Received: 28 ก.พ. 2566 **Revised:** 15 มี.ค. 2567 **Accepted:** 18 มี.ค. 2567

การตรวจจับสิ่งเจือปนด้วยการประมวลผลภาพบนหัวอ่านฮาร์ดไดร์ฟ

Detection of contaminants using image processing on a hard drive reader

สราวุฒิ ภัทรภูวดล[1] สนิท สิทธิ[2] ภานุวัฒน์ เมฆะ[3] และ พยุงศักดิ์ เกษมสำราญ[4*]
[1234]สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้ เชียงใหม่ 50290 ประเทศไทย

Sarawut Phattaraphuwadol[1], Snit Sitti[2],
Panuwat Mekha[3] and Payungsak Kasemsumran[4*]
[1234]Department of Computer Science, Faculty of Science, Maejo University,
ChiangMai, 50290, Thailand
[*]Corressponding anthor: payungsak.kae@gmail.com

Abstract

This research has conducted experiments to detect small contaminants through image processing using the Python language. The objective is to develop an algorithm for detecting contaminants on the Head Gimbal Assembly (HGA) of hard disk drives. This is aimed at addressing the performance issues of the read/write head caused by dust particles. The small and delicate size of the read/write head makes it susceptible to dust, which can affect its efficiency. Traditional visual inspection by humans is time-consuming and prone to errors.

The research has developed an algorithm to be used with machinery for detecting small contaminants, providing assistance and reducing the burden of visual inspections by humans alone. Experimental results show that this algorithm can successfully detect contaminants on the read/write heads of hard disk drives with an accuracy of 86.25% and a specificity of 91.67%. This is achieved through the use of image pixelation and logic gates in the image transformation process.

Keywords : *Head Gimbal Assembly; Image Processing; Python language*

บทคัดย่อ

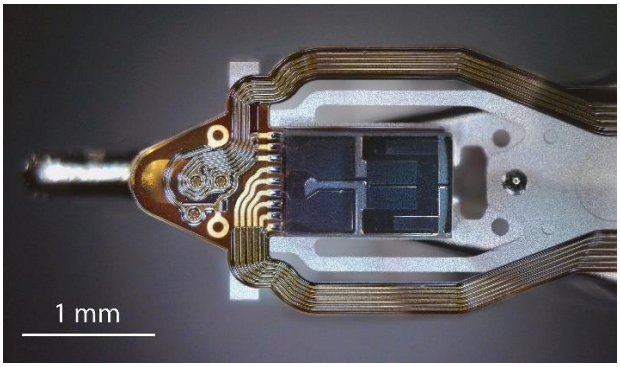
งานวิจัยนี้ได้ทำการทดลองเพื่อตรวจจับสิ่งปนเปื้อนขนาดเล็กผ่านการประมวลผลภาพโดยใช้ภาษา Python วัตถุประสงค์คือเพื่อพัฒนาอัลกอริธึมสำหรับการตรวจจับสิ่งปนเปื้อนบน Head Gimbal Assembly (HGA) ของฮาร์ดดิสก์ไดรฟ์ โดยมีจุดมุ่งหมายเพื่อแก้ไขปัญหาประสิทธิภาพของหัวอ่าน/เขียนที่เกิดจากอนุภาคฝุ่น หัวอ่าน/เขียนที่มีขนาดเล็กและละเอียดอ่อนทำให้ไวต่อฝุ่น ซึ่งอาจส่งผลต่อประสิทธิภาพ การตรวจสอบด้วยสายตาแบบดั้งเดิมโดยมนุษย์ใช้เวลานานและมีแนวโน้มที่จะเกิดข้อผิดพลาด

การวิจัยได้พัฒนาอัลกอริธึมสำหรับใช้กับเครื่องจักรในการตรวจจับสิ่งปนเปื้อนขนาดเล็ก ให้ความช่วยเหลือและลดภาระในการตรวจสอบด้วยสายตาโดยมนุษย์เพียงอย่างเดียว ผลการทดลองแสดงให้เห็นว่าอัลกอริทึมนี้สามารถตรวจจับสิ่งปนเปื้อนบนหัวอ่าน/เขียนของฮาร์ดดิสก์ไดรฟ์ได้สำเร็จด้วยความแม่นยำ 86.25% และความจำเพาะ 91.67% ซึ่งสามารถทำได้โดยการใช้พิกเซลของภาพและลอจิกเกตในกระบวนการแปลงภาพ

คำสำคัญ : *หัวอ่านเขียนของฮาร์ดดิสก์; การประมวลผลภาพ; ภาษา Python*

1. บทนำ

ยุคสมัยปัจจุบันมีองค์ความรู้และข้อมูลดิบมากมายที่ถูกเก็บรักษาในรูปแบบของดิจิทัล ทำให้อุปกรณ์ที่ใช้ในการบันทึกข้อมูลต้องมีความทนทานและความถูกต้องสูง เพื่อไม่ให้ข้อมูลที่ถูกบันทึกมีความเสียหาย การเก็บบันทึกข้อมูลด้วยฮาร์ดดิสก์ไดร์ฟ (Hard Disk Drive : HDD) ซึ่งเป็นฮาร์ดดิสก์ที่มีการบันทึกข้อมูลแบบจานหมุนเพื่อป้องกันการสูญหายของข้อมูลจากกรณีต่างๆ เพราะถ้าตัวจานหมุนไม่ได้รับความเสียหายยังคงกู้คือข้อมูลที่บันทึกไว้ได้ เทคโนโลยีของฮาร์ดดิสก์ในปัจจุบันข้อมูลในฮาร์ดดิสก์จะถูกบันทึกและอ่านจากจานด้วยหัวอ่าน/เขียนฮาร์ดดิสก์ (Head Gimbal Assembly : HGA) ซึ่งมีลักษณะเป็นเข็มขนาดเล็กที่บริเวณส่วนหัวของเข็มจะมีชิ้นส่วนสำคัญที่มีลักษณะเป็นสี่เหลี่ยมเรียกว่า Slider ดังรูปที่ 1 ช่วยในการถ่ายเทประจุไฟฟ้าซึ่งได้เร็วที่สุดเมื่อยิ่งใกล้แผ่นแม่เหล็กจะทำให้ประสิทธิภาพสูงขึ้น แต่ประสิทธิภาพอาจจะตกลงเมื่อมีสิ่งเจือปนบนหัวอ่าน ยกตัวอย่างเช่น ฝุ่นละออง ขนาดเล็กที่ลอยอยู่ในอากาศ



รูปที่ 1 Head Gimbal Assembly Slider

(ที่มา : Disk read-and-write head - Wikipedia)

จากปัญหาเรื่องฝุ่นละอองที่กล่าวมาข้างต้นที่อาจปะปนบนชิ้นงาน ซึ่งจะส่งผลให้ประสิทธิภาพของหัวอ่านมีค่าไม่เป็นไปมาตามมาตรฐาน จึงต้องมีการตรวจสอบก่อนที่จะนำไปผ่านขั้นตอนต่อไปในกระบวนการผลิต แต่ด้วยจำนวนของชิ้นงานที่ต้องตรวจสอบมีปริมาณมาก จึงทำให้เกิดปัญหาความถี่ถ้วนของมนุษย์ที่เกิดจากการใช้สายตาเป็นเวลานานในการตรวจสอบ ยกตัวอย่างเช่น งานวิจัยความสัมพันธ์ระหว่างความล้าของสายตากับการตรวจสมรรถภาพทางสายตา ในกลุ่มผู้ใช้งานคอมพิวเตอร์ในโรงพยาบาลอุตรดิสถ์ (Charoon Chitnayee, Wirungrong Charuchart, Sasithorn Chitnayee (2013))

ดังนั้นคณะผู้วิจัยจึงมีแนวคิดพัฒนาระบบตรวจจับสิ่งเจือปนด้วยการประมวลผลภาพขึ้นมา โดยใช้ภาษา Python และไลบารี่ OpenCV เพื่อแก้ไขปัญหาการตรวจสอบผิดพลาดจากสายตามนุษย์เพียงอย่างเดียวจากการควบคุมคุณภาพผลิตภัณฑ์ในแบบเก่า ด้วยการตรวจจับสิ่งเจือปนขนาดเล็กไว้ก่อนแล้วในมนุษย์ยืนยันซ้ำเพื่อความถูกต้องและช่วยลดภาระจากปริณานชิ้นงานและลดข้อผิลพลาดที่อาจเกิดจากการใช้สายตาเป็นเวลานาน โดยงานวิจัยนี้จะสามารถแยกรูปภาพออกเป็น 2 ประเภท ได้แก่รูปที่มีสิ่งเจือปน กับรูปที่ไม่มีสิ่งเจือปน และเพื่อความสะดวกต่อการตรวจสอบในปริมาณมากจึงมีการรายงานออกมาเป็นไฟล์ Excel และเพิ่มประสิทธิภาพการควบคุมคุณภาพผลิตภัณฑ์ โดย วิธีที่ใช้เป็นการทำ Preprocessing Image ก่อนที่จะนำมาตรวจจับเส้นขอบวัตถุที่มีขนาดเล็ก

2. วัตถุประสงค์

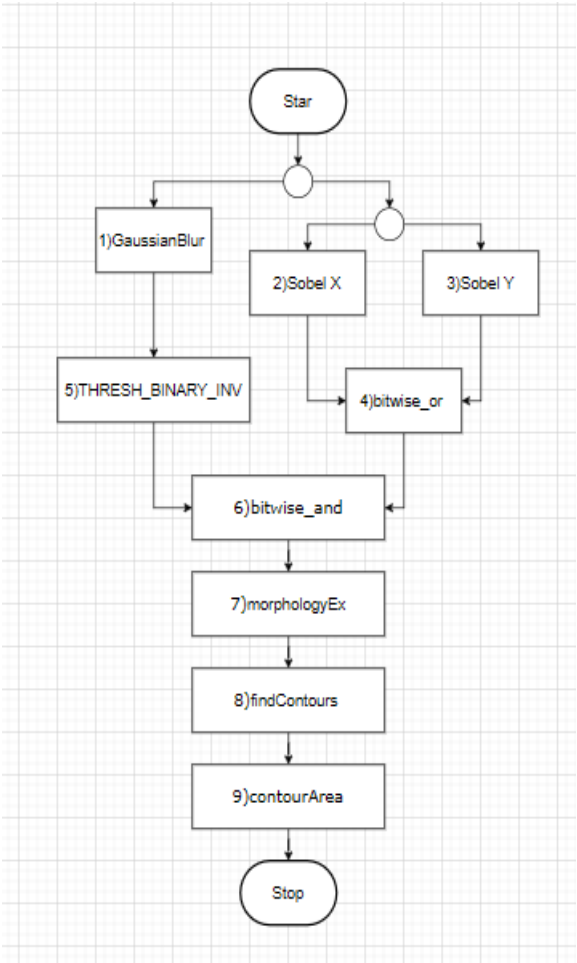
2.1. เพื่อพัฒนาอัลกอริทึมตรวจจับวัตถุในรูปภาพวัตถุปนเปื้อนบน Head Gimbal Assembly ด้วยภาษา Python ในการช่วยเหลือพนักงานควบคุมคุณภาพผลิตภัณฑ์

3. วิธีดำเนินการ

3.1. เครื่องมือที่ใช้ในการศึกษา

1. ฮาร์ดแวร์ที่ใช้ในการพัฒนาโครงงาน
   - คอมพิวเตอร์โน้ตบุ๊ก Dell Inc. Latitude 3400 (สเปคคอมของบริษัทจึงไม่สามารถเปิดเผยได้)
2. ซอฟต์แวร์ที่ใช้ในการพัฒนาโครงงาน
   - ระบบปฏิบัติการ Windows 10 Enterprise
   - โปรแกรมที่ใช้ในการพัฒนาระบบ Vis ual Studio Code
   - Command Prompt
3. ภาษาที่ใช้ในการพัฒนา
   - Python version 2.7

3.2. ขั้นตอนการทำงานของระบบ



1. GaussianBlur จะนำรูปภาพต้นฉบับมาเบลอเพื่อปรับความละเอียดของพื้นผิวของวัตถุภายในภาพ

2. Sobel หาเส้นขอบของวัตถุในภาพตามแนวแกน X

3. Sobel หาเส้นขอบของวัตถุในภาพตามแนวแกน Y

4. รวมผลลัพธ์ของรูปเข้าด้วยกันหลักการทำงานคล้ายคลึงกับ Logic Gate OR

5. ใช้ในการหาพื้นผิดในความสูงที่กำหนดไว้

6. รวมผลลัพธ์ของรูปเข้าด้วยกันหลักการทำงานคล้ายคลึงกับ Logic Gate AND

7. ปรับขนาดจุดให้มีความใหญ่ขึ้นและเข็มขึ้น

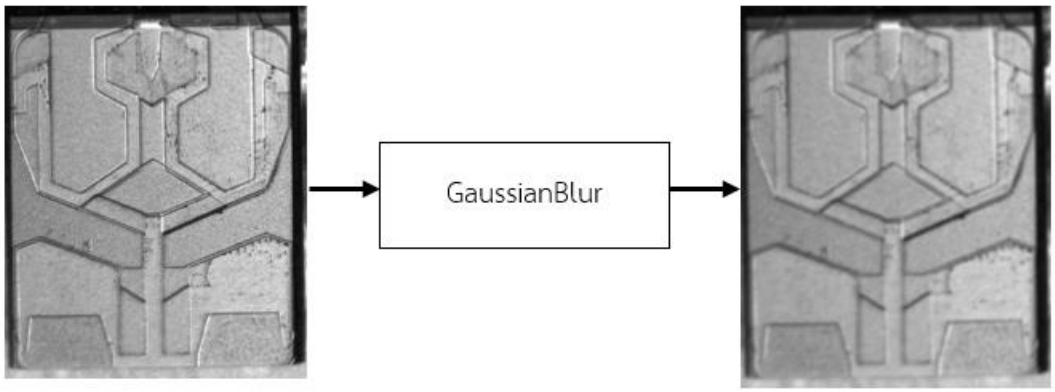
8. มองภาพเป็นจุดพิกเซล

9. นับจำนวนจุดพิกเซลที่อยู่ติดกันเพื่อหาสิ่งเจือปน

รูปที่ 2 Flowchart Diagram

3.3. ขั้นตอนการวิจัย

<u>ขั้นตอนที่ 1</u>

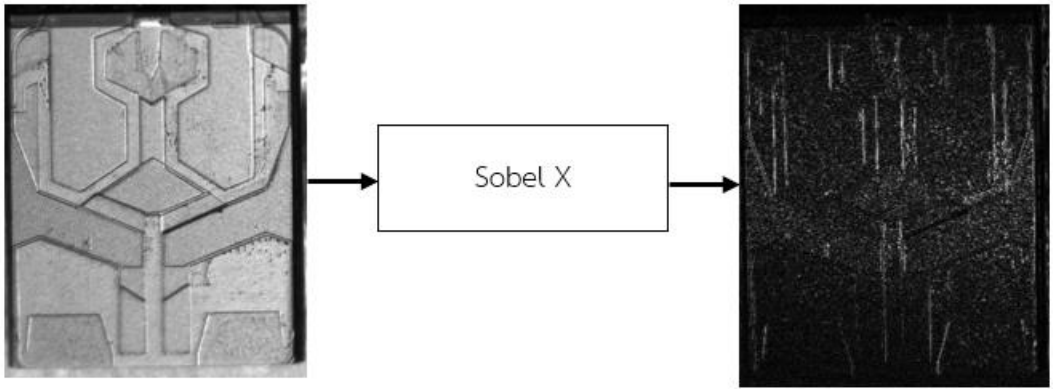


รูปที่ 3 GaussianBlur

จากรูปที่ 3 จะเห็นได้ว่าตัวกรอง GaussianBlur จะทำการเบลอภาพต้นฉบับเพื่อปรับความละเอียดของพื้นผิวของวัตถุภายในภาพ

คุณสมบัติ: การใช้ฟิลเตอร์ Gaussian blur จะทำให้ภาพเบลอลง, โดยมีการแปรผลของค่าพิกเซลรอบ ๆ พิกเซลที่กำหนดโดยการกรอง Gaussian. มีความสามารถในการลบรายละเอียดที่เล็กน้อยและเสียแต่งการในภาพ.

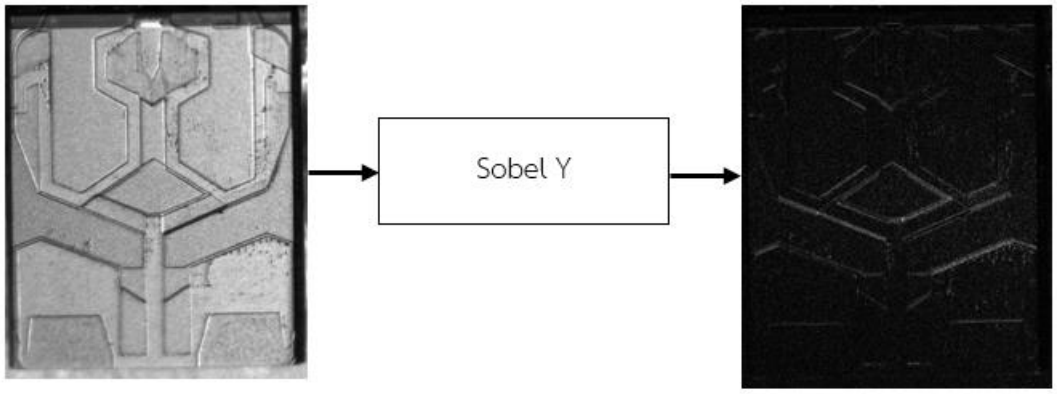
<u>ขั้นตอนที่ 2</u>



รูปที่ 4 Sobel X

จากรูปที่ 4 คือผลลัพธ์ของการใช้ Sobel หาเส้นขอบของวัตถุในภาพตามแนวแกน X โดยการนำรูปภาพต้นฉบับมาประมวลผล

<u>ขั้นตอนที่ 3</u>

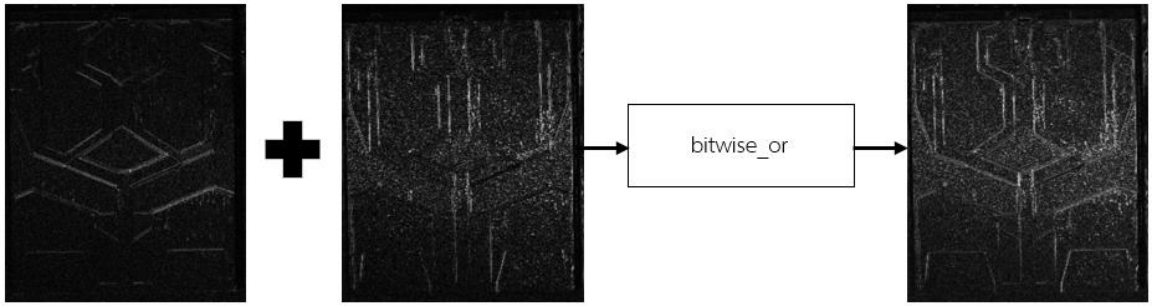


รูปที่ 5 Sobel Y

จากรูปที่ 5 คือผลลัพธ์ของการใช้ Sobel หาเส้นขอบของวัตถุในภาพตามแนวแกน Y โดยการนำรูปภาพต้นฉบับมาประมวลผล

คุณสมบัติ: ฟิลเตอร์ Sobel ใช้ในการตรวจจับขอบ (edge detection) ในภาพ. มักนำมาใช้ในการหาความเข้มของการเปลี่ยนแปลงของความเข้มของพิกเซลในทิศทางตามแนวนอนและแนวตั้ง

<u>ขั้นตอนที่ 4</u>



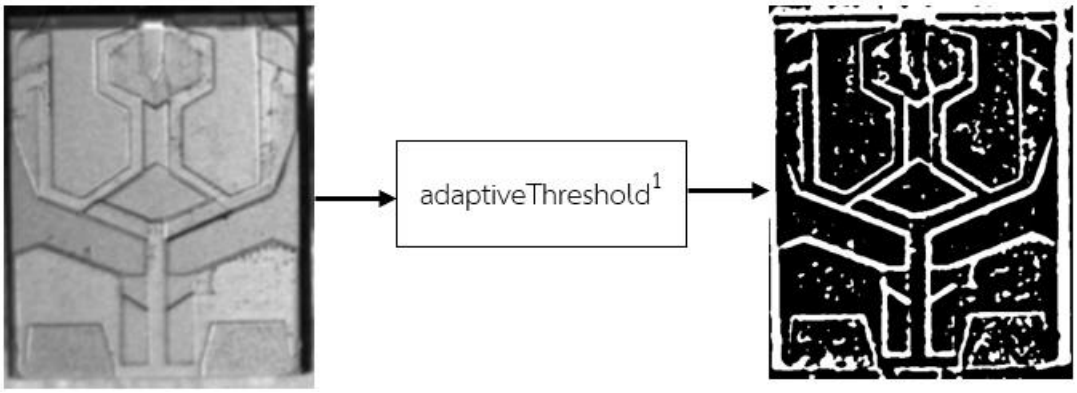
รูปที่ 6 Bitwise OR

จากรูปที่ 6 คือการรวมผลลัพธ์ขั้นตอนที่ 3 และ 4 เข้าด้วยกันด้วยคำสั่ง bitwise_or ที่มีหลักการทำงานคล้ายคลึงกับ Logic Gate OR

โดยทฤษฎี OR Gate มีสมการพิชคณิตลอจิกคือ $Y = f(A,B) = A + B$

Input		Output
A	B	f(A,B) = A+B
0	0	0
0	1	1
1	0	1
1	1	1

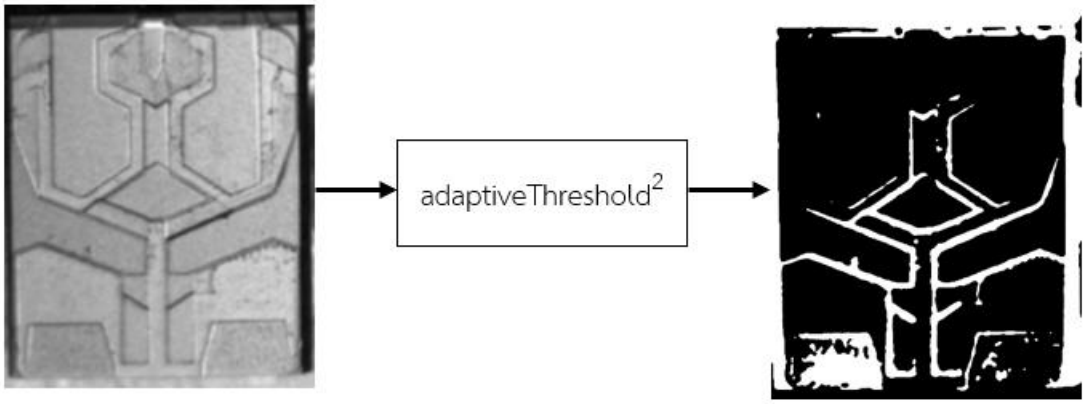
<u>ขั้นตอนที่ 5</u>



รูปที่ 7 cv2.THRESH_BINARY_INV (101, 5)

จากรูปที่ 7 เกิดจากการใช้คำสั่ง THRESH_BINARY_INV โดยมีการใช้ข้อมูลรูปภาพที่มีการผ่านตัวกรอง GaussianBlur มาแล้ว คำสั่งจะมีการปรับค่าพิกเซลข้อมูลปลายทางเป็น 0 กับ 255 และมีค่าตัวแปรในการปรับความสูงของพื้นผิวที่ต้องการด้วยค่าตัวเลข 101 และ 5

<u>ขั้นตอนที่ 6</u>



รูปที่ 8 cv2.THRESH_BINARY_INV (1001, 10)

จากรูปที่ 8 เกิดจากการใช้คำสั่ง THRESH_BINARY_INV โดยมีการใช้ข้อมูลรูปภาพที่มีการผ่านตัวกรอง GaussianBlur มาแล้ว คำสั่งจะมีการปรับค่าพิกเซลข้อมูลปลายทางเป็น 0 กับ 255 และมีค่าตัวแปรในการปรับความสูงของพื้นผิวที่ต้องการด้วยค่าตัวเลข 1001 และ 10

THRESH_BINARY_INV เป็นการกำหนดวิธีการที่ใช้ใน adaptive thresholding เพื่อแปลงภาพเป็นภาพขาว-ดำ (binary image) โดยให้ค่าพิกเซลที่มีความเข้มของภาพต่ำกว่าค่าที่กำหนดมีค่าเป็นสีขาว (1) และค่าพิกเซลที่มีความเข้มของภาพสูงกว่าค่าที่กำหนดมีค่าเป็นสีดำ (0).

โดยค่าตัวเลข 101,5 และ 1001,10 ได้มาจากการทดลองเพิ่มค่าและลดค่าตัวเลขทั้งสองจนกระทั่งได้ข้อมูลรูปภาพที่ต้องการที่จะนำไปใช้ต่อในส่วนต่อไป

คุณสมบัติ: Adaptive thresholding ใช้เพื่อแปลงภาพให้กลายเป็นภาพขาว-ดำ (binary image) โดยให้ค่าความเข้มของแต่ละพิกเซลมีการกำหนดค่าโค้งและค่าปรับให้เหมาะสมกับความเข้มของพิกเซลในบริเวณนั้น ๆ.

<u>ขั้นตอนที่ 7</u>

คือการรวมผลลัพธ์ของรูปที่ 7 และรูปที่ 6 เข้าด้วยกันด้วยคำสั่ง bitwise_and ที่มีหลักการทำงานคล้ายคลึงกับ Logic Gate AND

<u>ขั้นตอนที่ 8</u>

คือการรวมผลลัพธ์ของรูปที่ 8 และรูปที่ 6 เข้าด้วยกันด้วยคำสั่ง bitwise_and ที่มีหลักการทำงานคล้ายคลึงกับ Logic Gate AND

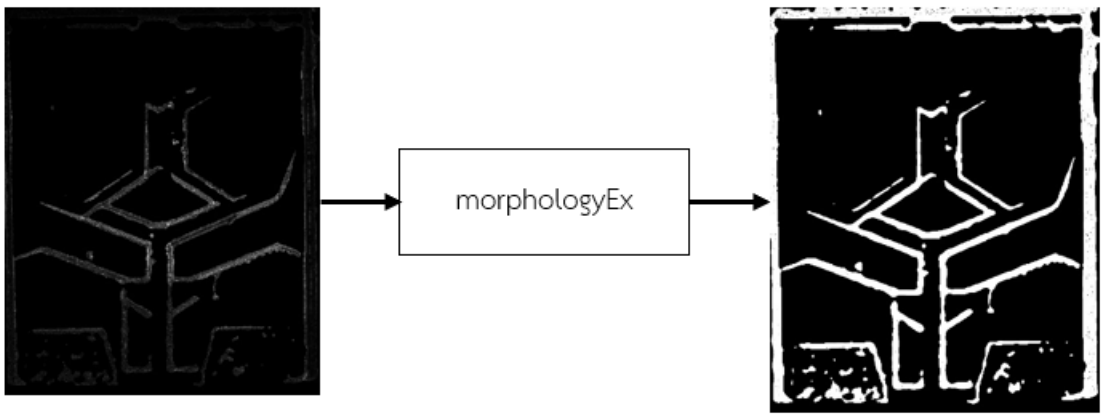
<u>ขั้นตอนที่ 9</u>

คือการรวมผลลัพธ์ของขั้นตอนที่ 7 และ ผลลัพธ์ของขั้นตอนที่ 8 เข้าด้วยกันด้วยคำสั่ง bitwise_and ที่มีหลักการทำงานคล้ายคลึงกับ Logic Gate AND จะเห็นได้ว่าข้อมูลภาพที่ได้รับมาจะเห็นเฉพาะเส้นขอบของตัวชิ้นงานส่วนใหญ่และเส้นขอบของฝุ่นที่เจือปนอยู่บนพื้นผิวของตัวชิ้นงาน

โดยทฤษฎี AND Gate มีสมการพิชคณิตลอจิกคือ $Y = f(A,B) = A * B$

Input		Output
A	B	f(A,B) = A*B
0	0	0
0	1	0
1	0	0
1	1	1

<u>ขั้นตอนที่ 10</u>

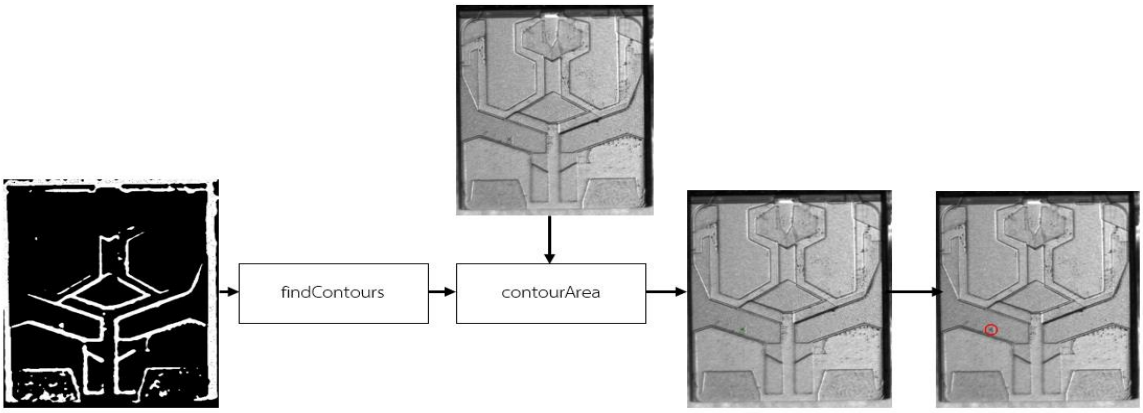


รูปที่ 9 Reprocess

จากรูปที่ 9 คือผลลัพธ์ของการทำการปรับปรุงคุณภาพด้วยการปรับขนาดพิกเซลด้วย cv2.morphologyEx และทำปรับค่าพิกเซลให้มีค่าตั้งแต่ 1 ถึง 255 เป็น 255 ทั้งหมด เพื่อเป็นการขยายพิกเซล

คุณสมบัติ: Morphological transformations ใช้ในการปรับปรุงรูปร่างของวัตถุในภาพ, เช่น การขยาย, การหด, การเปิด, และการปิด. มักนำมาใช้ในการลบขอบเสียง (noise) หรือการปรับปรุงโครงสร้างของวัตถุในภาพ.

<u>ขั้นตอนที่ 11</u>

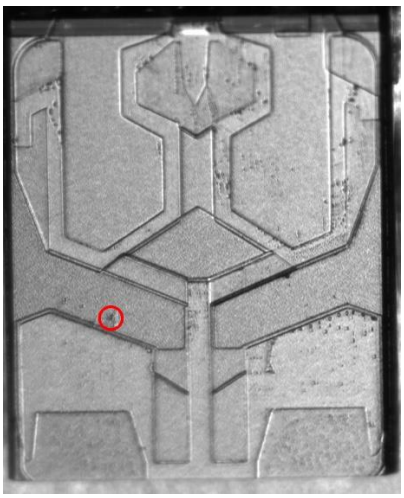


รูปที่ 10 ตรวจจับสิ่งเจือปน

จากรูปที่ 10 คือการใช้คำสั่งในการค้นหาวัตถุที่มีขนาดเล็กด้วย โดยส่ง ผลลัพธ์ของขั้นตอนที่ 10 เข้ามาทำงาน โดยกำหนดให้ ส่งรูปร่างมาโดยมาในรูปแบบวงกลม ด้วยคำสั่ง cv2.circle(img,center,radius,(0,0,255),4)ในการวงกลมวัตถุลงไปทับบนรูปภาพที่ทำการระบุไว้ โดยตัวแปร img คือ รูปต้นฉบับ การวงกลมสีแดงจากการระบุค่าตัวแปรเป็น (0,0,255) วัตถุที่มีจำนวนพิกเซลอยู่ในช่วงที่เหมาะสมสำหรับภาพแต่ละชนิด เช่นตัวอย่างนี้จะใช้เป็น 200 ถึง 230

3.4. ตัวอย่างผลลัพธ์

3.4.1.ตัวอย่างผลลัพธ์ True Positive



รูปที่ 11 ตัวอย่างผลลัพธ์ True Positive

จากรูปที่ 11 จะเห็นได้ว่าระบบสามารถระบุสิ่งเจือปนที่อยู่บนตัวชิ้นงานได้อย่างถูกต้อง โดยทำการใส่วงกลมสีแดงลงในรูปภาพบริเวณที่มีสิ่งเจือปนอยู่

3.4.2.ตัวอย่างผลลัพธ์ False Positive

รูปที่ 12 ตัวอย่างผลลัพธ์ False Positive

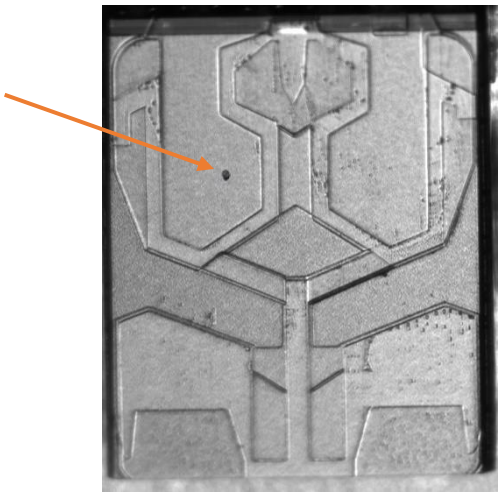
จากรูปที่ 12 จะเห็นได้ว่าระบบไม่สามารถระบุสิ่งเจือปนที่อยู่บนตัวชิ้นงานได้อย่างถูกต้อง โดยทำการใส่วงกลมสีแดงลงในรูปภาพบริเวณที่ไม่มีสิ่งเจือปนอยู่

3.4.3. ตัวอย่างผลลัพธ์ True Negative

รูปที่ 13 ตัวอย่างผลลัพธ์ True Negative

จากรูปที่ 13 จะเห็นได้ว่าระบบสามารถระบุความสะอาดที่อยู่บนตัวชิ้นงานได้อย่างถูกต้อง โดยที่ไม่ใส่วงกลมสีแดงลงในรูปภาพบริเวณใดๆ

3.4.4.ตัวอย่างผลลัพธ์ False Negative



รูปที่ 14 ตัวอย่างผลลัพธ์ False Negative

จากรูปที่ 14 จะเห็นได้ว่าระบบไม่สามารถระบุความสะอาดที่อยู่บนตัวชิ้นงานได้อย่างถูกต้อง โดยที่ไม่ใส่วงกลมสีแดงลงในรูปภาพบริเวณที่มีสิ่งเจือปน

4. ผลการทดลอง

ในการทดลองจะมีการใช้ชุดตัวอย่างรูปภาพทั้งหมด 4 ชุด ในแต่ละชุดมีรูปภาพทั้งหมด 20 รูป และมีการทำซ้ำทั้งหมด 3 ครั้ง ต่อ 1 ชุดตัวอย่างเพื่อความแม่นยำในผลของการทดลองกับแต่ละชุดตัวอย่าง รวมรูปทั้งหมดที่ใช้ในการทดสอบเป็นจำนวน 240 รูป

- True Positive (TP) คือ โปรแกรมที่ทำนายว่าจริง และคนบอกว่าเป็นจริง
- True Negative (TN) คือ โปรแกรมที่ทำนายว่าไม่เป็นจริง และคนบอกว่าไม่เป็นจริง
- False Positive (FP) คือ โปรแกรมที่ทำนายว่าจริง และคนบอกว่าไม่เป็นจริง
- False Negative (FN) คือ โปรแกรมที่ทำนายว่าไม่เป็นจริง และคนบอกว่าเป็นจริง
- Accuracy (ACC) คือ ค่าการทำนายที่ถูกต้อง

ตารางที่ 1 Confusion Matrix ชุดอย่างที่ 1 ทั้ง 3 ครั้ง

N = 60	Predicted Negative	Predicted Positive	
Actual Negative	TN = 51	FP = 3	54
Actual Positive	FN = 3	TP = 3	6
	54	6	ACC = 0.9

ตารางที่ 2 Confusion Matrix ชุดอย่างที่ 2 ทั้ง 3 ครั้ง

N = 60	Predicted Negative	Predicted Positive	
Actual Negative	TN = 51	FP = 6	57
Actual Positive	FN = 3	TP = 0	3
	54	6	ACC = 0.85

ตารางที่ 3 Confusion Matrix ชุดอย่างที่ 3 ทั้ง 3 ครั้ง

N = 60	Predicted Negative	Predicted Positive	
Actual Negative	TN = 48	FP = 6	54
Actual Positive	FN = 6	TP = 0	6
	54	6	ACC = 0.8

ตารางที่ 4 Confusion Matrix ชุดอย่างที่ 4 ทั้ง 3 ครั้ง

N = 60	Predicted Negative	Predicted Positive	
Actual Negative	TN = 48	FP = 3	51
Actual Positive	FN = 3	TP = 6	9
	51	9	ACC = 0.9

ตารางที่ 5 Confusion Matrix รวมทั้ง 4 ชุดตัวอย่างและทั้ง 3 ครั้ง

N = 240	Predicted Negative	Predicted Positive	
Actual 0	TN = 198	FP = 18	216
Actual 1	FN = 15	TP = 9	24
	213	27	ACC = 0.86

ตารางที่ 6 ตารางสรุป Confusion Matrix

Measure		**Calculated Value**
Error rate	ERR	0.1375
Accuracy	ACC	0.8625
Sensitivity True positive rate Recall	SN TPR REC	0.3750
Specificity True negative rate	SP TNR	0.9167
Precision Positive predictive value	PREC PPV	0.3333
False positive rate	FPR	0.0833

สูตรคำนวณที่ใช้

1.Error rate (ERR)

ERR = (FP+FN)/(TP+TN+FN+FP)

2.Accuracy (ACC)

ACC = (TP+TN)/(TP+TN+FN+FP)

3. Sensitivity/True positive rate/Recall (SN)

SN = TP/(TP+FN)

4. Specificity/True negative rate (SP)

SP = TN/(TN+FP)

5. Precision/Positive predictive value (PREC)

PREC = TP/(TP+FP)

6. False positive rate (FPR)

FPR = FP/(TN+FP)

5. สรุปผลและอภิปรายผล

ระบบตรวจจับการสิ่งเจือปนบนวัตถุด้วยการประมวลผลภาพ ถูกพัฒนามาเพื่อให้บริษัทผลิตสายฮาร์ดไดร์ นำไปช่วยหรือพนักงานในการตรวจหาสิ่งเจือปนบนชิ้นงานก่อนที่จะนำไปทศสอบในเครื่องทำสอบของทางบริษัท โดยที่ระบบจะช่วยแก้ปัญหาเรื่องระยะเวลาในการตรวจสอบสิ่งเจือปนบนชิ้นงาน เนื่องจากจำนวนชิ้นงานต่อชุดมีจำนวนมากและต้องการความเร็วในสายการผลิต ด้วยการตรวจจับสิ่งเจือปนบนชิ้นการจากรูปถ่ายที่ติดอยู่บนสายการผลิต และจะแจ้งเตือนข้อมูลโดยสรุปต่อชุดเพื่อนำไปเขียนในรหัสชิ้นงาน ทั้งนี้ตัวระบบได้ผ่านการทดสอบจากชุดตัวอย่างทั้งหมด 4 ชุด โดยชุดละ 3 ครั้ง และได้ค่าต่างๆมาดังนี้ True Positive อยู่ที่ 9 True Negative อยู่ที่ 198 False Positive อยู่ที่ 18 และ False Negative อยู่ที่ 15 จากจำนวนทั้งหมด 240 รูป

หลังจากสรุปข้อมูลงานวิจัยนี้ค่าความถูกต้องอยู่ที่คิดเป็นร้อยละ 86.25 โดยที่ค่าความเฉพาะเจาะจงคิดเป็นร้อยละ 91.67 แสดงให้ทราบว่างานวิจัยมีความสามารถที่ตรวจความสะอาดบน Slider ของ Head Gimbal Assembly ค่อนข้างสูงและหวังว่างานวิจัยนี้จะสามารถนำไปต่อยอดใช้กับสิ่งต่างๆ ได้

True Positive Rate และ Precision ที่มีผลค่อนข้างต่ำ หมายถึงผลการตรวจจับสิ่งแปลกปลอมทำได้ต่ำ ซึ่งหากพิจารณาผลตาราง 2 และ 3 จะเห็นได้ว่าไม่สามารถตรวจจับได้ เนื่องจากมีกรณีที่สิ่งเจือปนบนอยู่บริเวณร่องของตัวชิ้นงานและมีสีใกล้เคียงกำตัวชิ้นงานมากจึงไม่สามารถแยกออกได้

6. กิตติกรรมประกาศ

ขอขอบคุณ บริษัทในสายการผลิตฮาร์ดไดร์ รวมทั้ง บุคลากรสาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้เชียงใหม่ ที่ให้การสนับสนุนสถานที่ทำงานวิจัยและให้ความรู้คำแนะนำเกี่ยวกับ Head Gimbal Assembly และ image processing

7. เอกสารอ้างอิง

สารัช ตันติกิตติ. (2016,22 July). **Image Processing for Detecting Dengue Virus from WBC[วิทยานิพนธ์ปริญญามหาฎีบัณฑิต,มหาวิทยาลัยสยาม].** แหล่งที่มา : การประมวลผลภาพเพื่อบ่งชี้โรคไข้เลือดออกเดงกีจากเม็ดเลือดขาว (siam.edu)

Anisha Swain.(2561,3 กันยายน). **Noise filtering in Digital Image Processing. Medium.** แหล่งที่มา : Noise filtering in Digital Image Processing | by Anisha Swain | Image Vision | Medium

Charoon Chitnayee, Wirungrong Charuchart, Sasithorn Chitnayee (2013). **งานวิจัยความสัมพันธ์ระหว่างความล้าของสายตากับการตรวจสมรรถภาพทางสายตา ในกลุ่มผู้ใช้งานคอมพิวเตอร์ในโรงพยาบาลอุตรดิสถ์**
แหล่งที่มา : sunsanee,+Journal+manager,+18346-39617-1-CE.pdf

Geeksforgeeks. (2021,18 Aug). **Python | Create and write on excel file using xlsxwriter module**. แหล่งที่มา : Python | Create and write on excel file using xlsxwriter module - GeeksforGeeks

Shawn Gomez. (2014). **scikit-image: image processing in Python.**
แหล่งที่มา : https://doi.org/10.7717/peerj.453