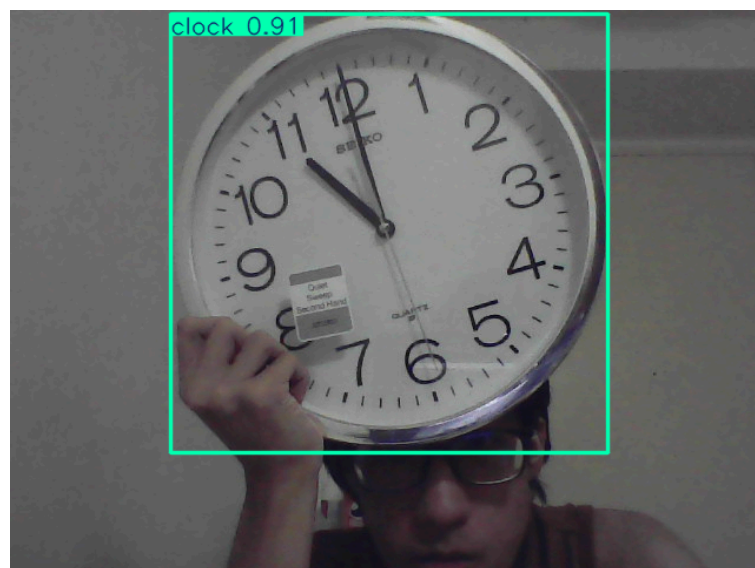


โปรแกรมเพื่ออ่านเวลาจากนาฬิกาเข็ม ส่วนใหญ่ algorithm ที่ใช้จะทุ่มเทให้กับการอ่านเวลาจากภาพนิ่ง เนื่องจากการที่เราจะใช้ 1 algorithm ในการตีเทคทุกสถานการณ์นั้นเป็นไปได้ยาก แต่โปรแกรมสามารถใช้ webcam หรือ video ได้

เราจะอธิบายโดยแบ่งเป็นขั้นตอน ดังนี้

1. ตรวจหานาฬิกาจากภาพ

ในขั้นตอนนี้เราจะใช้ yolov8n ซึ่งเป็น deep learning เพื่อช่วยในการหา นาฬิกาที่แม่นยำ



2. รีสเกลใหม่เพื่อให้ง่ายในการเขียนโปรแกรม

ใช้ฟังก์ชัน `cv2.resize` เพื่อรีสเกลของรูปให้อยู่ในขนาดสี่เหลี่ยมจัตุรัส

3. ตรวจหาเส้นภาพในรูปที่รีสเกลมาแล้ว

ในขั้นตอนนี้เริ่มมีการใช้ฟังก์ชันต่างๆของ opencv และมี step ตามนี้

1. แปลงภาพให้อยู่ในรูปแบบของ gray scale เพื่อลดสีที่จะมารบกวนในการ detect วัตถุอื่นๆ และ ใช้เพื่อจะใช้ในการกำจัด noise ในขั้นตอนถัดไป
2. ใช้ gaussian blur ทำการลด noise ลง
3. ใช้ canny edge detection เพื่อ หาขอบของวัตถุ

4. ใช้ hough line เพื่อ หาเส้นในกรอบของนาฬิกาในที่นี้คือพยายามหาเข็ม โดยปรับ minLineLenght=10 จะได้ไม่ detect เส้นที่อาจจะไม่ใช่เข็มหรือไม่มีความสำคัญในการคำนวณเวลา

```
gray = cv2.cvtColor(clock, cv2.COLOR_RGB2GRAY)
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
edges = cv2.Canny(blurred, 50, 150)
lines = cv2.HoughLinesP(edges, 1, np.pi / 180, 100, minLineLength=10, maxLineGap=20)
```

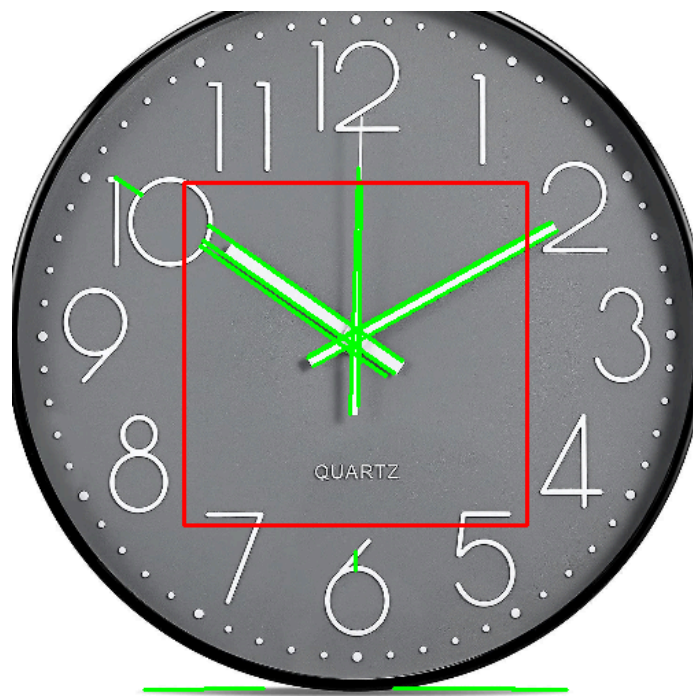


line detection

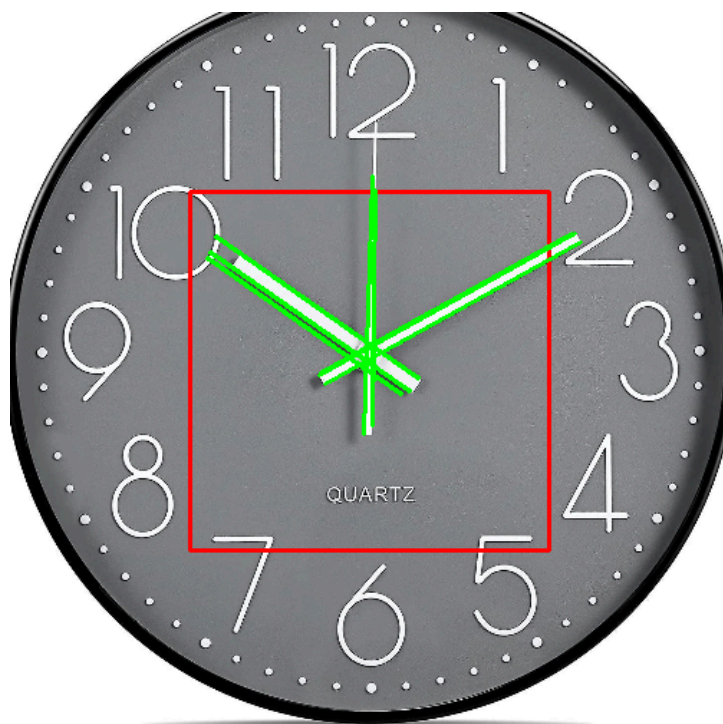
4. หาเข็มจากเส้นที่ทำการ detect มา

ในหัวข้อที่แล้วเราได้เส้นที่ทำการ detect มาจะเห็นได้ว่ามีเส้นที่ไม่ใช่เส้นของนาฬิกาอยู่ในหัวข้อนีเราจะทำการตัดเส้นพวกนั้นออก โดยการหาว่าเส้นในอยู่ในขอบเขตที่เราต้องการหรือไม่ในที่นี้กำหนดให้ขอบเขตมีขนาด 50%ของขนาดรูป

```
hands = []
width_point = np.array([center_x - center_x*t, center_x + center_x*t])
height_point = np.array([center_y - center_y*t, center_y + center_y*t])
if lines is not None:
    for line in list(lines):
        x1, y1, x2, y2 = list(line[0])
        if (width_point[0] < x1 < width_point[1] and height_point[0] < y1 < height_point[1]):
            hands.append((x1, y1, x2, y2))
        elif (width_point[0] < x2 < width_point[1] and height_point[0] < y2 < height_point[1]):
            hands.append((x1, y1, x2, y2))
```



ภาพก่อนตัดเส้นนอกสีเหลี่ยม



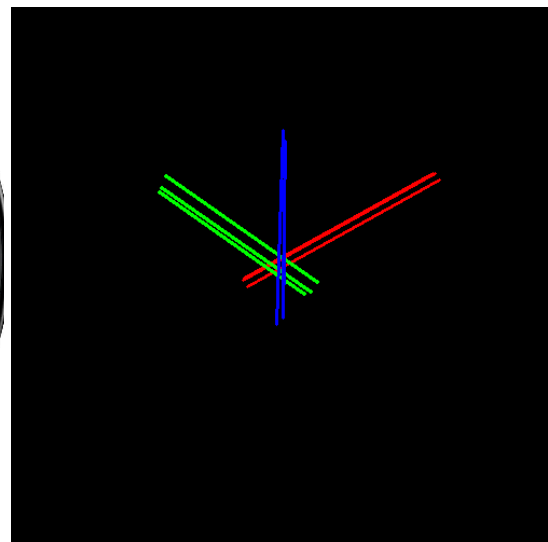
ภาพหลังตัดเส้นนอกสีเหลี่ยมออก

5. จำแนกด้วย kmean

แต่ก่อนที่เราจะแยกเข็มเราต้องทำให้มันใจก่อนว่า เส้นของเราจะอยู่ใน ปริภูมิขั้ว เพื่อคำนวณมุมที่ถูกต้องเพื่อใช้เวลาที่แม่นยำ เราจะกำหนดให้ เส้นที่อยู่ในกรอบสี่เหลี่ยมใกล้เคียงจุดศูนย์กลางที่สุดเป็นจุดศูนย์กลางของ ปริภูมิขั้ว



จากนั้นใช้ kmean ในการจำแนกเข็ม



6. หาว่ากลุ่มไหนที่ได้จาก kmean เป็น เข็มอะไร

ตอนนี้เราจะแบ่งวิธีการหาออกเป็น 2 ขั้นตอนคือ

1. หาความหนาของเส้นเพื่อหาเข็มวินาที โดย

1. **คำนวณเวกเตอร์ของเส้นทั้งสอง:**

- vector1 คือเวกเตอร์ของเส้นแรก (line1)
- vector2 คือเวกเตอร์ของเส้นที่สอง (line2)

2. **คำนวณเวกเตอร์ระหว่างจุดเริ่มต้นของเส้นทั้งสอง:**

- คือเวกเตอร์ที่เชื่อมจากจุดเริ่มต้นของเส้นที่หนึ่งไปยังจุดเริ่มต้นของเส้นที่สอง

3. **คำนวณระยะทาง:**

- ใช้ **cross product** ของ vector1 กับ **เวกเตอร์ระหว่างจุดเริ่มต้นของเส้นทั้งสอง** เพื่อหา "ระยะตั้งฉาก" ระหว่างเส้นตรง
- หาค่าความยาวของ vector1 เพื่อหาค่า **ระยะทางตั้งฉาก** ที่แท้จริง

2. หาเปรียบเทียบความยาวของเส้นเพื่อหาเข็มชั่วโมงและนาที



7. คำนวนเวลาจากองศาของเข็ม

คำนวณจากการหมุนปฏิกิริยาของเข็มไป 90 องศาทวนเข็มนาฬิกาแล้วค่อย
พลิกปฏิกิริยา โดยการนำมุมลบ 180 เพื่อให้ปฏิกิริยาเปลี่ยนเป็นทิศตามเข็มนาฬิกา