ชื่อ-สกุล:
บันทึกไฟล์เป็น Name LastName-Day01.pdf เช่น Wichai Srisuruk-Day01.pdf ส่งงาน ก่อนเลิกเรียน ก่อน 20260822-1700 ที่ https://forms.gle/qkEVhifgsX682nzc8 ส่งงาน รอบสมบูรณ์ ก่อน 20260828-0600 ที่ https://forms.gle/qkEVhifgsX682nzc8

หัวข้อที่ 1: พื้นฐาน Machine Vision และการเตรียมภาพ

Q11: เขียนโปรแกรมอ่านภาพชิ้นงานจากกล้อง แสดงค่าความละเอียด (Resolution) และจำนวนช่องสัญญาณสี

Capture ผลการทำงาน
Capture ผลการทำงาน
คำอธิบายผลการทำงาน

Q12: ทดลองตัด ROI เฉพาะบริเวณที่ต้องการตรวจสอบ แล้วบันทึกเป็นไฟล์ภาพแยกต่างหาก

Capture ผลการทำงาน
Capture ผลการทำงาน
คำอธิบายผลการทำงาน

Q13: เปรียบเทียบภาพเดียวกันที่ถ่ายภายใต้ไฟส่องสว่างต่างชนิดกัน (Direct vs Diffuse) แล้วอภิปรายผลต่างของคุณภาพภาพ

คำอธิบาย

Q14: ให้บอกแนวทางการใช้งาน Machine Vision กับงานที่ผู้เรียนเกี่ยวข้องหรือสนใจ

คำตอบ

หัวข้อที่ 2: การประมวลผลภาพขั้นพื้นฐาน (OpenCV)

Q21: ถ่ายภาพเหรียญจำนวนหนึ่ง แล้วใช้โปรแกรมนับจำนวนวัตถุ โดยปรับค่า param1, param2, minRadius และ maxRadius ของ HoughCircles แล้วสังเกตผลกระทบต่อจำนวนวัตถุที่ตรวจพบ

Capture ผลการทำงาน
Capture ผลการทำงาน
คำอธิบายผลการทำงาน

Q23: จากไปป์ไลน์ตัวอย่าง เปรียบเทียบผมของการทำ Median Blur กับ Gaussian Blur แล้ววิเคราะห์ความแตกต่างของผลลัพธ์การนับวัตถุ

Capture ผลการทำงาน
Capture ผลการทำงาน
คำอธิบายผลการทำงาน

Q22: เลือกภาพจากอินเทอร์เน็ตจำนวน 3 ภาพ (X1, X2, X3) แต่ละภาพให้เปรียบเทียบผลลัพธ์การหาขอบระหว่าง Sobel, Laplacian และ Canny บนภาพชิ้นงานเดียวกัน

Capture ผลการทำงาน
Capture ผลการทำงาน
คำอธิบายผลการทำงาน

Q24: เลือกภาพจากอินเทอร์เน็ตจำนวน 3 ภาพ (Y1, Y2, Y3) แต่ละภาพให้ใช้ HoughLinesP ตรวจหาขอบแนวตรงของชิ้นงานสี่เหลี่ยม แล้วคำนวณความยาวของแต่ละด้านเป็นหน่วยพิกเซล

Capture ผลการทำงาน
Capture ผลการทำงาน
คำอธิบายผลการทำงาน

เอกสารประกอบการอบรม: หลักสูตร Machine Vision และ AI สำหรับงานอุตสาหกรรม (4 วัน)
รายละเอียดการอบรม วันที่ 1
หัวข้อที่ 1: พื้นฐาน Machine Vision และการเตรียมภาพ
หัวข้อที่ 2: การประมวลผลภาพขั้นพื้นฐาน (OpenCV)

หัวข้อที่ 1: พื้นฐาน Machine Vision และการเตรียมภาพ

1.1 บทนำ Machine Vision

Machine Vision (การมองเห็นของเครื่องจักร) คือระบบที่ใช้กล้องถ่ายภาพร่วมกับซอฟต์แวร์ประมวลผลภาพ เพื่อให้เครื่องจักรหรือระบบอัตโนมัติสามารถ "มองเห็น" และวิเคราะห์ข้อมูลจากภาพได้โดยอัตโนมัติ แล้วนำผลลัพธ์ที่ได้ไปใช้ในการตัดสินใจหรือควบคุมกระบวนการผลิตในสายการผลิตอุตสาหกรรม ระบบ Machine Vision แตกต่างจากการประมวลผลภาพ (Image Processing) ทั่วไปตรงที่ต้องทำงานแบบ Real-time ในสภาพแวดล้อมจริง และต้องให้ผลลัพธ์ที่แม่นยำ ทำซ้ำได้ และเชื่อถือได้ในระดับที่นำไปใช้งานตัดสินใจเชิงกระบวนการ (Process Decision) ได้จริง

องค์ประกอบหลักของระบบ Machine Vision ประกอบด้วย 5 ส่วนสำคัญที่ทำงานร่วมกัน ได้แก่ ระบบไฟส่องสว่าง (Illumination) เลนส์ (Optics) กล้อง/เซนเซอร์ภาพ (Camera/Sensor) ฮาร์ดแวร์ประมวลผล (Processing Unit เช่น PC, Smart Camera, Embedded GPU) และซอฟต์แวร์ประมวลผลภาพ (Vision Software/Algorithm) โดยแต่ละส่วนต้องได้รับการออกแบบและเลือกให้เหมาะสมกับลักษณะงานตรวจสอบ เพราะความผิดพลาดที่จุดใดจุดหนึ่งจะส่งผลกระทบต่อความแม่นยำของระบบทั้งหมด

กระบวนการทำงานของระบบ Machine Vision (Vision System Pipeline)

ระบบ Machine Vision ทุกระบบไม่ว่าจะใช้งานเพื่อวัตถุประสงค์ใด จะมีกระบวนการทำงานพื้นฐานเป็นลำดับขั้นตอนที่คล้ายกัน โดยแบ่งออกเป็น 5 ขั้นตอนหลัก ดังตารางต่อไปนี้ ซึ่งเนื้อหาของหลักสูตรในวันแรกนี้จะครอบคลุมตั้งแต่ขั้นตอนที่ 1 ถึง 2 เป็นหลัก ส่วนขั้นตอนที่ 3-5 จะเรียนรู้ในหัวข้อถัดไปของหลักสูตร

ลำดับ	ขั้นตอน	รายละเอียด
1	Image Acquisition (การรับภาพ)	การถ่ายภาพวัตถุด้วยกล้อง เลนส์ และไฟส่องสว่างที่ออกแบบไว้
2	Preprocessing (การเตรียมภาพ)	การปรับปรุงคุณภาพภาพและกรอง Noise ก่อนนำไปวิเคราะห์
3	Segmentation (การแบ่งส่วนภาพ)	การแยกวัตถุที่สนใจออกจากพื้นหลัง (เรียนใน Day-02)
4	Feature Extraction (การสกัดคุณลักษณะ)	การวัดขนาด รูปทรง หรือดึงคุณลักษณะเชิงตัวเลขจากวัตถุ
5	Decision/Output (การตัดสินใจ)	การเปรียบเทียบผลกับเกณฑ์ที่ตั้งไว้ และส่งผลลัพธ์ไปยังระบบควบคุม

จะเห็นได้ว่าคุณภาพของผลลัพธ์ในขั้นตอนท้ายสุดขึ้นอยู่กับคุณภาพของภาพต้นฉบับจากขั้นตอนที่ 1 และ 2 เป็นอย่างมาก หากภาพที่ได้มามีคุณภาพต่ำตั้งแต่ต้น ต่อให้ใช้อัลกอริทึมที่ซับซ้อนเพียงใดในขั้นตอนถัดไปก็ไม่สามารถให้ผลลัพธ์ที่แม่นยำและเชื่อถือได้ ด้วยเหตุนี้เนื้อหาของ Day-01 จึงเน้นการวางรากฐานที่ถูกต้องตั้งแต่การเลือกฮาร์ดแวร์และการเตรียมภาพเป็นสำคัญ

การประยุกต์ใช้งาน Machine Vision ในอุตสาหกรรม

งาน Machine Vision ในสายการผลิตสามารถแบ่งออกเป็น 4 กลุ่มงานหลัก ดังตารางต่อไปนี้

กลุ่มงาน	รายละเอียด	ตัวอย่างการใช้งาน
Inspection (ตรวจสอบคุณภาพ)	ตรวจหาข้อบกพร่อง รอยขีดข่วน ตำหนิบนผิวชิ้นงาน	ตรวจรอยร้าวบนกระจก, ตรวจฉลากสินค้า
Measurement (การวัดขนาด)	วัดระยะ ขนาด มุม พื้นที่ของชิ้นงานโดยไม่สัมผัส	วัดเส้นผ่านศูนย์กลางรูเจาะ, วัดความหนา
Guidance (นำทางหุ่นยนต์)	บอกตำแหน่งและทิศทางของชิ้นงานให้หุ่นยนต์หยิบจับ	Pick and place, การประกอบชิ้นส่วน
Identification (ระบุ/จำแนก)	อ่านรหัส บาร์โค้ด ตัวอักษร หรือจำแนกประเภทชิ้นงาน	อ่าน QR/Barcode, OCR, การคัดแยกสินค้า

1.2 ชนิดและรูปแบบภาพดิจิทัล (Pixel, Channels)

ภาพดิจิทัลประกอบขึ้นจากหน่วยย่อยที่เรียกว่า Pixel (Picture Element) ซึ่งเป็นจุดภาพที่เล็กที่สุดในภาพ แต่ละพิกเซลจะเก็บค่าความสว่างหรือค่าสีในรูปแบบตัวเลข ภาพที่มีความละเอียด (Resolution) สูง เช่น 1920x1080 หมายถึงภาพมีจำนวนพิกเซลตามแนวนอน 1920 จุด และแนวตั้ง 1080 จุด รวมทั้งหมด 2,073,600 พิกเซล ยิ่งจำนวนพิกเซลมาก ภาพยิ่งมีรายละเอียดสูงขึ้น แต่ก็ต้องแลกกับขนาดไฟล์และเวลาในการประมวลผลที่เพิ่มขึ้นตามไปด้วย

ค่าความลึกบิต (Bit Depth) คือจำนวนบิตที่ใช้เก็บค่าความสว่างของแต่ละพิกเซล ภาพขาวดำ (Grayscale) ทั่วไปใช้ 8 บิตต่อพิกเซล ทำให้มีค่าความสว่างได้ตั้งแต่ 0 (ดำสนิท) ถึง 255 (ขาวสนิท) รวม 256 ระดับ ในงานอุตสาหกรรมที่ต้องการความละเอียดของค่าความสว่างสูงกว่าปกติ เช่น การตรวจสอบชิ้นงานโลหะที่มีความต่างของแสงเล็กน้อย อาจใช้กล้องที่ให้ภาพ 10, 12 หรือ 16 บิตต่อพิกเซล

ช่องสัญญาณภาพ (Channels)

ภาพดิจิทัลแบ่งตามจำนวนช่องสัญญาณสีได้ดังนี้

- Grayscale (1 Channel): เก็บเฉพาะค่าความสว่าง ไม่มีข้อมูลสี เหมาะกับงานที่สนใจรูปทรงและความสว่างเป็นหลัก เช่น การวัดขนาดและการตรวจขอบ
- RGB (3 Channels): เก็บค่าความเข้มของแม่สีแดง (Red) เขียว (Green) และน้ำเงิน (Blue) ต่อพิกเซล ใช้แสดงผลภาพสีตามธรรมชาติ
- RGBA (4 Channels): เพิ่มช่อง Alpha สำหรับกำหนดค่าความโปร่งใส นิยมใช้ในงานกราฟิกมากกว่างานตรวจสอบอุตสาหกรรม
- HSV/HSL (3 Channels): แยกค่า Hue (สี) Saturation (ความอิ่มตัวของสี) และ Value/Lightness (ความสว่าง) ออกจากกัน ทำให้การแยกวัตถุตามสีทำได้ง่ายและทนต่อการเปลี่ยนแปลงของแสงมากกว่าระบบ RGB

ในเชิงโครงสร้างข้อมูล ภาพดิจิทัลหนึ่งภาพจะถูกแทนด้วยอาร์เรย์ (Array) หลายมิติ โดยภาพขาวดำแทนด้วยอาร์เรย์ 2 มิติขนาด (สูง x กว้าง) ส่วนภาพสีแทนด้วยอาร์เรย์ 3 มิติขนาด (สูง x กว้าง x จำนวนช่องสี) ซึ่งเป็นรูปแบบที่ไลบรารี NumPy และ OpenCV ใช้ในการจัดเก็บและประมวลผลภาพ

รูปแบบไฟล์ภาพที่ใช้ในงาน Machine Vision

รูปแบบไฟล์	การบีบอัด	ลักษณะเด่น	การใช้งานทั่วไป
BMP	ไม่บีบอัด	ไฟล์ใหญ่ คุณภาพเต็ม ไม่มีการสูญเสียข้อมูล	การประมวลผลระหว่างขั้นตอนภายในระบบ
PNG	Lossless	รองรับ Alpha, บีบอัดไม่สูญเสียข้อมูล	บันทึกผลลัพธ์, ภาพที่ต้องการความแม่นยำ
JPEG	Lossy	ไฟล์เล็ก แต่สูญเสียรายละเอียดบางส่วน	การจัดเก็บภาพจำนวนมาก, ภาพเพื่อรายงาน
TIFF	Lossless/ไม่บีบอัด	รองรับหลาย Bit depth และหลาย Layer	งานวิทยาศาสตร์ กล้องอุตสาหกรรม ความละเอียดสูง
RAW	ไม่บีบอัด (ข้อมูลดิบจากเซนเซอร์)	ข้อมูลตรงจากเซนเซอร์ ยังไม่ผ่าน Demosaic	งานวิเคราะห์เชิงลึกที่ต้องการข้อมูลต้นฉบับ

1.3 พื้นฐานฮาร์ดแวร์: การเลือกกล้อง เลนส์ และระบบไฟส่องสว่าง

ฮาร์ดแวร์เป็นปัจจัยชี้ขาดคุณภาพของระบบ Machine Vision เนื่องจากซอฟต์แวร์ที่ดีที่สุดก็ไม่สามารถแก้ไขภาพที่มีคุณภาพต่ำจากจุดเริ่มต้นได้ หลักการออกแบบระบบที่ถูกต้องคือ "ทำให้ภาพถูกต้องตั้งแต่การถ่าย" (Get it right at the source) ก่อนที่จะนำไปประมวลผลด้วยซอฟต์แวร์

1.3.1 การเลือกกล้อง (Camera Selection)

เซนเซอร์ภาพ (Image Sensor) ที่ใช้ในกล้องอุตสาหกรรมมี 2 เทคโนโลยีหลัก คือ CCD (Charge-Coupled Device) และ CMOS (Complementary Metal-Oxide-Semiconductor) ซึ่งมีคุณสมบัติแตกต่างกันดังตาราง

คุณสมบัติ	CCD	CMOS
คุณภาพภาพ/Noise	ต่ำ คุณภาพสม่ำเสมอสูง	ปรับปรุงดีขึ้นมากในปัจจุบัน ใกล้เคียง CCD
ความเร็วในการอ่านข้อมูล	ช้ากว่า	เร็วกว่า รองรับ Global Shutter ความเร็วสูง
การใช้พลังงาน	สูงกว่า	ต่ำกว่า
ราคา	สูงกว่าโดยทั่วไป	ถูกกว่าและหาซื้อได้ง่ายกว่า
แนวโน้มการใช้งานปัจจุบัน	งานเฉพาะทางที่ต้องการคุณภาพสูงมาก	นิยมใช้เป็นมาตรฐานในงานอุตสาหกรรมทั่วไป

นอกจากเทคโนโลยีเซนเซอร์แล้ว ยังต้องพิจารณาชนิดของ Shutter ด้วย โดย Global Shutter จะอ่านค่าทุกพิกเซลพร้อมกันในเวลาเดียว เหมาะกับวัตถุที่เคลื่อนที่เร็วบนสายพาน ในขณะที่ Rolling Shutter จะอ่านค่าทีละแถวทำให้เกิดภาพเบี้ยว (Motion Skew) หากวัตถุเคลื่อนที่เร็ว จึงเหมาะกับวัตถุนิ่งหรือเคลื่อนที่ช้าเท่านั้น

ประเภทกล้องตามลักษณะการรับภาพ

- Area Scan Camera: รับภาพเป็นเฟรมทั้งภาพในครั้งเดียวคล้ายกล้องถ่ายรูปทั่วไป เหมาะกับชิ้นงานที่มีขอบเขตชัดเจนและหยุดนิ่งหรือเคลื่อนที่เป็นจังหวะ
- Line Scan Camera: รับภาพทีละแถวแล้วนำมาต่อกันเป็นภาพยาวต่อเนื่อง เหมาะกับวัตถุที่เคลื่อนที่ต่อเนื่องด้วยความเร็วคงที่ เช่น ผ้า แผ่นเหล็กม้วน หรือสายพานลำเลียงความเร็วสูง

ปัจจัยสำคัญในการเลือกกล้อง

ปัจจัยสำคัญในการเลือกกล้องมี 3 ค่าหลักที่ต้องพิจารณาร่วมกัน คือ ความละเอียด (Resolution), อัตราเฟรม (Frame Rate) และอินเทอร์เฟซการเชื่อมต่อ (Interface) โดยความละเอียดที่ต้องการคำนวณจากขนาดวัตถุที่ต้องการตรวจสอบและความละเอียดของข้อบกพร่องที่เล็กที่สุดที่ต้องตรวจจับได้ (ตามหลัก Nyquist ควรมีอย่างน้อย 2 พิกเซลต่อขนาดข้อบกพร่องที่เล็กที่สุด)

จำนวนพิกเซลที่ต้องการ = ขนาดพื้นที่ตรวจสอบ (mm) / ขนาดข้อบกพร่องเล็กสุด (mm) × 2

ตัวอย่างการคำนวณความละเอียดกล้องที่ต้องการ

สมมติต้องการตรวจสอบชิ้นงานที่มีพื้นที่ตรวจสอบกว้าง 100 มิลลิเมตร และต้องการตรวจจับตำหนิที่มีขนาดเล็กที่สุดเพียง 0.2 มิลลิเมตรให้ได้ สามารถคำนวณความละเอียดกล้องที่ต้องการตามแนวแกนนั้นได้ดังนี้

จำนวนพิกเซล = 100 mm / 0.2 mm × 2 = 1,000 พิกเซล

จากผลการคำนวณ กล้องที่เลือกใช้จะต้องมีความละเอียดตามแนวแกนที่พิจารณาอย่างน้อย 1,000 พิกเซล เช่น กล้องความละเอียด 1280x1024 หรือสูงกว่า จึงจะสามารถตรวจจับตำหนิขนาด 0.2 มิลลิเมตรบนพื้นที่ตรวจสอบ 100 มิลลิเมตรได้อย่างน่าเชื่อถือ ทั้งนี้ในทางปฏิบัติควรเผื่อค่าความละเอียดเพิ่มอีกประมาณ 20-30% เพื่อรองรับความคลาดเคลื่อนจากการติดตั้งและความไม่สมบูรณ์ของเลนส์

Smart Camera กับ PC-based Vision System

ในการออกแบบระบบ Machine Vision สามารถเลือกสถาปัตยกรรมฮาร์ดแวร์ประมวลผลได้ 2 แนวทางหลัก ซึ่งมีข้อดีข้อเสียแตกต่างกันดังตาราง

คุณสมบัติ	Smart Camera	PC-based Vision System
สถาปัตยกรรม	กล้อง+ประมวลผลรวมในตัวเดียว	กล้องแยกจากคอมพิวเตอร์ประมวลผล
ความยืดหยุ่นของอัลกอริทึม	จำกัดตามที่ผู้ผลิตกำหนด	สูงมาก ปรับแต่งโค้ดได้อิสระ
พลังการประมวลผล	จำกัด เหมาะงานไม่ซับซ้อน	สูง รองรับ Deep Learning ได้ดี
ขนาดติดตั้งและสายไฟ	กะทัดรัด ติดตั้งง่าย	ต้องมีพื้นที่วาง PC และเดินสาย
ค่าใช้จ่ายเริ่มต้น	ต่ำกว่าสำหรับงานเดี่ยว	สูงกว่า แต่คุ้มค่าเมื่อขยายหลายกล้อง

มาตรฐานอินเทอร์เฟซกล้องอุตสาหกรรม

อินเทอร์เฟซ	ความเร็วโดยประมาณ	ระยะสายสูงสุด	จุดเด่น
USB3 Vision	สูงสุด ~350 MB/s	~5 เมตร (ไม่มี repeater)	ติดตั้งง่าย ราคาประหยัด นิยมในงานทั่วไป
GigE Vision	~125 MB/s ต่อพอร์ต	สูงสุด 100 เมตร	ระยะสายไกล เหมาะกับโรงงานขนาดใหญ่
Camera Link	สูงสุดหลาย GB/s	~10 เมตร	ความเร็วสูงมาก ใช้กับ Line scan ความเร็วสูง
CoaXPress	สูงสุดหลาย GB/s	~40 เมตร (สายโคแอกเชียล)	ความเร็วสูงและระยะไกลพร้อมกัน

1.3.2 เลนส์ (Lens)

เลนส์ทำหน้าที่รวมแสงจากวัตถุให้ตกลงบนเซนเซอร์ภาพอย่างชัดเจน การเลือกเลนส์ที่เหมาะสมต้องพิจารณาความยาวโฟกัส (Focal Length) มุมมองภาพ (Field of View: FOV) ระยะทำงาน (Working Distance: WD) และรูรับแสง (Aperture) ร่วมกัน โดยมีความสัมพันธ์เชิงเรขาคณิตดังสมการประมาณดังนี้

FOV = (ขนาดเซนเซอร์ × ระยะทำงาน) / ความยาวโฟกัส

จากสมการข้างต้น หากต้องการ FOV กว้างขึ้นในระยะทำงานเท่าเดิม จะต้องเลือกเลนส์ที่มีความยาวโฟกัสสั้นลง ในทางกลับกันหากต้องการมองเห็นรายละเอียดของวัตถุขนาดเล็กในพื้นที่แคบ จะต้องเลือกเลนส์ความยาวโฟกัสสูงหรือเลนส์มาโคร (Macro Lens)

มาตรฐานเมาท์เลนส์ (Lens Mount)

ประเภทเมาท์	ระยะ Flange Focal Distance	ลักษณะการใช้งาน
C-Mount	17.526 mm	มาตรฐานที่พบมากที่สุดในกล้องอุตสาหกรรม
CS-Mount	12.526 mm	คล้าย C-mount แต่ระยะสั้นกว่า ต้องใช้เลนส์เฉพาะ
F-Mount	46.5 mm	ใช้กับเซนเซอร์ขนาดใหญ่ ความละเอียดสูงมาก

นอกจากนี้ในงานวัดขนาด (Measurement) ที่ต้องการความแม่นยำสูงและไม่ต้องการความคลาดเคลื่อนจากมุมมองภาพ (Perspective Error) เมื่อวัตถุอยู่ห่างจากระนาบโฟกัสเล็กน้อย นิยมใช้เลนส์ชนิด Telecentric ซึ่งมีคุณสมบัติให้ขนาดภาพคงที่ไม่ขึ้นกับระยะห่างของวัตถุภายในช่วง Depth of Field ที่กำหนด

1.3.3 ระบบไฟส่องสว่าง (Lighting) สำหรับงานอุตสาหกรรม

แสงสว่างถือเป็นตัวแปรที่สำคัญที่สุดตัวหนึ่งของระบบ Machine Vision เนื่องจากเป็นตัวกำหนดคุณภาพของภาพต้นฉบับโดยตรง การออกแบบระบบไฟที่ดีจะช่วยเน้นลักษณะเด่นของวัตถุที่ต้องการตรวจสอบ พร้อมทั้งลดทอนรายละเอียดที่ไม่เกี่ยวข้องหรือก่อให้เกิดสัญญาณรบกวน (Noise) ในภาพ

ชนิดไฟส่องสว่าง	หลักการทำงาน	การใช้งานที่เหมาะสม
Backlight (ไฟส่องด้านหลัง)	วางไฟด้านหลังวัตถุ ให้เห็นเป็นเงาดำ (Silhouette)	การวัดขนาด รูปทรงภายนอก ตรวจรูรั่ว
Direct/Ring Light	ส่องตรงจากด้านหน้า รอบเลนส์	งานทั่วไป ตรวจสอบพื้นผิวเรียบ
Diffuse Dome Light	แสงกระจายรอบทิศทางเป็นโดม ลดเงาและแสงสะท้อน	วัตถุผิวมันวาว โค้ง หรือสะท้อนแสงสูง
Dark Field (มุมเฉียง)	ส่องแสงมุมต่ำเฉียงกับผิวชิ้นงาน	เน้นรอยขีดข่วน รอยนูน ตำหนิบนผิวเรียบ
Structured/Line Light	ฉายแสงเป็นเส้นหรือลวดลายบนวัตถุ	การวัดความสูง โปรไฟล์ 3 มิติ
Coaxial Light	ส่องผ่านกระจกกึ่งสะท้อนในแนวแกนเดียวกับเลนส์	ผิวสะท้อนแสงสูง เช่น โลหะขัดเงา

แหล่งกำเนิดแสงที่นิยมใช้ในปัจจุบันคือหลอด LED เนื่องจากมีอายุการใช้งานยาวนาน ให้ความเข้มแสงคงที่ ควบคุมการเปิด-ปิดได้รวดเร็วในระดับไมโครวินาที ประหยัดพลังงาน และสามารถเลือกความยาวคลื่นแสง (สีของแสง) ให้เหมาะกับวัตถุแต่ละประเภทได้ เช่น แสงสีแดงช่วยเพิ่มความคมชัดของภาพขาวดำ หรือแสง UV ใช้ตรวจจับสารเรืองแสงบนบรรจุภัณฑ์

ข้อควรพิจารณาด้านสภาพแวดล้อมการติดตั้ง

นอกจากการเลือกชนิดของกล้อง เลนส์ และไฟส่องสว่างแล้ว การติดตั้งระบบ Machine Vision ในสภาพแวดล้อมโรงงานจริงยังต้องคำนึงถึงปัจจัยแวดล้อมต่อไปนี้ร่วมด้วยเสมอ

- แสงรบกวนจากภายนอก (Ambient Light): ควรติดตั้งฉากกำบังแสง (Light Shield/Enclosure) เพื่อป้องกันแสงจากภายนอกเข้ามารบกวนสภาพแสงที่ออกแบบไว้ โดยเฉพาะบริเวณใกล้หน้าต่างหรือใต้แสงแดด
- การสั่นสะเทือน (Vibration): ควรติดตั้งขาจับกล้องและเลนส์ให้มั่นคงแข็งแรง เนื่องจากแม้การสั่นสะเทือนเพียงเล็กน้อยก็อาจทำให้ภาพเบลอ (Motion Blur) ได้ โดยเฉพาะเมื่อใช้ Shutter Speed ต่ำ
- มาตรฐานการป้องกันฝุ่นและน้ำ (IP Rating): ในพื้นที่ที่มีฝุ่น ความชื้น หรือน้ำมันหล่อลื่น ควรเลือกกล้องและอุปกรณ์ที่มีมาตรฐาน IP65 ขึ้นไป หรือติดตั้งภายในกล่องป้องกัน (Enclosure) ที่เหมาะสม
- อุณหภูมิใช้งาน: อุณหภูมิสูงต่อเนื่องอาจทำให้เกิด Noise เพิ่มขึ้นในเซนเซอร์ภาพ และลดอายุการใช้งานของอุปกรณ์อิเล็กทรอนิกส์ จึงควรตรวจสอบช่วงอุณหภูมิใช้งานที่ผู้ผลิตกำหนดเสมอ

1.4 Industrial Camera คืออะไร? ทำไมโรงงานอัตโนมัติถึงขาดเทคโนโลยีนี้ไม่ได้

https://www.wjtechnology.com/BlogDetail/industrial-camera-machine-vision

ในยุคที่การผลิตต้อง เร็ว แม่นยำ และได้มาตรฐาน การตรวจสอบด้วยสายตามนุษย์เพียงอย่างเดียวอาจไม่เพียงพออีกต่อไป โรงงานจำนวนมากจึงหันมาใช้ระบบ “Machine Vision” หรือระบบการมองเห็นของเครื่องจักร เพื่อยกระดับกระบวนการผลิตให้เป็นอัตโนมัติอย่างแท้จริง

หัวใจของระบบนี้คือ กล้องอุตสาหกรรม (Industrial Camera) อุปกรณ์ที่ไม่ได้มีไว้แค่ “ถ่ายภาพ” แต่เป็นเสมือน “ดวงตาอัจฉริยะ” ที่ช่วยให้เครื่องจักรสามารถวิเคราะห์ ตรวจจับ และตัดสินใจได้เองแบบเรียลไทม์ เมื่อเชื่อมต่อกับระบบซอฟต์แวร์

ในบทความนี้ เราจะพาคุณไปรู้จักว่า

- 1) กล้องอุตสาหกรรมคืออะไร
- 2) ต่างจากกล้องทั่วไปอย่างไร
- 3) และทำไมโรงงานที่ต้องการยกระดับคุณภาพและประสิทธิภาพ ถึงขาดเทคโนโลยีนี้ไม่ได้

1. Industrial Camera คืออะไร?

กล้องอุตสาหกรรม (Industrial Camera) คือ กล้องที่ออกแบบมาโดยเฉพาะสำหรับใช้งานในสายการผลิตและงานอุตสาหกรรม แตกต่างจากกล้องทั่วไปที่เน้นความสวยงามของภาพหรือการบันทึกความทรงจำ กล้องอุตสาหกรรมถูกพัฒนาเพื่อใช้งานร่วมกับระบบอัตโนมัติ เช่น

- การตรวจสอบคุณภาพ (Quality Control)
- การควบคุมกระบวนการผลิต (Process Control)
- การเก็บข้อมูลเพื่อวิเคราะห์ในระยะยาว

กล้องเหล่านี้มักมาพร้อมกับ:

- เซ็นเซอร์ความละเอียดสูง
- ความสามารถในการจับภาพในสภาวะแสงน้อย
- การเชื่อมต่อกับระบบควบคุม เช่น PLC (Programmable Logic Controller), SCADA (Supervisory Control and Data Acquisition)

ที่สำคัญคือสามารถ ส่งข้อมูลไปยังระบบประมวลผลได้แบบเรียลไทม์ เพื่อให้กระบวนการผลิตตอบสนองอัตโนมัติทันที

Industrial Camera VS. Consumer Camera
Source: www.deltaww.com

2. เทียบให้ชัด กล้องอุตสาหกรรม vs. กล้องทั่วไป ต่างอย่างไร

หัวข้อเปรียบเทียบ	Industrial Camera (กล้องอุตสาหกรรม)	Consumer Camera (กล้องทั่วไป)
วัตถุประสงค์การใช้งาน	ใช้ในงานอุตสาหกรรม เช่น ตรวจสอบคุณภาพ , ระบบอัตโนมัติ, วิชั่นซิสเต็ม	ใช้ในชีวิตประจำวัน เช่น ถ่ายภาพ, วิดีโอ, งานสื่อสาร
ตัวอย่างกล้อง	Area Scan, Line Scan, Smart Camera, 3D Vision	DSLR, Mirrorless, Compact, Action Camera
ความแม่นยำ	สูงมาก (ระดับไมครอน)	ปานกลางถึงสูง (แต่ไม่แม่นยำพอสำหรับงาน QC)
ความเร็วในการจับภาพ (Frame Rate)	มีตั้งแต่เริ่มต้นถึงสูงมาก (รองรับ 10-1000+ FPS)	ปกติ 30–60 FPS
ประเภท Shutter	มีทั้ง Global และ Rolling Shutter (ขึ้นอยู่กับงาน)	ส่วนใหญ่เป็น Rolling Shutter
เลนส์ (Lens Options)	Standard C-mount, F-mount	ต้องใช้ตาม format ของผู้ผลิตกล้อง
การเชื่อมต่อระบบ Automation	รองรับการ Trigger, เชื่อมกับ PLC, Robot	ไม่รองรับ
อินเทอร์เฟซ (Interface)	GigE, USB3, Camera Link, CoaXPress	USB, HDMI, Wi-Fi, Bluetooth
ความทนทาน	ทนความร้อน, ฝุ่น, การสั่นสะเทือน	ไม่เหมาะกับสภาพแวดล้อมอุตสาหกรรม
ซอฟต์แวร์และการประมวลผล	ใช้ร่วมกับ Vision Software สำหรับงาน QC, Inspection, Measurement	เน้นภาพสวย ใช้กับโปรแกรมแต่งภาพ
การควบคุมภาพ	ทุกรุ่นปรับได้ละเอียดผ่านซอฟต์แวร์หรือ SDK เช่น Exposure, Gain, Trigger	จำกัดผ่านเมนูกล้องหรือปรับได้แค่บางรุ่น
การใช้งาน 24/7	ออกแบบมาสำหรับใช้งานต่อเนื่อง	ไม่แนะนำ เนื่องจากส่วนใหญ่ไม่ได้ออกแบบเพื่อใช้งานต่อเนื่อง
การบำรุงรักษา	ทนทาน บำรุงรักษาน้อย อายุการใช้งานยาว	มีโอกาสเสื่อมจาก Battery, อายุ shutter ที่เป็นม่านกลไก

3. แล้วทำไมโรงงานถึงขาดเทคโนโลยี Machine Vision ไม่ได้?

1. **ลดข้อผิดพลาดจากมนุษย์ได้เกือบ 100%**

   พนักงาน QC ที่ตรวจชิ้นงานตลอดทั้งวัน ย่อมมีโอกาสหลุดพลาดจากความเหนื่อยล้า, ความเคยชิน, หรือแม้แต่ "ความรู้สึกว่าไม่เป็นไร" แต่ระบบ Machine Vision ไม่มีความล้า ไม่มีอคติ และตรวจชิ้นงานทุกชิ้นด้วย เกณฑ์เดียวกันเสมอ

2. **เพิ่มความเร็วในสายการผลิต**

   การตรวจสอบที่เคยต้องใช้คนยืนประจำจุด สามารถตรวจแบบเรียลไทม์ในเวลาไม่กี่มิลลิวินาที/ชิ้น ด้วยการใช้ กล้องอุตสาหกรรมความเร็วสูง (Industrial Camera) ร่วมกับซอฟต์แวร์วิเคราะห์ภาพ

3. **เก็บข้อมูลแบบดิจิทัล ช่วยวิเคราะห์ต้นเหตุของปัญหา**

   ระบบ Machine Vision สามารถบันทึกภาพชิ้นงานที่ผ่าน/ไม่ผ่าน พร้อม log ข้อมูลแบบเรียลไทม์ นำไปสู่ การวิเคราะห์เชิงสถิติ (Quality Trend Analysis) ที่ช่วยปรับปรุงกระบวนการผลิตแบบต่อเนื่อง

4. **คุ้มค่าในระยะยาว (ROI ชัดเจน)**

   โรงงานหลายแห่งที่ลงทุนระบบ Machine Vision พบว่าแม้จะต้องใช้ต้นทุนเริ่มต้นสูงกว่าแรงงาน แต่เมื่อคิดรวมเวลาที่ลดลง, ของเสียที่น้อยลง, และคุณภาพที่สม่ำเสมอขึ้น กลับคืนทุนได้ในเวลาไม่นานและยังลดต้นทุนระยะยาวได้อย่างมหาศาล

กรณีศึกษาจากโรงงาน

- โรงงานผลิตยา – ตรวจสอบฝาและระดับของเหลวในขวดด้วยความเร็ว 400 ขวด/นาที

โรงงานผลิตยาน้ำบรรจุขวด เผชิญปัญหาในการตรวจสอบความสมบูรณ์ของบรรจุภัณฑ์ในไลน์ที่มีความเร็วอยู่ที่ 400 ขวดต่อนาที โดยจะเช็ค Broken Cap, Ring และ Water Level และมีข้อจำกัดของการตรวจด้วยสายตามนุษย์ โรงงานจึงเปลี่ยนมาใช้ Machine Vision System และ Air reject system แทน

ผลลัพธ์จากการใช้งาน

- ตรวจความสมบูรณ์ของบรรจุภัณฑ์ครบตามต้องการด้วยความเร็ว 400 ขวด /นาที
- ลดข้อผิดพลาดจากแรงงานคน และลดแรงงานในส่วนนี้เหลือเพียงผู้ดูแลระบบเพียงคนเดียว
- ระบบมีการบันทึกภาพ NG ทุกชิ้น เพื่อใช้ในการตรวจสอบย้อนกลับและการปรับปรุงกระบวนการ
- เพิ่มความมั่นใจในคุณภาพผลิตภัณฑ์ก่อนส่งออก

สรุปส่งท้าย

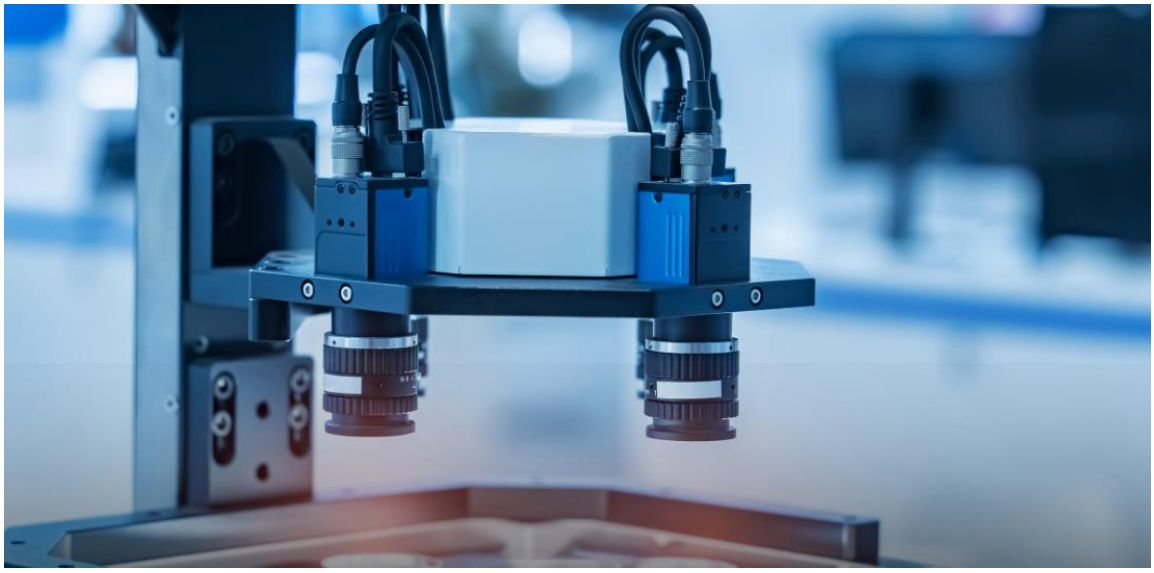
กล้องอุตสาหกรรมไม่ใช่แค่เครื่องมือถ่ายภาพ แต่มันคือ “ดวงตาอัจฉริยะ” ที่โรงงานยุคใหม่ต้องมี ช่วยให้การผลิตแม่นยำ รวดเร็ว ตรวจสอบได้และลดต้นทุนอย่างยั่งยืน

แต่จะเลือก "ดวงตา" ให้ถูกข้างได้อย่างไร? การเลือกกล้องอุตสาหกรรมที่เหมาะสมกับงานไม่ใช่เรื่องง่าย เพราะแต่ละประเภทก็มีจุดเด่นและข้อจำกัดที่ต่างกัน

1.5 เจาะลึก: กล้องอุตสาหกรรม 4 ประเภท และแนวทางการเลือกให้เหมาะกับไลน์ผลิต

https://www.wjtechnology.com/BlogDetail/4Types-of-IndustrialCamera

กล้องไม่ได้มีแค่แบบเดียว และการเลือกกล้องที่ "ถูกตา ถูกงาน" อาจหมายถึงการประหยัดค่าใช้จ่ายได้หลักแสน



ในหัวข้อก่อนหน้า เราได้รู้จักกับกล้องอุตสาหกรรมในฐานะ "ดวงตา" ของระบบอัตโนมัติยุคใหม่ไปแล้ว ซึ่งก็ได้เห็นกันชัดว่าเทคโนโลยี Machine Vision ไม่ใช่แค่ถ่ายภาพ แต่มันคือการ "มองเห็นเพื่อให้เครื่องจักรตัดสินใจ" ในกระบวนการผลิตจริง

- แต่ดวงตานั้นมีหลายแบบ แล้วแบบไหน...ที่ใช่สำหรับงานของคุณ?
- บางโรงงานเลือกกล้องราคาสูงสุด แต่ใช้งานได้ไม่ถึงครึ่งศักยภาพ
- บางโรงงานใช้กล้องราคาถูกเกินไป จนตรวจไม่ทันกระบวนการผลิตจริง
- และบางโรงงาน...ต้องเปลี่ยนระบบใหม่หมด เพราะเลือกกล้องผิดตั้งแต่ต้น
- เพราะกล้องแต่ละประเภท ถูกออกแบบมาให้ตอบโจทย์ใน "ลักษณะงาน" ที่ต่างกัน

ในหัวข้อนี้ เราจะพาไปรู้จักกับกล้องอุตสาหกรรมทั้ง 4 ประเภทหลักที่คนนิยมใช้งาน พร้อมแนะแนวทางการเลือกกล้องให้ตรงงาน เพื่อให้ทุกการลงทุนได้ผลตอบแทนคุ้มค่าที่สุด ทั้งในแง่ของคุณภาพ ความเร็วและกำไรที่มากขึ้น

1) Area Scan Camera – กล้องถ่ายภาพแบบพื้นที่สองมิติ (2D)

Area Scan Camera (Source: www.baslerweb.com)

หากเปรียบกล้องในระบบวิชันเป็นกล้องถ่ายรูป Area Scan Camera ก็เปรียบได้กับกล้อง DSLR ที่เราคุ้นเคย แต่เซ็นเซอร์ของกล้องอุตสาหกรรมจะมีความละเอียดสูงและทนทานกว่า

กล้องชนิดนี้จะจับภาพทั้งเฟรมในครั้งเดียว ให้ภาพออกมาเป็นกรอบสี่เหลี่ยมตามขนาดเซ็นเซอร์ กล้องชนิดนี้ได้รับความนิยมมากที่สุดในงานวิชัน เพราะสามารถใช้งานได้แทบทุกประเภทในสายการผลิต เหมาะกับงานที่มองเห็นวัตถุทั้งหมดในช็อตเดียว เช่น ตรวจรอยตำหนิบนชิ้นส่วน ตรวจฉลากสินค้า หรือตรวจสอบการประกอบว่าครบถ้วนหรือไม่ เป็นต้น โดยมากชิ้นงานจะถูกหยุดนิ่งชั่วขณะ (เคลื่อนที่ช้า) ขณะที่กล้องถ่ายภาพ เพื่อให้ได้ภาพที่คมชัด หากชิ้นงานเคลื่อนที่เร็วมาก อาจใช้การ Trigger ให้กล้องถ่ายภาพเมื่อชิ้นงานอยู่ในตำแหน่งที่กำหนด กล้อง Area Scan มีทั้งรุ่นที่ใช้ Global Shutter (อ่านค่าทุกพิกเซลพร้อมกัน) เหมาะกับชิ้นงานเคลื่อนที่เร็ว ส่วนรุ่นที่ใช้ Rolling Shutter (อ่านค่าไล่ทีละแถว) เหมาะกับชิ้นงานนิ่งหรือเคลื่อนที่ช้ามาก

จุดเด่น

- สามารถเลือกความละเอียดได้ตั้งแต่ไม่กี่แสนพิกเซลจนถึงหลายสิบล้านพิกเซล
- รองรับความเร็วของไลน์ผลิตได้ตั้งแต่ช้าถึงระดับหลายร้อยชิ้นต่อนาที (ความเร็วตั้งแต่ระดับ 10 FPS ถึงระดับ 1000 FPS ในรุ่นกล้องความเร็วสูง)
- ติดตั้งง่าย เพียงกำหนดมุมกล้องให้ครอบคลุมชิ้นงาน
- ใช้แสงไฟส่องพื้นที่ทั้งหมดของภาพในครั้งเดียว ไม่ต้องประสานกับการเคลื่อนที่เหมือนกล้อง Line Scan

ข้อจำกัด

- มุมมองภาพจำกัดตามขนาดเซ็นเซอร์ หากชิ้นงานมีขนาดใหญ่กว่าขอบเขตที่กล้องจะมองเห็น อาจต้องใช้กล้องหลายตัวเรียงต่อกัน ซึ่งทำให้ต้นทุนสูงและการปรับเทียบยุ่งยาก
- หากใช้งานกับชิ้นงานเคลื่อนที่เร็ว ควรใช้รุ่น Global Shutter เพื่อป้องกันภาพเบลอ

เหมาะกับงาน

- ตรวจสอบชิ้นส่วนอิเล็กทรอนิกส์
- ตรวจฉลาก, บาร์โค้ด, QR Code
- ตรวจชิ้นงานบนสายพานแบบแยกชิ้น

2) Line Scan Camera – กล้องสแกนเส้นหนึ่งมิติ (1D)

Line Scan Camera (Source: www.baslerweb.com)

Line Scan Camera ต่างจากกล้องทั่วไปตรงที่มันจะจับภาพได้ครั้งละ 1 แถวของพิกเซล แล้วใช้การเคลื่อนที่ของวัตถุหรือกล้องประกอบกับซอฟต์แวร์เพื่อต่อเส้นภาพเหล่านั้นให้กลายเป็นภาพสองมิติเต็มรูป ซึ่งกล้อง Line Scan จะเหมาะกับชิ้นงานที่มีลักษณะต่อเนื่องหรือยาวมากและต้องการตรวจสอบแบบ 100% ตลอดความยาว เช่น การตรวจสอบผิวกระดาษ ผ้าหรือฟิล์มพลาสติกบนสายการผลิต การตรวจสอบสเกลตลับเมตร เป็นต้น นอกจากนี้ยังนิยมใช้กับงานสแกนรอบชิ้นงานแบบ 360 องศา เช่น การตรวจฉลากกระป๋องรอบด้าน โดยการหมุนวัตถุให้กล้องสแกนครบหนึ่งรอบ

ในการใช้งานกล้อง Line Scan ต้องประสานการทำงานกับการเคลื่อนที่อย่างแม่นยำ มักจะต้องมี Encoder วัดระยะการเคลื่อนที่ของสายพานหรือชิ้นงาน เพื่อให้ซอฟต์แวร์ต่อภาพได้สัดส่วนถูกต้อง ซึ่งหากความเร็วสายพานไม่คงที่ ภาพอาจยืดหรือหดผิดสัดส่วนได้

จุดเด่น

- เก็บภาพได้ละเอียดระดับไมครอนแบบไม่มีขีดจำกัดในแนวยาว (เส้นเดียว)
- ความเร็วในการจับภาพต่อเนื่องสูงหลายพันเส้นต่อวินาที (บางรุ่นเป็นแสนเส้นต่อวินาที) ทำให้รองรับไลน์ผลิตที่มีความเร็วสายพานสูงมาก
- ใช้ไฟส่องเป็นเส้นแคบ ๆ เฉพาะบริเวณที่เซ็นเซอร์อ่านอยู่ ไม่ต้องส่องเต็มพื้นที่ใหญ่ ช่วยลดปัญหาแสงจ้าสะท้อนและประหยัดพลังงานกว่าการส่องทั้งชิ้น (แต่ต้องใช้แสงความเข้มสูงและสม่ำเสมอตลอดแนวสแกน)

ข้อจำกัด

- การติดตั้งซับซ้อนกว่าแบบ Area Scan เนื่องจากต้องออกแบบระบบให้สัมพันธ์กับการเคลื่อนที่ของวัตถุอย่างละเอียด ทั้งเรื่องความเร็ว การกระตุกสั่นของสายพาน การใช้ Encoder และการประมวลต่อภาพแบบเรียลไทม์ หากซิงโครไนซ์ไม่ดีจะได้ภาพที่ผิดเพี้ยน
- ไม่เหมาะกับการถ่ายชิ้นงานหยุดนิ่ง เนื่องจากกล้อง Line Scan จะเห็นเพียงเส้นเดียว ต้องให้ชิ้นงานหรือกล้องขยับ ถึงจะได้ภาพเต็ม
- กล้อง Line Scan มักใช้เลนส์ที่รองรับขนาดเซ็นเซอร์ยาว เช่น F-Mount รวมถึงบางงานต้องใช้ Frame Grabber และพอร์ตเชื่อมต่อความเร็วสูง เช่น Camera Link, CoaXPress เพื่อรองรับแบนด์วิดท์ข้อมูลภาพขนาดใหญ่ ทำให้ต้นทุนต่อจุดสูงกว่ากล้อง Area Scan ในระบบเล็ก ๆ

เหมาะกับงาน

- ตรวจสอบชิ้นงานต่อเนื่องแบบ 100% ความละเอียดสูงและความเร็วสูง เช่น โรงงานตลับเมตรที่ต้องการตรวจสอบความถูกต้องของสเกลตลับเมตร โรงงานกระดาษที่ต้องการตรวจหาจุดตำหนิบนแผ่นกระดาษตลอดความยาว
- ตรวจสอบวัตถุทรงกลมหรือทรงกระบอกแบบ 360° เช่น รอยตำหนิรอบขวดน้ำ ขวดแก้ว หรือท่อ

หลักการทำงาน Area Scan Camera และ Line Scan Camera (Source: https://www.voltrium.com.sg/)

3) Smart Camera – กล้องอัจฉริยะพร้อมระบบประมวลผลในตัว

Smart Camera รุ่น NEON-2000-JNX Series จาก ADLINK (Source: www.adlinktech.com)

Smart Camera หรือ Vision Sensor แบบสมาร์ท เป็นการรวมเอาเซ็นเซอร์กล้อง หน่วยประมวลผล และซอฟต์แวร์ประมวลผลภาพ ไว้ในอุปกรณ์เดียวกัน จึงสามารถทำงาน "วิเคราะห์ ตัดสินใจและส่งผลลัพธ์" ได้โดยไม่ต้องใช้คอมพิวเตอร์ภายนอก เหมาะสำหรับงานตรวจสอบที่ "จบในจุดเดียว" ต้องการความเรียบง่ายในการติดตั้งและใช้งานโดยไม่ต้องเขียนโปรแกรมเอง ซึ่งผู้ใช้งานสามารถปรับแต่งค่าพารามิเตอร์ผ่านโปรแกรมที่ติดมากับกล้องได้เลย เช่น การกำหนดเกณฑ์ตรวจสอบ พื้นที่ที่สนใจ (ROI), ค่าการประมวลผลต่าง ๆ เป็นต้น

ตัวอย่าง Smart Camera ที่เป็นที่รู้จักในอุตสาหกรรม เช่น Cognex In-Sight, Keyence IV Series, ADLINK Neon, Datasensing CS Series เป็นต้น

ลักษณะการใช้งาน เพียงต่อกล้องเข้ากับแหล่งจ่ายไฟและสัญญาณ I/O Network ก็สามารถทำงานได้เลย นิยมติดตั้ง Smart Camera 1 ตัว ต่อ 1 จุดตรวจสอบ ทำงานอิสระของตัวมันเอง โดยจะส่งสัญญาณผลลัพธ์ออกไปยัง PLC หรือระบบควบคุมโดยตรง

จุดเด่น

- ติดตั้งง่าย ใช้งานรวดเร็ว ต่อไฟและสัญญาณ I/O ก็เริ่มทำงานได้เลย
- มีซอฟต์แวร์พร้อมใช้ในตัว ไม่เชี่ยวชาญด้านโปรแกรมวิชันก็เริ่มต้นระบบได้
- มีหน่วยประมวลผลในตัว ไม่ต้องใช้ PC ภายนอก ทำให้ลดจำนวนอุปกรณ์ลง
- ขนาดเล็กกะทัดรัดและทนทาน เหมาะกับโรงงานอุตสาหกรรมโดยเฉพาะ บางรุ่นเป็น IP67 กันฝุ่นกันละอองน้ำ
- Smart Camera รุ่นใหม่ที่รองรับการประมวลผลแบบ AI Deep Learning ช่วยให้สามารถตรวจจับงานที่มีความซับซ้อนได้ เช่น การตรวจจับว่าพนักงานสวมชุดนิรภัยครบถ้วนก่อนเข้าสู่พื้นที่อันตรายหรือไม่

ข้อจำกัด

- ความยืดหยุ่นในการปรับแต่งน้อย ต้องใช้ฟังก์ชันที่ผู้ผลิตมีมาให้ ไม่สามารถเขียนโค้ดเพิ่มเติมได้เหมือนระบบ PC-Based
- สเปคกล้องมีให้เลือกจำกัด ไม่ว่าจะเป็นความละเอียด ความเร็วในการจับภาพ (Frame Rate), ประเภทเซ็นเซอร์มีให้เลือกน้อยกว่ากล้องแบบมาตรฐาน
- ไม่เหมาะกับงานที่ซับซ้อนหรือหลายขั้นตอน เช่น การประมวลผลภาพหลายจุดพร้อมกัน เนื่องจากมีชิปประมวลผลตัวเดียวในกล้อง
- ต้นทุนต่อหนึ่งจุดสูง ซึ่งหากโรงงานต้องการติดตั้งจุดตรวจสอบหลายสิบจุด การใช้ Smart Camera ในทุกจุดจะมีต้นทุนรวมที่สูงกว่าการใช้ระบบวิชันแบบดั้งเดิมที่กล้องหลายตัวต่อกับคอมพิวเตอร์อุตสาหกรรมเครื่องเดียว

เหมาะกับงาน

- งานตรวจสอบที่ค่อนข้างเฉพาะจุดและไม่ซับซ้อนมาก เช่น การตรวจฉลากและบาร์โค้ดบนสายพานบรรจุภัณฑ์ การตรวจสอบว่าฝาขวดมีหรือไม่ การตรวจจับว่าพนักงานสวมชุดนิรภัยครบถ้วนก่อนเข้าสู่พื้นที่อันตรา
- โรงงานที่ใหม่กับระบบ Machine Vision และต้องการทดลองใช้งานระบบนี้ การเริ่มต้นด้วย Smart Camera เป็นทางเลือกที่ลดความเสี่ยงและความยุ่งยากได้ดี เนื่องจากมีทั้งกล้อง ซอฟต์แวร์ และหน่วยประมวลผลรวมอยู่ในอุปกรณ์เดียว พร้อมใช้งานได้ทันทีโดยไม่ต้องเขียนโปรแกรม

4) 3D Vision Camera – กล้องมองภาพสามมิติ

3D Vision Camera (Source: www.baslerweb.com)

3D Camera ไม่ได้มองแค่ภาพสองมิติ แต่สามารถ "วัดระยะลึก" ได้ด้วย ช่วยให้เครื่องจักรมองเห็นวัตถุในแบบที่ใกล้เคียงสายตามนุษย์มากขึ้น เช่น การตรวจความสูงของฝาขวด ตรวจปริมาตรสินค้า และการช่วยให้หุ่นยนต์หยิบชิ้นงานในถังได้อย่างแม่นยำ ซึ่งเทคโนโลยีที่ใช้มีหลายแบบ เช่น Stereo Vision ใช้กล้องสองตัวถ่ายภาพจากมุมที่ต่างกัน แล้วคำนวณความลึกจากภาพคู่, Structured Light การยิงแสงลวดลายแล้ววัดการบิดเบี้ยวของลายบนวัตถุ, Time-of-Flight (ToF) วัดเวลาที่แสงใช้สะท้อนกลับจากวัตถุ เพื่อวัดระยะห่างแบบเรียลไทม์ เป็นต้น

ตัวอย่าง 3D Camera ที่เป็นที่รู้จักในอุตสาหกรรม เช่น Cognex, Keyence, Basler, HIKROBOT, Mech-Mind Robotics, ifm electronic

จุดเด่น

- เห็น "ระดับความลึก" และความแตกต่างเชิงปริมาตร ที่กล้อง 2D มองไม่เห็น
- วัดขนาด 3 มิติได้อย่างแม่นยำ เช่น วัดความสูง ปริมาตร ความเอียง
- เพิ่มความยืดหยุ่นในการตรวจจับ กรณีที่ชิ้นงานวางซ้อนกัน ซึ่งกล้อง 3D สามารถแยกวัตถุจากกันตามระยะใกล้–ไกลได้

ข้อจำกัด

- ราคาสูงกว่ากล้อง 2D ทั่วไป เนื่องจากกล้อง 3D ใช้เทคโนโลยีขั้นสูง ทั้งในด้านเซนเซอร์และระบบประมวลผล ทำให้ต้นทุนต่อหน่วยสูงกว่าอย่างชัดเจน
- ใช้ทรัพยากรประมวลผลสูง โดยเฉพาะหากต้องการประมวลผลแบบเรียลไทม์
- ติดตั้งและใช้งานซับซ้อนกว่า ต้องวางตำแหน่งกล้อง จัดแสง และคาลิเบรตอย่างแม่นยำกว่ากล้อง 2D เพื่อให้ได้ข้อมูลเชิงลึก (Depth) ที่ถูกต้อง ไม่เพี้ยน
- เฟรมเรตต่ำกว่ากล้อง 2D ทำให้ไม่เหมาะกับงานสายพานความเร็วสูง
- อ่อนไหวต่อแสงรบกวน

เหมาะกับงาน

- งานที่ต้องวัดความสูง ความลึกหรือรูปร่าง 3 มิติ เช่น วัดปริมาตรกล่อง วัดความสูงระดับพื้นผิวของชิ้นงาน ตรวจตำหนิบนพื้นผิวไม่เรียบ ตรวจหลุมหรือรอยนูน
- งานหยิบจับด้วยหุ่นยนต์ (Robot Guidance) ช่วยหาตำแหน่งของชิ้นงานในถังหรือบนพาเลต แม้ว่าชิ้นงานจะวางซ้อนกันอยู่
- ชิ้นงานที่มีรูปร่างไม่แน่นอน เช่น ผัก ผลไม้ ใช้แยกประเภทหรือคัดเกรดโดยดูจากรูปร่างจริง

ปัจจัยสำคัญในการเลือกกล้องให้เหมาะกับไลน์ผลิต 6 ข้อ

1. ลักษณะและประเภทของชิ้นงาน

ก่อนจะเลือกกล้อง เราต้องดูก่อนว่า ชิ้นงานมีลักษณะแบบใด

ลักษณะชิ้นงาน	กล้องที่เหมาะสม
ชิ้นงานเดี่ยว ๆ ชิ้นต่อชิ้น เช่น ตรวจชิ้นงานบนสายพานแบบแยกชิ้น, Barcode, QR Code	Area Scan หรือ Smart Camera
ชิ้นงานยาวต่อเนื่องที่ต้องตรวจละเอียด เช่น สเกลตลับเมตร, รอยตำหนิรอบขวด	Line Scan Camera
ชิ้นงานแบบ 3 มิติ วัดความลึกหรือปริมาตร	3D Camera
ตรวจเฉพาะจุดแบบง่าย ๆ เช่น มี/ไม่มี, ครบไม่ครบ	Smart Camera
ชิ้นงานมีขนาดใหญ่ เกินขอบเขตภาพของกล้องเดี่ยว	พิจารณาใช้เลนส์มุมกว้าง, กล้องความละเอียดสูง หรือกล้องหลายตัวทำงานร่วมกัน

2. ความเร็วของไลน์ผลิต

ความเร็วสายพานมีผลโดยตรงต่อการเลือกกล้อง

- ไลน์เร็วมาก เช่น มากกว่า 300 ชิ้น/นาที
    - ต้องใช้กล้องที่มี Frame Rate สูง และใช้ Global Shutter เพื่อป้องกันภาพเบลอ
- ความเร็วต่อเนื่องและตรวจต่อเนื่อง เช่น รอยตำหนิกระดาษม้วน
    - ใช้กล้อง Line Scan จะให้ภาพต่อเนื่องแบบไม่มีรอยต่อ
- ไลน์ความเร็วปานกลาง
    - เลือกใช้กล้องที่มี Frame Rate อย่างน้อย 30–60 FPS

3. ความละเอียดและขนาดพิกเซลที่เหมาะสม

ความละเอียด

บางงานต้อง "เห็นให้ชัดถึงระดับเส้นขน" แต่บางงานแค่รู้ว่า "มีหรือไม่มี" ก็เพียงพอแล้ว การเลือกความละเอียดของกล้องจึงต้องขึ้นอยู่กับ "วัตถุประสงค์ของการตรวจสอบ" ว่าต้องการความละเอียดมากแค่ไหน

- งานที่ต้องการความละเอียดสูงระดับไมครอน เช่น ตรวจรอยขีดข่วนบนแผ่นเวเฟอร์ ตรวจตำหนิบนโลหะ หรือวัดขนาดชิ้นส่วนระดับละเอียด
    - ควรเลือกกล้องที่ความละเอียดสูงตั้งแต่ 5MP – 20MP+ ร่วมกับเลนส์คุณภาพสูง และเซ็นเซอร์ที่มี SNR ดี แต่ต้องระวังว่า ยิ่งกล้องมีความละเอียดสูง เฟรมเรตมักต่ำลง และใช้พลังประมวลผลมากขึ้น
    - เทคนิคที่ช่วยลดโหลดข้อมูลคือ การกำหนด ROI (Region of Interest) เลือกให้กล้องอ่านเฉพาะจุดที่สนใจ เพื่อเพิ่มเฟรมเรต และลดข้อมูลที่ไม่จำเป็น
- งานที่ไม่ต้องการความละเอียดสูง เช่น การตรวจว่าฝาขวดมีหรือไม่, ตรวจนับจำนวนชิ้น, อ่านบาร์โค้ด, ตรวจข้อความ (OCR)
    - ใช้กล้องที่ความละเอียด 1MP – 5MP ก็เพียงพอแล้ว และยังช่วยลดภาระของภาพรวมระบบ

ขนาดพิกเซล

ขนาดพิกเซล (Pixel Size) หมายถึง "ความกว้างของพิกเซลแต่ละจุดในเซ็นเซอร์ของกล้อง" โดยวัดเป็นไมโครเมตร (μm)

- พิกเซลขนาดใหญ่ (เช่น 5–10 μm)
    - รับแสงได้มาก ⟶ ให้ภาพสว่าง ชัด และนอยส์ต่ำ เหมาะกับสภาพแสงน้อย
    - ใช้ในงานที่ไม่เน้นความละเอียดสูงมาก แต่ต้องการความเสถียรของภาพ
- พิกเซลขนาดเล็ก (เช่น 2–3.45 μm)
    - บรรจุพิกเซลได้มากในพื้นที่เซ็นเซอร์เท่าเดิม ⟶ ทำให้กล้องมีความละเอียดสูง
    - เหมาะกับงานที่ต้องการจับรายละเอียดมาก ๆ หรือมีพื้นที่มองเห็น (FOV) กว้าง

4. สภาพแวดล้อมการทำงาน

พิจารณาจากสภาพหน้างานจริงที่กล้องจะถูกติดตั้ง

- หน้างานมีฝุ่นหรือน้ำกระเด็น ควรใช้กล้องที่มีมาตรฐานป้องกันที่ IP67/IP69 หรือเสริม Housing ป้องกัน
- หน้างานอุณหภูมิสูงหรือต่ำมาก ๆ ควรเลือกรุ่นที่ทนช่วงอุณหภูมิกว้าง
- หน้างานมีการสั่นสะเทือน ควรใช้ขาตั้งกล้องและเมาท์เลนส์ที่มั่นคง และหลีกเลี่ยงกล้อง Rolling Shutter

5. การเชื่อมต่อและอินเทอร์เฟซ

เลือกให้เหมาะกับระยะการวางกล้องและความเร็วในการถ่ายโอนข้อมูล

อินเทอร์เฟซ	ความเร็ว	ความยาวสายเคเบิล	การใช้งาน
USB 3.0/3.1	เร็ว	~5 เมตร	การใช้งานทั่วไป, คุ้มค่า
GigE (Gigabit Ethernet)	ปานกลาง	~100 เมตร	การใช้งานในอุตสาหกรรม, ระยะทางไกล
Camera Link	เร็วมาก	~10 เมตร	การใช้งานความเร็วสูง, แบนด์วิดท์สูง
CoaXPress	เร็วมาก	~40 เมตร	การใช้งานระดับสูง (เช่น การตรวจสอบเซมิคอนดักเตอร์)

6. งบประมาณและความคุ้มค่าทางการลงทุน (ROI)

แม้ราคาจะเป็นปัจจัยสำคัญในการตัดสินใจ แต่การเลือก "กล้องที่ถูกที่สุด" ณ จุดซื้อ อาจไม่ใช่ทางเลือกที่คุ้มค่าที่สุดในระยะยาว เช่น การเลือกกล้องราคาประหยัดที่ไม่ทนทาน อาจต้องเปลี่ยนใหม่บ่อยครั้ง หรือหากเลือกรุ่นที่ไม่เหมาะกับลักษณะงาน อาจทำให้การตรวจสอบผิดพลาด ต้องแก้ไขระบบใหม่ทั้งหมด ซึ่งเสียทั้งเงิน เวลา และโอกาส

การประเมินที่ดีควรมองทั้งระบบ ไม่ใช่แค่ราคากล้อง รวมถึงเลนส์, ไฟ, คอมพิวเตอร์, ซอฟต์แวร์, การติดตั้ง และการดูแลระยะยาว แล้วเทียบกับมูลค่าที่จะได้รับ เช่น

- ลดแรงงานคนได้เท่าไหร่
- ลดของเสียได้กี่เปอร์เซ็นต์
- เพิ่มความเร็วในการผลิตได้แค่ไหน

ถ้าคุณตอบคำถามเหล่านี้ได้ชัด ก็จะเห็นเลยว่า ควรลงทุนเท่าไหร่ และสเปคแบบไหนถึงจะคุ้มค่ากับงานของคุณที่สุด

กล้องที่ใช่ ไม่ได้วัดแค่ที่ความละเอียดหรือราคา แต่ต้องตอบโจทย์ "ความต้องการจริงในหน้างาน"

1.6 ปฏิบัติการ: การติดตั้งเครื่องมือและการจัดการภาพพื้นฐาน

ในภาคปฏิบัติของหัวข้อนี้ ผู้เรียนจะเริ่มต้นติดตั้งเครื่องมือพัฒนา (Python และ OpenCV) และฝึกเขียนโปรแกรมพื้นฐานสำหรับการอ่าน แสดงผล และจัดการภาพดิจิทัลเบื้องต้น ซึ่งเป็นทักษะพื้นฐานที่จำเป็นก่อนเข้าสู่การประมวลผลภาพขั้นสูงในหัวข้อถัดไป

ขั้นตอนที่ 1: การติดตั้ง Python และไลบรารีที่จำเป็น

แนะนำให้ใช้ Python เวอร์ชัน 3.9 ขึ้นไป และติดตั้งไลบรารีหลักที่ใช้ตลอดหลักสูตรผ่านคำสั่ง pip ดังนี้

```
# ตรวจสอบเวอร์ชัน Python
python --version

# ติดตั้งไลบรารีที่จำเป็น
pip install opencv-python numpy matplotlib

# ตรวจสอบการติดตั้ง OpenCV
python -c "import cv2; print(cv2.__version__)"
```

ขั้นตอนที่ 2: การอ่านและแสดงผลภาพด้วย OpenCV

ฟังก์ชันพื้นฐานที่ใช้บ่อยที่สุดในการทำงานกับไฟล์ภาพของ OpenCV คือ cv2.imread() สำหรับอ่านภาพ cv2.imshow() สำหรับแสดงผลภาพในหน้าต่าง และ cv2.imwrite() สำหรับบันทึกภาพลงไฟล์ ข้อควรระวังที่สำคัญ คือ OpenCV จะจัดเก็บและอ่านภาพสีในลำดับช่องสัญญาณแบบ BGR (Blue-Green-Red) ซึ่งต่างจากไลบรารีอื่น เช่น Matplotlib หรือ PIL ที่ใช้ลำดับ RGB จึงต้องแปลงลำดับช่องสีก่อนนำไปแสดงผลด้วย Matplotlib เสมอ

```
import cv2

# อ่านภาพจากไฟล์ (BGR order)
img = cv2.imread('part.jpg')

# แสดงขนาดภาพ: (height, width, channels)
print('Shape:', img.shape)
print('Dtype:', img.dtype)

# แสดงผลภาพในหน้าต่าง
cv2.imshow('Original Image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

# บันทึกภาพ
cv2.imwrite('output.png', img)
```

ขั้นตอนที่ 3: การเข้าถึงค่าพิกเซลและการตัดพื้นที่ภาพ (ROI)

ภาพที่อ่านเข้ามาใน OpenCV จะถูกจัดเก็บเป็นอาร์เรย์ NumPy ทำให้สามารถเข้าถึงค่าพิกเซลหรือตัดพื้นที่เฉพาะส่วน (Region of Interest: ROI) ได้โดยใช้เทคนิค Array Slicing ของ NumPy โดยตรง ซึ่งเป็นเทคนิคที่ใช้บ่อยมากในการจำกัดพื้นที่ตรวจสอบให้อยู่เฉพาะบริเวณที่สนใจ เพื่อลดเวลาการประมวลผลของระบบ

```
# ค่าพิกเซลที่ตำแหน่ง (row=100, col=200) รูปแบบ BGR
pixel = img[100, 200]
print('BGR value:', pixel)

# ตัดพื้นที่สนใจ (ROI): แถว 50-250, คอลัมน์ 100-400
roi = img[50:250, 100:400]
cv2.imshow('ROI', roi)
cv2.waitKey(0)

# แปลงเป็นภาพขาวดำ (Grayscale) เพื่อลดข้อมูลก่อนประมวลผลต่อ
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
print('Grayscale shape:', gray.shape)
```

ขั้นตอนที่ 4: การอ่านภาพแบบต่อเนื่องจากกล้องเว็บแคม (Real-time Capture)

ในงาน Machine Vision จริง ภาพส่วนใหญ่จะถูกอ่านเข้ามาแบบต่อเนื่องเป็นวิดีโอสตรีมจากกล้อง ไม่ใช่จากไฟล์ภาพนิ่งเพียงอย่างเดียว OpenCV มีคลาส cv2.VideoCapture สำหรับเชื่อมต่อกับกล้องเว็บแคมหรือกล้องอุตสาหกรรมที่รองรับ USB Video Class เพื่ออ่านภาพทีละเฟรมแบบวนลูปได้ทันที

```
import cv2

cap = cv2.VideoCapture(0)   # 0 = กล้องตัวแรกที่เชื่อมต่อกับเครื่อง
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('Live Camera', gray)

    # กด 'q' เพื่อออกจากลูป
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

ข้อควรระวังในการอ่านภาพแบบต่อเนื่องคือประสิทธิภาพของลูปประมวลผล (Loop) ต้องเร็วเพียงพอที่จะตามอัตราเฟรมของกล้องได้ทัน หากขั้นตอนประมวลผลภายในลูปใช้เวลานานเกินไป จะทำให้ภาพที่แสดงผลหน่วงหรือกล้องไม่สามารถส่งภาพได้ทันความเร็วที่ตั้งไว้ ซึ่งเป็นประเด็นสำคัญที่ต้องคำนึงถึงเมื่อออกแบบระบบตรวจสอบแบบ Real-time บนสายการผลิตจริง

ขั้นตอนที่ 5(เพิ่มเติม): ทดสอบใช้งาน OpenCV ด้วย Google Colab

ขั้นตอนที่ 6(เพิ่มเติม): ทดสอบใช้งาน OpenCV ตามเอกสาร OpenCV 101 – Beginner

ขั้นตอนที่ 7(เพิ่มเติม): ทดสอบใช้งาน OpenCV ตามเอกสาร OpenCV 102 – Image and Video

1.7 แบบฝึกหัดท้ายหัวข้อ

Q11: เขียนโปรแกรมอ่านภาพชิ้นงานจากกล้อง แสดงค่าความละเอียด (Resolution) และจำนวนช่องสัญญาณสี

Q12: ทดลองตัด ROI เฉพาะบริเวณที่ต้องการตรวจสอบ แล้วบันทึกเป็นไฟล์ภาพแยกต่างหาก

Q13: เปรียบเทียบภาพเดียวกันที่ถ่ายภายใต้ไฟส่องสว่างต่างชนิดกัน (Direct vs Diffuse) แล้วอภิปรายผลต่างของคุณภาพภาพ

Q14: ให้บอกแนวทางการใช้งาน Machine Vision กับงานที่ผู้เรียนเกี่ยวข้องหรือสนใจ

หัวข้อที่ 2: การประมวลผลภาพขั้นพื้นฐาน (OpenCV)

หลังจากที่สามารถอ่าน แสดงผล และจัดการภาพเบื้องต้นได้แล้ว หัวข้อนี้จะเจาะลึกเทคนิคการประมวลผลภาพขั้นพื้นฐานด้วย OpenCV ซึ่งเป็นพื้นฐานสำคัญที่ต้องใช้ต่อยอดไปสู่งาน Segmentation การวัดขนาด และ Deep Learning ในหัวข้อถัดไปของหลักสูตร

2.1 การปรับปรุงคุณภาพภาพ (Image Enhancement)

ภาพที่ได้จากกล้องในสภาพแสงจริงมักมีความสว่าง (Brightness) หรือความต่างระดับสี (Contrast) ที่ไม่เหมาะสมต่อการประมวลผลขั้นถัดไป การปรับปรุงคุณภาพภาพจึงเป็นขั้นตอนที่ช่วยเพิ่มความแม่นยำให้กับอัลกอริทึมที่ทำงานต่อจากขั้นตอนนี้

2.1.1 การปรับ Contrast และ Brightness

การปรับความสว่างและความต่างระดับสีสามารถทำได้ด้วยการแปลงเชิงเส้น (Linear Transformation) ของค่าพิกเซลตามสมการ

$$g(x,y) = \alpha f(x,y) + \beta$$

โดยที่ f(x,y) คือค่าพิกเซลต้นฉบับ

g(x,y) คือค่าพิกเซลหลังปรับ

α (Gain) ควบคุมค่า Contrast โดย α มากกว่า 1 จะเพิ่มความต่างระดับสี

β (Bias) ควบคุมค่า Brightness โดยค่าบวกจะทำให้ภาพสว่างขึ้น

```
import cv2

img = cv2.imread('part.jpg')

# alpha = contrast (1.0-3.0), beta = brightness (0-100)
adjusted = cv2.convertScaleAbs(img, alpha=1.5, beta=30)
cv2.imshow('Adjusted', adjusted)
cv2.waitKey(0)
```

2.1.2 Histogram Equalization และ CLAHE

Histogram Equalization เป็นเทคนิคปรับกระจายค่าความสว่างของภาพให้ครอบคลุมทุกระดับอย่างสม่ำเสมอ ทำให้รายละเอียดในส่วนมืดหรือสว่างจัดมองเห็นได้ชัดเจนขึ้น เหมาะกับภาพที่มีแสงไม่สม่ำเสมอทั้งภาพ แต่หากใช้กับภาพที่มีความต่างของแสงเฉพาะบางบริเวณ ควรใช้เทคนิค CLAHE (Contrast Limited Adaptive Histogram Equalization) ซึ่งแบ่งภาพเป็นบล็อกย่อยแล้วปรับ Histogram แยกทีละบล็อก จึงให้ผลลัพธ์ที่เป็นธรรมชาติกว่าและไม่ขยาย Noise มากเกินไป

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Histogram Equalization แบบทั่วไป
eq = cv2.equalizeHist(gray)

# CLAHE: ปรับ Histogram แบบแบ่งพื้นที่ย่อย
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
clahe_img = clahe.apply(gray)
```

2.1.3 การสร้างและอ่านค่า Histogram ของภาพ

Histogram คือกราฟแสดงการกระจายตัวของค่าความสว่างของพิกเซลทั้งหมดในภาพ แกนนอนแทนระดับความสว่าง (0–255) และแกนตั้งแทนจำนวนพิกเซลที่มีค่าความสว่างนั้น ๆ การอ่านค่า Histogram ช่วยให้ผู้ตรวจสอบทราบได้อย่างรวดเร็วว่าภาพมืดเกินไป สว่างเกินไป หรือมี Contrast ต่ำหรือไม่ ก่อนตัดสินใจเลือกเทคนิคปรับปรุงภาพที่เหมาะสม

```
import cv2
import matplotlib.pyplot as plt

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
hist = cv2.calcHist([gray], [0], None, [256], [0, 256])

plt.plot(hist)
plt.title('Grayscale Histogram')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
plt.show()
```

หาก Histogram มีลักษณะกระจุกตัวอยู่ในช่วงแคบ ๆ ด้านซ้าย (ค่าน้อย) แสดงว่าภาพมืดเกินไป หากกระจุกตัวด้านขวา (ค่ามาก) แสดงว่าภาพสว่างเกินไป และหากกระจุกตัวอยู่ตรงกลางช่วงแคบ ๆ แสดงว่าภาพมี Contrast ต่ำ ซึ่งในทั้งสามกรณีสามารถแก้ไขได้ด้วยเทคนิคที่กล่าวมาข้างต้น เช่น การปรับ Contrast/Brightness หรือ Histogram Equalization

2.2 การกรองสัญญาณรบกวน (Filtering)

สัญญาณรบกวน (Noise) ในภาพอุตสาหกรรมมักเกิดจากคุณภาพเซนเซอร์ สภาพแสงไม่คงที่ หรือฝุ่นละอองในสภาพแวดล้อมการผลิต ซึ่งอาจส่งผลให้ขั้นตอนการหาขอบหรือ Segmentation ในภายหลังผิดพลาดได้ การกรองภาพจึงเป็นขั้นตอนที่ช่วยลด Noise ก่อนนำไปประมวลผลต่อ

ตัวกรอง (Filter)	หลักการทำงาน	จุดเด่น / ข้อจำกัด
Averaging / Blur	หาค่าเฉลี่ยของพิกเซลในหน้าต่าง Kernel	ทำง่าย เร็ว แต่ทำให้ขอบภาพเบลอ
Gaussian Blur	ถ่วงน้ำหนักตามระยะห่างจากจุดศูนย์กลางแบบ Gaussian	ลด Noise แบบสุ่มได้ดี ภาพเบลอน้อยกว่า Averaging
Median Blur	แทนค่าพิกเซลด้วยค่ามัธยฐานในหน้าต่าง Kernel	กำจัด Salt-and-Pepper Noise ได้ดีเยี่ยม รักษาขอบได้ดี
Bilateral Filter	ถ่วงน้ำหนักตามทั้งระยะห่างและความต่างค่าสี	ลด Noise ขณะรักษาขอบภาพให้คมชัด แต่ประมวลผลช้ากว่า

```
# Gaussian Blur: kernel 5x5
gauss = cv2.GaussianBlur(img, (5, 5), 0)

# Median Blur: เหมาะกับ salt-and-pepper noise
median = cv2.medianBlur(img, 5)

# Bilateral Filter: ลด noise แต่รักษาขอบภาพ
bilateral = cv2.bilateralFilter(img, 9, 75, 75)
```

2.3 การแปลงพื้นที่สี (Color Space Conversion)

การเลือกพื้นที่สี (Color Space) ที่เหมาะสมมีผลอย่างมากต่อความง่ายในการประมวลผลขั้นถัดไป โดยเฉพาะงานแยกวัตถุตามสี (Color Segmentation) ระบบสี HSV (Hue, Saturation, Value) ได้รับความนิยมมากกว่า RGB ในงาน Machine Vision เนื่องจากแยกข้อมูล "สี" (Hue) ออกจาก "ความสว่าง" (Value) ทำให้การแยกวัตถุตามสียังคงทำงานได้ดีแม้สภาพแสงเปลี่ยนแปลงไปบ้าง ซึ่งต่างจาก RGB ที่ค่าทั้งสามช่องจะเปลี่ยนแปลงไปพร้อมกันเมื่อความสว่างเปลี่ยน

```
# แปลงพื้นที่สี
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
hsv  = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

# ตัวอย่าง: แยกวัตถุสีแดงด้วย HSV thresholding
lower_red = (0, 120, 70)
upper_red = (10, 255, 255)
mask = cv2.inRange(hsv, lower_red, upper_red)
result = cv2.bitwise_and(img, img, mask=mask)
```

นอกจาก HSV แล้ว ยังมีพื้นที่สีแบบ Lab (Lightness, a, b) ซึ่งออกแบบมาให้ใกล้เคียงกับการรับรู้สีของสายตามนุษย์มากที่สุด ค่า L แทนความสว่าง ส่วนค่า a และ b แทนตำแหน่งของสีบนแกนเขียว-แดง และน้ำเงิน-เหลืองตามลำดับ พื้นที่สีนี้มักถูกใช้ในงานตรวจสอบสีของผลิตภัณฑ์ (Color Inspection) ที่ต้องการความแม่นยำสูง เช่น การตรวจสอบสีของสีพ่นรถยนต์หรือสีของบรรจุภัณฑ์อาหาร เนื่องจากค่าความต่างของสี (Color Difference, ΔE) ที่คำนวณในพื้นที่สี Lab จะสอดคล้องกับความรู้สึกของสายตามนุษย์มากกว่าการคำนวณในพื้นที่สี RGB โดยตรง

2.4 การดำเนินการทางสัณฐานวิทยาเบื้องต้น (Morphological Operations)

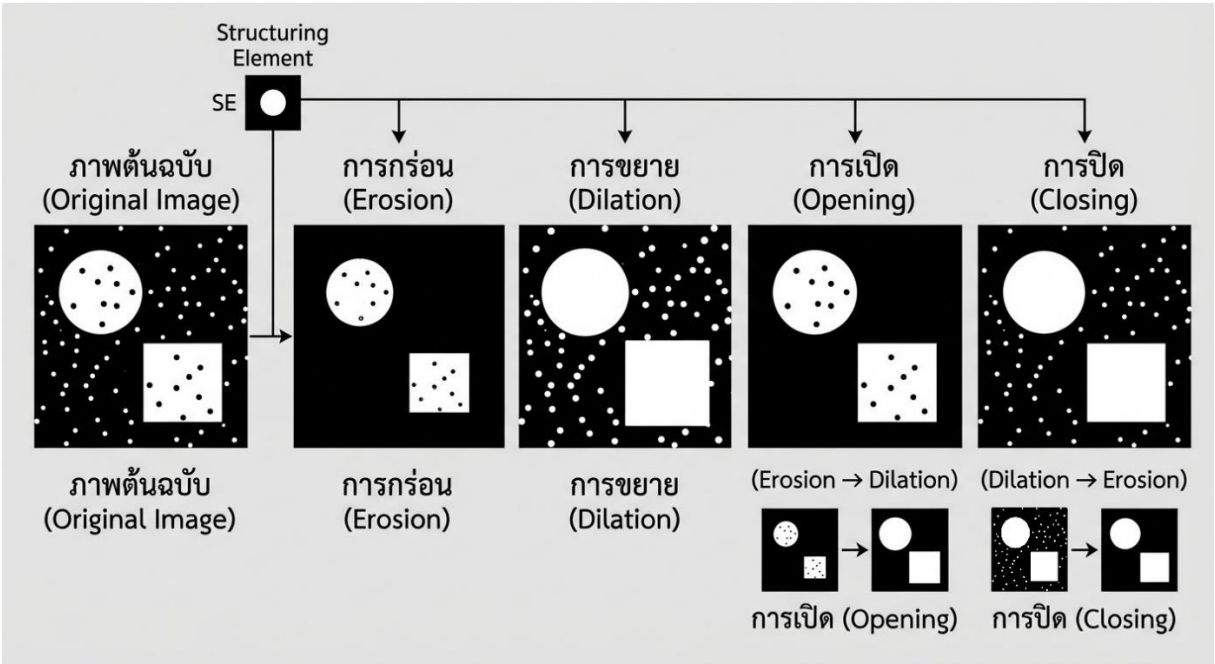
การดำเนินการทางสัณฐานวิทยาเป็นเทคนิคที่ทำงานบนภาพไบนารี (Binary Image) หรือภาพขาวดำ โดยใช้โครงสร้างเล็ก ๆ ที่เรียกว่า Kernel หรือ Structuring Element เลื่อนไปทั่วภาพเพื่อปรับแต่งรูปทรงของวัตถุ เทคนิคนี้มีประโยชน์มากในการลบจุด Noise เล็ก ๆ หรือเชื่อมช่องว่างเล็ก ๆ ในวัตถุก่อนนำไปวัดขนาดหรือนับจำนวนในขั้นตอนถัดไป

การดำเนินการ	หลักการ	ผลลัพธ์ที่ได้
Erosion (การกัดกร่อน)	ลบพิกเซลขอบของวัตถุออก	วัตถุมีขนาดเล็กลง จุด Noise เล็ก ๆ หายไป
Dilation (การขยาย)	เพิ่มพิกเซลที่ขอบของวัตถุ	วัตถุมีขนาดใหญ่ขึ้น ช่องว่างเล็ก ๆ ถูกเติมเต็ม
Opening (Erosion แล้ว Dilation)	กัดกร่อนก่อนแล้วจึงขยายกลับ	ลบจุด Noise ขนาดเล็กโดยไม่เปลี่ยนขนาดวัตถุหลัก
Closing (Dilation แล้ว Erosion)	ขยายก่อนแล้วจึงกัดกร่อนกลับ	ปิดรูหรือช่องว่างเล็ก ๆ ภายในวัตถุ

```
import numpy as np
kernel = np.ones((5, 5), np.uint8)

erosion  = cv2.erode(binary_img, kernel, iterations=1)
dilation = cv2.dilate(binary_img, kernel, iterations=1)
opening  = cv2.morphologyEx(binary_img, cv2.MORPH_OPEN, kernel)
closing  = cv2.morphologyEx(binary_img, cv2.MORPH_CLOSE, kernel)
```

ในทางปฏิบัติ Opening มักถูกใช้ก่อนขั้นตอนการนับจำนวนวัตถุเสมอ เพื่อกำจัดจุด Noise ขนาดเล็กที่หลงเหลือจากขั้นตอน Thresholding ไม่ให้ถูกนับเป็นวัตถุปลอม ในขณะที่ Closing มักใช้เมื่อวัตถุที่ต้องการวัดมีรูพรุนเล็กๆ อยู่ภายในซึ่งไม่ควรถูกนับเป็นข้อบกพร่อง


Structuring Element
SE
ภาพต้นฉบับ (Original Image)
การกร่อน (Erosion)
การขยาย (Dilation)
การเปิด (Opening)
การปิด (Closing)
ภาพต้นฉบับ (Original Image)
การกร่อน (Erosion)
การขยาย (Dilation)
(Erosion → Dilation)
(Dilation → Erosion)
การเปิด (Opening)
การปิด (Closing)

2.5 การค้นหาขอบ (Edge Detection)

ขอบภาพ (Edge) คือตำแหน่งที่ค่าความสว่างของพิกเซลเปลี่ยนแปลงอย่างรวดเร็ว ซึ่งมักตรงกับขอบเขตของวัตถุในภาพ การหาขอบเป็นขั้นตอนพื้นฐานที่สำคัญมากในงานวัดขนาดและตรวจสอบรูปทรง โดยหลักการทางคณิตศาสตร์ใช้การหาอนุพันธ์ (Gradient) ของค่าความสว่างตามแนวแกน x และ y

2.5.1 ตัวกรองแบบ Sobel และ Laplacian

Sobel Operator คำนวณค่าอนุพันธ์อันดับหนึ่งของภาพในแนวนอน (Gx) และแนวตั้ง (Gy) แยกกัน แล้วนำมารวมเป็นค่าขนาดของ Gradient ส่วน Laplacian คำนวณอนุพันธ์อันดับสองซึ่งไวต่อ Noise มากกว่า จึงมักใช้ร่วมกับการเบลอภาพก่อนเสมอ

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

sobel_x = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
sobel_y = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)
laplacian = cv2.Laplacian(gray, cv2.CV_64F)
```

2.5.2 Canny Edge Detection

Canny เป็นอัลกอริทึมหาขอบที่ได้รับความนิยมสูงสุดในงาน Machine Vision เนื่องจากให้เส้นขอบที่บางและต่อเนื่อง อัลกอริทึมประกอบด้วย 4 ขั้นตอนหลัก ได้แก่

- 1) ลด Noise ด้วย Gaussian Filter ก่อนเสมอ
- 2) คำนวณ Gradient ของความสว่างด้วย Sobel Operator ทั้งขนาดและทิศทาง
- 3) Non-Maximum Suppression: ลดความหนาของเส้นขอบให้เหลือ 1 พิกเซล โดยเก็บเฉพาะจุดที่มีค่า Gradient สูงสุดในทิศทางตั้งฉากกับขอบ
- 4) Hysteresis Thresholding: ใช้ค่า Threshold สองระดับ (สูง/ต่ำ) เพื่อตัดสินว่าพิกเซลใดเป็นขอบจริง โดยเชื่อมต่อขอบที่มีค่าอยู่ระหว่างสองระดับเข้ากับขอบที่มีค่าสูงกว่าระดับบน

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (5, 5), 0)

# threshold1 = lower bound, threshold2 = upper bound
edges = cv2.Canny(blur, 50, 150)
cv2.imshow('Canny Edges', edges)
cv2.waitKey(0)
```

การตั้งค่า Threshold ทั้งสองระดับของ Canny มีผลอย่างมากต่อผลลัพธ์ หาก Threshold ต่ำเกินไปจะได้เส้นขอบที่มี Noise ปะปนมาก ในขณะที่ Threshold สูงเกินไปจะทำให้ขอบที่ต้องการขาดหายไป จึงควรปรับค่าและทดสอบกับภาพตัวอย่างจริงก่อนนำไปใช้งานเสมอ

2.5.3 การเลือก Threshold อัตโนมัติสำหรับ Canny

ในงานที่ต้องประมวลผลภาพจำนวนมากหรือภาพที่มีสภาพแสงเปลี่ยนแปลงตลอดเวลา การกำหนดค่า Threshold แบบตายตัวอาจไม่เหมาะสมกับทุกภาพ เทคนิคที่นิยมใช้ในทางปฏิบัติคือการคำนวณค่า Threshold จากค่ามัธยฐาน (Median) ของความสว่างในภาพ แล้วกำหนดขอบเขตบนและล่างเป็นสัดส่วนจากค่ามัธยฐานนั้น ซึ่งช่วยให้ค่า Threshold ปรับตัวตามลักษณะของแต่ละภาพโดยอัตโนมัติ

```
import numpy as np

def auto_canny(gray, sigma=0.33):
    v = np.median(gray)
    lower = int(max(0, (1.0 - sigma) * v))
    upper = int(min(255, (1.0 + sigma) * v))
    return cv2.Canny(gray, lower, upper)

edges = auto_canny(blur)
```

ค่า sigma เป็นพารามิเตอร์ควบคุมความกว้างของช่วง Threshold รอบค่ามัธยฐาน โดยค่าที่นิยมใช้เริ่มต้นคือ 0.33 หากต้องการให้ตรวจจับขอบได้ไวขึ้น (Sensitive) ให้เพิ่มค่า sigma และหากต้องการลดความไวต่อ Noise ให้ลดค่า sigma ลง

2.6 การแปลงแบบฮาฟ (Hough Transform)

Hough Transform เป็นเทคนิคทางคณิตศาสตร์ที่ใช้ตรวจหารูปทรงเรขาคณิตที่มีสมการแน่นอน เช่น เส้นตรงหรือวงกลม จากจุดขอบภาพที่ได้จากขั้นตอน Edge Detection โดยหลักการคือแปลงจุดในพื้นที่ภาพ (Image Space) ไปเป็นเส้นโค้งในพื้นที่พารามิเตอร์ (Parameter Space) แล้วหาจุดที่เส้นโค้งจำนวนมากตัดกัน ซึ่งจุดตัดนั้นจะสอดคล้องกับพารามิเตอร์ของรูปทรงที่ต้องการหา

2.6.1 Hough Line Transform

ใช้สำหรับตรวจหาเส้นตรงในภาพ โดยแทนเส้นตรงด้วยพารามิเตอร์ระยะตั้งฉากจากจุดกำเนิด (ρ) และมุม (θ) แทนที่จะใช้รูปแบบ y = mx + c เพื่อหลีกเลี่ยงปัญหาเส้นตรงแนวดิ่งที่ค่าความชัน m เป็นค่าอนันต์

```
edges = cv2.Canny(gray, 50, 150)

# HoughLinesP: คืนค่าจุดเริ่มต้น-สิ้นสุดของเส้นตรง (แนะนำ)
lines = cv2.HoughLinesP(edges, 1, 3.14159/180,
                        threshold=80, minLineLength=50, maxLineGap=10)

for line in lines:
    x1, y1, x2, y2 = line[0]
    cv2.line(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
```

2.6.2 Hough Circle Transform

ใช้สำหรับตรวจหาวงกลมในภาพ นิยมนำไปใช้ตรวจนับรูเจาะ ชิ้นงานทรงกลม หรือฝาขวดในสายการผลิต โดยมีพารามิเตอร์ที่ต้องปรับให้เหมาะสมกับขนาดวงกลมที่คาดว่าจะพบในภาพ

```
gray = cv2.medianBlur(gray, 5)

circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, dp=1, minDist=20,
                           param1=50, param2=30,
                           minRadius=5, maxRadius=50)

if circles is not None:
    circles = circles[0]
    print('พบวงกลมทั้งหมด:', len(circles), 'วง')
    for x, y, r in circles:
        cv2.circle(img, (int(x), int(y)), int(r), (0, 255, 0), 2)
```

Hough Transform เป็นเทคนิคที่มีประโยชน์มากในงานตรวจนับชิ้นงานทรงกลม เช่น การนับจำนวนรูเจาะบนแผ่นโลหะ หรือตรวจสอบตำแหน่งฝาขวดบนสายพาน แต่ต้องอาศัยการปรับพารามิเตอร์อย่างระมัดระวัง เนื่องจากมีความไวต่อค่า Threshold ของขั้นตอน Edge Detection ที่ทำมาก่อนหน้า

2.7 ตารางสรุปฟังก์ชัน OpenCV พื้นฐานที่ใช้ในหัวข้อนี้

เพื่อให้ผู้เรียนสามารถทบทวนและนำไปใช้อ้างอิงได้สะดวก ตารางต่อไปนี้สรุปฟังก์ชัน OpenCV หลักที่ได้เรียนรู้ตลอดหัวข้อการประมวลผลภาพขั้นพื้นฐาน พร้อมหมวดหมู่และหน้าที่การทำงานโดยย่อ

ฟังก์ชัน	หมวดหมู่	หน้าที่การทำงาน
cv2.imread cv2.imshow cv2.imwrite	I/O พื้นฐาน	อ่าน แสดงผล และบันทึกไฟล์ภาพ
cv2.cvtColor	Color Space	แปลงพื้นที่สี เช่น BGR⟶GRAY, BGR⟶HSV
cv2.convertScaleAbs	Enhancement	ปรับค่า Contrast และ Brightness
cv2.equalizeHist cv2.createCLAHE	Enhancement	ปรับกระจาย Histogram ให้สม่ำเสมอ
cv2.calcHist	Analysis	คำนวณค่า Histogram ของภาพ
cv2.GaussianBlur cv2.medianBlur cv2.bilateralFilter	Filtering	ลดสัญญาณรบกวนในภาพ
cv2.inRange cv2.bitwise_and cv2.bitwise_or	Segmentation	แยกวัตถุตามช่วงค่าสีที่กำหนด
cv2.erode cv2.dilate cv2.morphologyEx	Morphology	ปรับแต่งรูปทรงวัตถุในภาพไบนารี opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel) closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
cv2.Sobel cv2.Laplacian cv2.Canny	Edge Detection	คำนวณและตรวจหาขอบภาพ
cv2.HoughLinesP cv2.HoughCircles	Shape Detection	ตรวจหาเส้นตรงและวงกลมในภาพ
cv2.VideoCapture	Real-time I/O	อ่านภาพต่อเนื่องจากกล้อง/วิดีโอ

2.8 ปฏิบัติการ: ทดลองใช้ฟังก์ชัน OpenCV ขั้นพื้นฐาน

ในภาคปฏิบัติสุดท้ายของ Day-01 ผู้เรียนจะประกอบขั้นตอนทั้งหมดที่เรียนมาเข้าด้วยกันเป็นไปป์ไลน์ (Pipeline) เดียว เพื่อพัฒนาโปรแกรมตรวจนับวัตถุทรงกลมอย่างง่ายจากภาพถ่าย ซึ่งเป็นรากฐานของงานตรวจนับชิ้นงานในสายการผลิตจริง

ไปป์ไลน์ตัวอย่าง: ตรวจนับวัตถุทรงกลมจากภาพ

```
import cv2

# 1. อ่านภาพและแปลงเป็นขาวดำ
img = cv2.imread('coins.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 2. ลด Noise ด้วย Median Blur
blur = cv2.medianBlur(gray, 5)

# 3. ปรับ Contrast ด้วย CLAHE ก่อนหาขอบ
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
enhanced = clahe.apply(blur)

# 4. ตรวจหาวงกลมด้วย Hough Circle Transform
circles = cv2.HoughCircles(enhanced, cv2.HOUGH_GRADIENT, dp=1,
                           minDist=25, param1=50, param2=30,
                           minRadius=8, maxRadius=60)

# 5. วาดผลลัพธ์และสรุปจำนวน
count = 0
if circles is not None:
    for x, y, r in circles[0]:
        cv2.circle(img, (int(x), int(y)), int(r), (0, 255, 0), 2)
        count += 1

print(f'ตรวจพบวัตถุทรงกลมทั้งหมด {count} ชิ้น')
cv2.imwrite('result.png', img)
```

โจทย์เพิ่มเติม: ตรวจจับขอบชิ้นงานสี่เหลี่ยมและวัดความยาวด้าน

นอกจากวัตถุทรงกลม งานตรวจสอบในสายการผลิตจริงจำนวนมากเป็นชิ้นงานทรงสี่เหลี่ยมหรือหลายเหลี่ยม ตัวอย่างไปป์ไลน์ต่อไปนี้แสดงการรวม Canny Edge Detection เข้ากับ HoughLinesP เพื่อตรวจจับด้านของชิ้นงานสี่เหลี่ยมและคำนวณความยาวแต่ละด้านเป็นหน่วยพิกเซล

```
import cv2, math

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (5, 5), 0)
edges = cv2.Canny(blur, 50, 150)

lines = cv2.HoughLinesP(edges, 1, 3.14159/180,
                        threshold=60, minLineLength=40, maxLineGap=5)

for line in lines:
    x1, y1, x2, y2 = line[0]
    length = math.hypot(x2 - x1, y2 - y1)
    print(f'ด้านความยาว: {length:.1f} พิกเซล')
    cv2.line(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
```

เมื่อทราบค่าอัตราส่วนพิกเซลต่อมิลลิเมตร (Calibration Factor) ที่ได้จากการปรับเทียบกล้องล่วงหน้าแล้ว สามารถแปลงความยาวที่วัดได้เป็นหน่วยจริง (มิลลิเมตร) เพื่อนำไปเปรียบเทียบกับค่ามาตรฐานของชิ้นงานในขั้นตอนการตัดสินใจต่อไปได้

2.9 แบบฝึกหัดท้ายหัวข้อ

Q21: ถ่ายภาพเหรียญจำนวนหนึ่ง แล้วใช้โปรแกรมนับจำนวนวัตถุ โดยปรับค่า param1, param2, minRadius และ maxRadius ของ HoughCircles แล้วสังเกตผลกระทบต่อจำนวนวัตถุที่ตรวจพบ

Q23: จากไปป์ไลน์ตัวอย่าง เปรียบเทียบผมของการทำ Median Blur กับ Gaussian Blur แล้ววิเคราะห์ความแตกต่างของผลลัพธ์การนับวัตถุ

Q22: เลือกภาพจากอินเทอร์เน็ตจำนวน 3 ภาพ (X1, X2, X3) แต่ละภาพให้เปรียบเทียบผลลัพธ์การหาขอบระหว่าง Sobel, Laplacian และ Canny บนภาพชิ้นงานเดียวกัน

Q24: เลือกภาพจากอินเทอร์เน็ตจำนวน 3 ภาพ (Y1, Y2, Y3) แต่ละภาพให้ใช้ HoughLinesP ตรวจหาขอบแนวตรงของชิ้นงานสี่เหลี่ยม แล้วคำนวณความยาวของแต่ละด้านเป็นหน่วยพิกเซล

สรุปหัวข้อวันที่ 1 และเชื่อมโยงสู่วันที่ 2

ในวันแรกของหลักสูตร ผู้เรียนได้เรียนรู้พื้นฐานสำคัญของระบบ Machine Vision ตั้งแต่หลักการทำงาน ชนิดของภาพดิจิทัล การเลือกฮาร์ดแวร์ (กล้อง เลนส์ ไฟส่องสว่าง) ไปจนถึงเทคนิคการประมวลผลภาพขั้นพื้นฐานด้วย OpenCV ได้แก่ การปรับปรุงคุณภาพภาพ การกรองสัญญาณรบกวน การแปลงพื้นที่สี การหาขอบ และการแปลงแบบฮาฟ ความรู้และทักษะทั้งหมดนี้เป็นรากฐานสำคัญที่จะนำไปต่อยอดสู่หัวข้อการแบ่งส่วนภาพและการวัดขนาดชิ้นงานด้วยวิธี Traditional Computer Vision ใน Day-02 ต่อไป