเร็วที่ตั้งไว้ล่วงหน้า
```

```
    ep_chassis.drive_wheels(w1=speed, w2=speed, w3=speed, w4=speed)
```

```
else:
```

```
    ep_chassis.drive_wheels(w1=0, w2=0, w3=0, w4=0)
```

โดยจะให้รถเดินไปด้วยความเร็ว 30 ตอนที่ ค่า target - ตำแหน่งปัจจุบัน มากกว่า0 แล้ว เมื่อน้อยกว่า0 ให้หยุดรถ

10.แสดงผลลัพธ์

```
cv2.imshow("Original image", image)
```

แสดงผลลัพธ์ออกมา ที่มีรูปพร้อมกรอบที่ detect ลูกไก่และขวดน้ำไว้

11.function ไว้สั่งหยุด

```
if cv2.waitKey(1) & 0xFF == ord("q"):
```

```
    break
```

กด q เพื่อหยุดการทำงาน

## 12.ปิดการทำงาน

```
cv2.destroyAllWindows()
```

```
ep_camera.stop_video_stream()
```

```
ep_chassis.unsub_position()
```

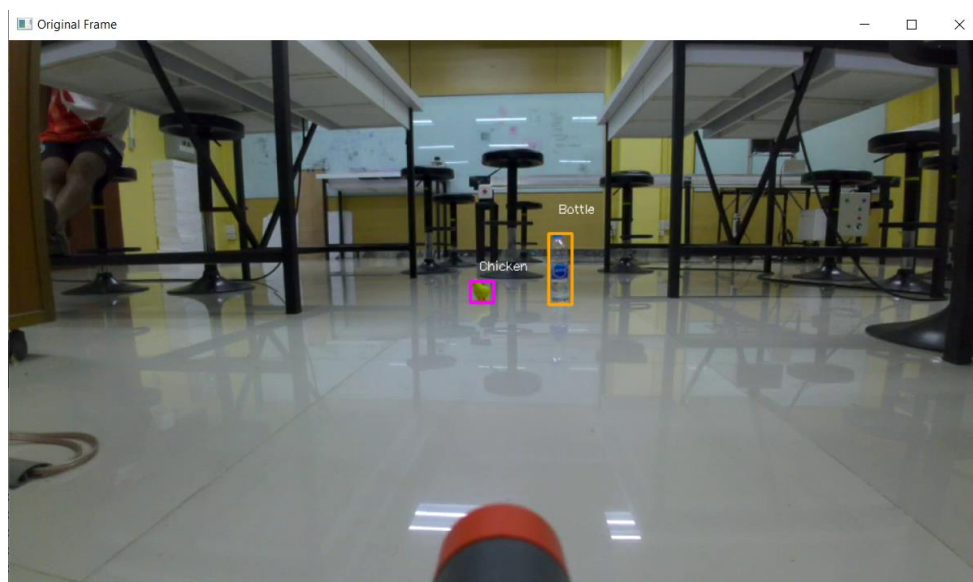
```
ep_robot.close()
```

โค้ดนี้ทำหน้าที่ในการปิดการทำงานและเคลียร์ทรัพยากรที่เกี่ยวข้องกับการประมวลผลวิดีโอและหุ่นยนต์ เมื่อการทำงานเสร็จสิ้นแล้ว

ผลลัพธ์ที่ได้

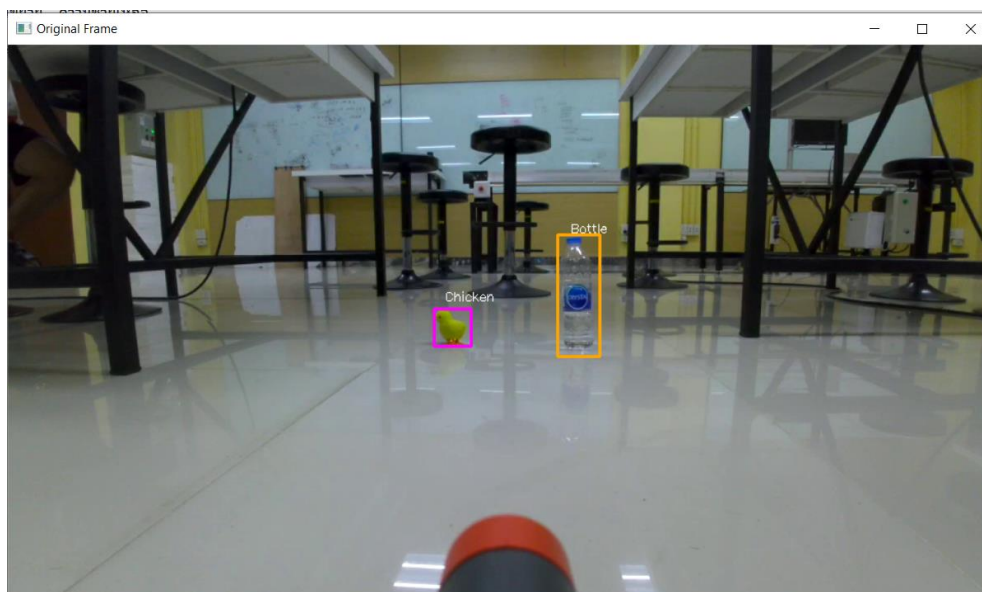
A	B	C	D
วัตถุ			หุ่นยนต์

ระยะ D



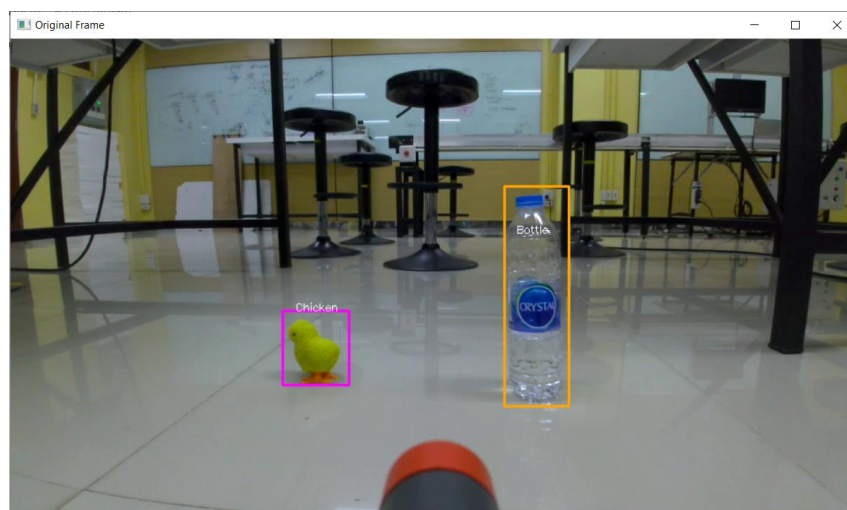
รูปแสดง การ detect วัตถุในระยะ 3กระเบื้อง(ระยะ D)

ระยะ C



รูปแสดง การ detect วัตถุในระยะ 2กระเบื้อง(ระยะ C)

ระยะ B



รูปแสดง การ detect วัตถุในระยะ 1กระเบื้อง(ระยะ B)



หน้าที่ของสมาชิก

- 1.การหาช่วงสี : 6610110475 ณัฐดนัย ชูกุล (เกมส์)
- 2.การตรวจจับวัตถุ : 6610110214 พีรณัฐ ปถมกุล (ก๊ส)
- 3.การปรับขนาด bounding box ลูกไก่ : 6610110126 ธัญพิสิษฐ์ แสงส่อง (ปรีณ)
- 4.การปรับขนาด bounding box ขวดน้ำ : 6610110492 ทักษลักษณ์ แย้มมาสุจริต (อุ๋)