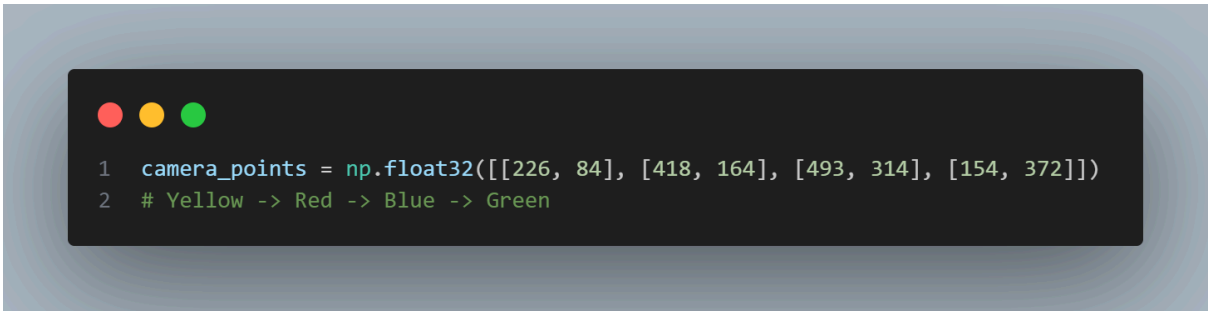
โค้ดที่ 1: Calibration Tool (เครื่องมือสอบเทียบ)
วัตถุประสงค์: ใช้สำหรับปรับแต่งค่าและหาจุดอ้างอิง (Calibration)

การทำงาน

1. ปรับค่า HSV เพื่อตรวจจับสีของบล็อกแต่ละสี (เหลือง, แดง, น้ำเงิน, เขียว)
2. ปรับค่า Distortion เพื่อแก้ไขความผิดเพี้ยนของเลนส์กล้อง
3. แสดงตำแหน่งของบล็อก ทั้งในพิกัดกล้อง (pixel) และพิกัดหุ่นยนต์ (mm)
4. กด P เพื่อพิมพ์พิกัดกล้องของบล็อกทั้งหมดในรูปแบบ



5. กด S เพื่อบันทึกค่า hsv ลง hsv_config.json

คีย์ควบคุม

- S = บันทึกค่าที่ปรับ
- R = โหลดค่าจาก hsv_config.json
- P = พิมพ์พิกัดกล้องของบล็อกที่ตรวจพบ
- [= แสดง mask รวมทุกสี
-] = สลับแสดง mask ทีละสี
- Q/ESC = ออกจากโปรแกรม

โค้ดที่ 2: Production System (ระบบทำงานจริง)
วัตถุประสงค์: ใช้สำหรับทำงานจริงกับแขนกล Dobot MG400

การทำงาน

1. อ่านค่า HSV จาก hsv_config.json ที่ได้จากโค้ดที่ 1
2. ใช้ Perspective Matrix ที่กำหนดไว้ เพื่อแปลงพิกัดกล้องเป็นพิกัดหุ่นยนต์
3. ติดต่อกับแขนกลผ่าน TCP/IP Socket (192.168.1.6:6601)

4. ทำงานแบบ Multi-threading
    - Vision Thread: ประมวลผลภาพและตรวจจับบล็อก
    - Mg400 Thread: รับคำสั่งจากหุ่นยนต์และส่งพิกัดกลับไป

การสื่อสารกับแขนกล

- แขนกล → Python: "pos?" (ถามตำแหน่งบล็อก)
- Python → แขนกล: "338.48,60.60,45.0" (ส่งพิกัด X, Y, มุมหมุน)
- แขนกล → Python: "start" (เริ่มตรวจหา)
- Python → แขนกล: "found" (พบบล็อกแล้ว)

คีย์ควบคุม

- Trackbar "Start Mg400" = เปิดการสื่อสารกับแขนกล
- Trackbar "Enable [สี]" = เลือกสีที่จะตรวจจับ
- S / R / Q = เหมือนในโค้ดที่ 1

ความสัมพันธ์ระหว่างโค้ดทั้งสอง

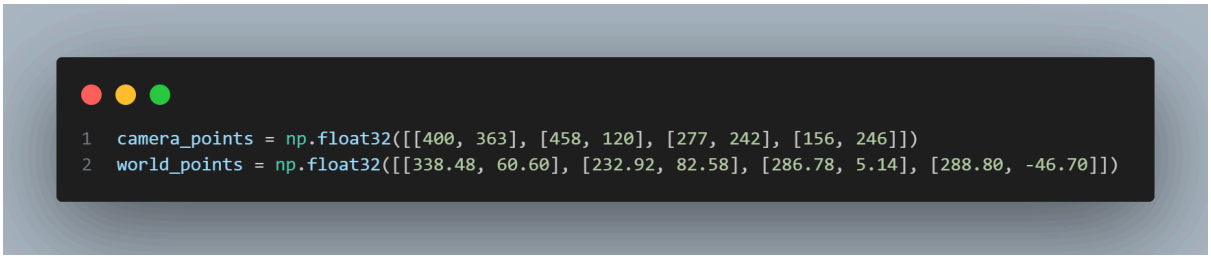
โค้ดที่ 1: Calibration Tool (ปรับค่า HSV, Distortion, หาพิกัดอ้างอิง)
↓
สร้างไฟล์ hsv_config.json
↓
โค้ดที่ 2: Production System (ทำงานจริง + สื่อสารกับแขนกล MG400)

วิธีการใช้งานทั้งหมด

ขั้นตอนที่ 1: Calibration (ใช้โค้ดที่ 1)

1. วางบล็อกสีทั้ง 4 สีในบริเวณที่แขนกลทำงาน
2. รันโค้ดที่ 1 ด้วยคำสั่ง
   python calibration_tool.py
3. ปรับค่า HSV Trackbars จนตรวจจับบล็อกได้ชัดเจน
4. ปรับค่า Distortion หรือ Morphology หากภาพบิดเบี้ยวหรือมี noise
5. กด P เพื่อดูพิกัดกล้อง (camera_points)

6. วัดพิกัดจริงของบล็อกด้วยไม้บรรทัดหรือเทปวัด (world_points)
7. อัปเดตค่าในโค้ดที่ 2



```
camera_points = np.float32([[400, 363], [458, 120], [277, 242], [156, 246]])
world_points = np.float32([[338.48, 60.60], [232.92, 82.58], [286.78, 5.14], [288.80, -46.70]])
```

8. กด S เพื่อบันทึกค่า hsv_config.json

ขั้นตอนที่ 2: ทำงานกับแขนกล (ใช้โค้ดที่ 2)

1. เชื่อมต่อแขนกล MG400 ที่ IP 192.168.1.6
2. รันโปรแกรมควบคุมแขนกลให้ส่งคำสั่ง
    - "pos?" เมื่อต้องการรู้ตำแหน่งบล็อก
    - "start" เมื่อต้องการให้เริ่มตรวจหา
3. รันโค้ดที่ 2 ด้วยคำสั่ง
   python production_system.py
4. เลื่อน Trackbar "Start Mg400" เป็น 1 เพื่อเริ่มการสื่อสาร
5. เลือกสีที่ต้องการหยิบด้วย Trackbar "Enable [สี]"
6. แขนกลจะทำงานอัตโนมัติ โดยระบบจะตรวจจับบล็อก ส่งพิกัด และควบคุมการหยิบ

การไหลของข้อมูล (Data Flow)

กล้อง → อ่านภาพ → ตรวจจับสีด้วย HSV → หา Contour → พิกัดกล้อง (pixel)
→ แปลงเป็นพิกัดหุ่นยนต์ (mm, องศา) → ส่งผ่าน Socket → แขนกล MG400 หยิบบล็อก

สรุป
โค้ดที่ 1 ใช้สำหรับปรับแต่งค่า HSV และหาค่าการสอบเทียบ (Calibration) ครั้งแรกหรือเมื่อเปลี่ยนตำแหน่งกล้อง
โค้ดที่ 2 ใช้สำหรับทำงานจริง ร่วมกับแขนกล MG400 เพื่อให้แขนกลตรวจจับและหยิบบล็อกอัตโนมัติ
ก่อนใช้งานจริง ต้องรันโค้ดที่ 1 เพื่อสร้างไฟล์ hsv_config.json และหาค่า Calibration Matrix เสมอ จากนั้นจึงใช้โค้ดที่ 2 ในการทำงานจริง