

ระบบการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ในกระบวนการประกอบ  
ฮาร์ดดิสก์ไดรฟ์อัตโนมัติด้วยเทคนิคการเรียนรู้เชิงลึก



นายอนุภาค ปัญญาทนต์สมบัติ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมเมคคาทรอนิกส์  
มหาวิทยาลัยเทคโนโลยีสุรนารี  
ปีการศึกษา 2564

VISUAL INSPECTION SYSTEM OF THE HARD DISK DRIVE  
COMPONENTS IN FULLY AUTOMATED ASSEMBLY LINE  
USING BY DEEP LEARNING TECHNIQUE



A Thesis Submitted in Partial Fulfillment of the Requirements for the  
Degree of Master of Engineering in Mechatronics Engineering  
Suranaree University of Technology  
Academic Year 2021

ระบบการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ในกระบวนการประกอบ  
ฮาร์ดดิสก์ไดรฟ์อัตโนมัติด้วยเทคนิคการเรียนรู้เชิงลึก

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้บัณฑิตวิทยาลัยฉบับนี้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์



(ผศ. ดร.ศโรภา แซ่เงงการ)

ประธานกรรมการ



(รศ. ดร.จิระพล ศรีเสรีธผล)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



(ผศ. ดร.ชัยยุทธ์ สัมภาวะคุปต์)

กรรมการ



(รศ. ดร.ฉัตรชัย โชติษฐยางกูร)

รองอธิการบดีฝ่ายวิชาการและประกันคุณภาพ



(รศ. ดร.พรศิริ จงกล)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

อนุภาค ปัญญาทวนสมบัติน : ระบบการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์  
ในกระบวนการประกอบฮาร์ดดิสก์ไดรฟ์อัตโนมัติด้วยเทคนิคการเรียนรู้เชิงลึก  
(VISUAL INSPECTION SYSTEM OF THE HARD DISK DRIVE COMPONENTS IN  
FULLY AUTOMATED ASSEMBLY LINE USING BY DEEP LEARNING TECHNIQUE)  
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.จิระพล ศรีเสริฐผล, 110 หน้า.

คำสำคัญ: การประมวลผลภาพ/การเรียนรู้ของเครื่องจักร/การเรียนรู้เชิงลึก/การตรวจจับวัตถุ/  
องค์ประกอบของฮาร์ดดิสก์ไดรฟ์

อุปกรณ์จัดเก็บข้อมูลดิจิทัล (Digital data) มีความสำคัญเป็นอย่างมากในยุคปัจจุบันและ  
อุปกรณ์จัดเก็บข้อมูลดิจิทัลในปัจจุบันมีหลากหลายชนิดมากซึ่งฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive :  
HDD) เป็นหนึ่งในอุปกรณ์จัดเก็บข้อมูลดิจิทัลที่สำคัญเนื่องจากมีพื้นที่ในการจัดเก็บข้อมูลจำนวนมาก  
ต่อหนึ่งหน่วยดังนั้นหนึ่งในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ที่สำคัญคือกระบวนการประกอบ  
ฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive Assembly : HDA) เป็นกระบวนการที่นำชิ้นส่วนต่างๆของฮาร์ดดิสก์  
ไดรฟ์มาประกอบรวมกันให้เป็นฮาร์ดดิสก์ไดรฟ์ ดังนั้นการตรวจสอบความถูกต้องของตำแหน่ง  
องค์ประกอบต่างๆภายในฮาร์ดดิสก์ไดรฟ์ในกระบวนการผลิตเป็นปัจจัยสำคัญอย่างหนึ่งเพื่อให้แน่ใจ  
ว่าผลิตภัณฑ์ไม่มีความผิดปกติขององค์ประกอบต่างๆภายในฮาร์ดดิสก์ไดรฟ์ระบบการตรวจสอบความ  
ถูกต้องขององค์ประกอบต่างๆที่ใช้ในกระบวนการผลิตโดยองค์ประกอบต่างๆของฮาร์ดดิสก์  
ไดรฟ์(Hard Disk Drive Elements : HDE) ซึ่งในปัจจุบันคือระบบการมองเห็นของเครื่องจักร  
(Machine vision systems) โดยใช้เทคนิคต่างๆของการประมวลผลภาพ (Image processing) ซึ่งมี  
ข้อจำกัดเกี่ยวกับขอบเขตการมองเห็น (Field Of View : FOV) และยังมีความยืดหยุ่นในการใช้งาน  
น้อย

ดังนั้น ในงานวิจัยนี้จึงได้นำเสนอเกี่ยวกับเทคนิคการเรียนรู้เชิงลึก (Deep Learning : DL)  
ถือเป็นหนึ่งในกระบวนการเรียนรู้ของเครื่องจักร (Machine Learning : ML) โดยมีโครงสร้างและการ  
ประมวลผลคล้ายกับสมองของมนุษย์โดยเรียกว่าโครงข่ายประสาทเทียม (Artificial Neural Network  
: ANN) ในงานวิจัยนี้จะนำเสนอเกี่ยวกับโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural  
Network : CNN) กับโวลเวอร์ชันห้าอัลกอริทึม (YOLOv5 Algorithm) เป็นหนึ่งในอัลกอริทึมสำคัญ  
สำหรับการตรวจจับวัตถุอัตโนมัติซึ่งมีประสิทธิภาพสูงและตรวจจับวัตถุได้อย่างรวดเร็วและมีความ  
ยืดหยุ่นสูง สำหรับระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ที่ทำการพัฒนาขึ้นมาจะทำการ  
ตรวจสอบองค์ประกอบในส่วนของมุมมองด้านบนของฮาร์ดดิสก์ไดรฟ์โดยจะมีองค์ประกอบทั้งหมด

16 ตำแหน่งสำหรับการตรวจสอบ ประสิทธิภาพของระบบการตรวจสอบองค์ประกอบของฮาร์ดดิสก์  
 ไดรฟ์ที่ได้ออกแบบมาใหม่นั้นมีเปอร์เซ็นต์ความถูกต้องในการตรวจสอบอยู่ที่ 99.49%



สาขาวิชา วิศวกรรมเมคคาทรอนิกส์  
 ปีการศึกษา 2564

ลายมือชื่อนักศึกษา Amk Pak  
 ลายมือชื่ออาจารย์ที่ปรึกษา [Signature]

ANUPAK PUNYATHANSOMBAT : VISUAL INSPECTION SYSTEM OF THE HARD DISK DRIVE COMPONENTS IN FULLY AUTOMATED ASSEMBLY LINE USING BY DEEP LEARNING TECHNIQUE. THESIS ADVISOR : ASSOC. PROF. JIRAPHON SRISERTPHOL, Ph.D., 110 PP.

Keyword: Image processing/Machine learning/Deep learning/Object detection/Hard disk drive components

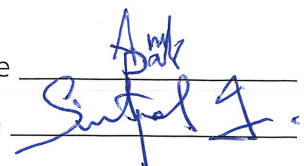
A hard disk drive is a very important data storage device in the era of big data. This thesis focuses on developing the visual inspection system of the hard disk drive components which is an important step in the hard disk drive assembly (HDA) process. The interesting problem with this procedure is the overkill of inspection. There are 17 positions of hard disk drive components such as platters, Head Stack Assembly (HSA), RAMP, Filter, etc. that are checked. The over rejection cause is the hard disk drive image out of the field of view (FOV) perhaps mistakenly checked from the machine vision system which is the current object detection system without the flexibility of label image. This causes a delay in the HDA process. Therefore, this problem has significant implications for the amount of HDA quality.

Therefore, in this research, deep learning (DL) techniques are presented as a machine learning (ML) tool that has a structure and processing similar to the human brain. which is called a Artificial Neural Network (ANN). The visual inspection system has been redesigned using convolutional neural networks (CNN) and YOLOv5 algorithm it is one of the key algorithms for automatic object detection with high efficiency and flexible. The performance of the newly designed hard disk drive component inspection system has a validity percentage of 99.49%

School of Mechatronic Engineering  
Academic Year 2021

Student's Signature \_\_\_\_\_

Advisor's Signature \_\_\_\_\_

The student's signature is written in blue ink above the line. The advisor's signature is written in blue ink below the line.

## กิตติกรรมประกาศ

ผู้วิจัยขอขอบพระคุณบุคคลต่าง ๆ ที่ช่วยเหลือทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงตามวัตถุประสงค์ทุกประการ ที่ให้คำแนะนำ ปรึกษา ทั้งด้านการดำเนินการวิจัยและวิชาการ ดังนี้

ขอขอบพระคุณ รองศาสตราจารย์ ดร.จิระพล ศรีเสริฐผล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ได้ให้ความช่วยเหลืออย่างยิ่ง ให้คำแนะนำ คำปรึกษา ในการดำเนินงานวิจัย คอยชี้แนะในแต่ละขั้นตอนรวมถึงให้ความรู้ ด้านวิชาการ การเขียนเล่มวิทยานิพนธ์ ด้านกระบวนการคิดและดำเนินชีวิต และยังช่วยตรวจสอบแก้ไขวิทยานิพนธ์ให้มีความถูกต้องและสมบูรณ์

ขอขอบพระคุณ รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์ อาจารย์ ดร.สุรเดช ตัญญูตรัยรัตน์ อาจารย์ ดร.อุเทน ลีตน ที่ให้ความรู้ด้านวิชาการซึ่งสามารถนำไปประยุกต์ใช้กับงานวิจัย ทำให้วิทยานิพนธ์มีความถูกต้องสมบูรณ์

ขอขอบพระคุณมหาวิทยาลัยเทคโนโลยีสุรนารี ที่ได้มอบทุนการศึกษาในระดับ มหาบัณฑิต ศึกษาแก่ผู้วิจัย และช่วยเหลือด้านเครื่องมืออุปกรณ์ต่าง ๆ ที่จำเป็นต่องานวิจัย

ขอขอบพระคุณ บริษัท เวสเทิร์น ดิจิตอล (ประเทศไทย) จำกัด ที่ได้ช่วยเหลืองานวิจัยนี้ให้สำเร็จลุล่วง สำหรับข้อมูลงานวิจัยและความรู้ด้านอุตสาหกรรมต่าง ๆ และช่วยเหลือผู้วิจัยให้มีประสบการณ์การทำงานที่บริษัท

ขอขอบพระคุณ คุณศิริพงษ์ ปะวะโก ที่ช่วยเหลือให้ความรู้และวิธีดำเนินการวิจัย ให้สำเร็จลุล่วงตามวัตถุประสงค์ทุกประการ

ขอขอบพระคุณ คุณประธาน ชมเมืองปัก คุณจักรวรรดิ ดียิ่ง คุณภาณุวัฒน์ รอดชม รวมถึงบุคลากรในแผนก Assembly and automation development และ แผนก AME บริษัท เวสเทิร์น ดิจิตอล (ประเทศไทย) จำกัด

ขอขอบพระคุณ เจ้าหน้าที่ประจำสำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ทุก ๆ ท่าน ที่ให้ความช่วยเหลือและอำนวยความสะดวกในด้านเอกสารและดำเนินการต่าง ๆ สำหรับวิทยานิพนธ์เพื่อให้สำเร็จลุล่วง

สุดท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณมารดา และสมาชิกครอบครัวของผู้วิจัยที่คอยช่วยเหลือสนับสนุน อุปการะเลี้ยงดู อบรมสั่งสอนให้เป็นคนดี ตลอดจนส่งเสริมด้านการศึกษา และคอยให้กำลังใจจนกระทั่งวิทยานิพนธ์ฉบับนี้สำเร็จ

อนุภาค ปัญญาชนสมบัติ

# สารบัญ

หน้า

|   |    |
|---|----|
| บทคัดย่อ (ภาษาไทย) .....  | ก  |
| บทคัดย่อ (ภาษาอังกฤษ) .....                                       | ค  |
| กิตติกรรมประกาศ .....   | ง  |
| สารบัญ .....  | จ  |
| สารบัญตาราง .....   | ช  |
| สารบัญรูป .....   | ณ  |
| <b>บทที่</b>  |    |
| <b>1 บทนำ</b> .....   | 1  |
| 1.1 ที่มาและความสำคัญของปัญหาการวิจัย .....                       | 1  |
| 1.2 วัตถุประสงค์ของงานวิจัย.....                                  | 4  |
| 1.3 ขอบเขตของงานวิจัย .....                                       | 4  |
| 1.4 ระเบียบวิธีวิจัย .....  | 5  |
| 1.5 สถานที่ทำงานวิจัย.....  | 6  |
| 1.6 เครื่องมือที่ใช้ในการทำวิจัย.....                             | 6  |
| 1.7 ประโยชน์ที่คาดว่าจะได้รับ .....                               | 6  |
| 1.8 การจัดทำรูปเล่มวิทยานิพนธ์ .....                              | 6  |
| <b>2 ปรัชญาวรรณกรรมและงานวิจัยที่เกี่ยวข้อง</b> .....             | 8  |
| 2.1 ส่วนประกอบของฮาร์ดดิสก์ไดรฟ์ .....                            | 8  |
| 2.2 การประมวลผลภาพ (Image Processing).....                        | 9  |
| 2.2.1 พิกเซล (Pixel).....   | 9  |
| 2.2.2 การครอบภาพ (Crop image).....                                | 9  |
| 2.3 ปัญญาประดิษฐ์ (Artificial Intelligence : AI).....             | 10 |
| 2.4 การเรียนรู้ของเครื่องจักร (Machine Learning : ML) .....       | 12 |
| 2.5 โครงข่ายประสาทเทียม (Artificial Neural Networks : ANNs) ..... | 14 |
| 2.5.1 ส่วนประกอบของโครงข่ายประสาท (Neural Network) .....          | 15 |



## สารบัญ (ต่อ)

หน้า

|          |   |           |
|----------|---|-----------|
| 2.5.2    | ส่วนประกอบของโครงข่ายประสาทเทียม (ANN components).....                                  | 15        |
| 2.5.3    | หลักการการทำงานของโครงข่ายประสาทเทียม .....   | 16        |
| 2.6      | การเรียนรู้เชิงลึก (Deep Learning : DL) .....   | 19        |
| 2.7      | แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network : CNN).....        | 19        |
| 2.7.1    | หลักการการทำงานของโครงข่ายประสาทเทียมแบบคอนโวลูชัน.....                                 | 20        |
| 2.8      | การตรวจจับวัตถุ (Object detection).....   | 23        |
| 2.8.1    | Sliding Window.....   | 25        |
| 2.8.2    | การวัดผล .....  | 25        |
| 2.9      | YOLOv5 (You Only Look Once Ver.5) .....   | 27        |
| 2.9.1    | หลักการการทำงานของ YOLOv5 .....   | 29        |
| 2.10     | ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง .....  | 31        |
| 2.10.1   | งานวิจัยเกี่ยวกับโครงข่ายประสาทเทียม .....  | 31        |
| 2.10.2   | งานวิจัยที่เกี่ยวข้องกับการตรวจจับวัตถุภายในภาพ .....                                   | 32        |
| 2.11     | สรุป.....   | 34        |
| <b>3</b> | <b>วิธีการดำเนินงานวิจัย.....</b>   | <b>35</b> |
| 3.1      | บทนำ.....   | 35        |
| 3.2      | วิเคราะห์ข้อมูลและการเก็บรวบรวมข้อมูลภาพ HDE view .....                                 | 36        |
| 3.2.1    | ประเภทของภาพ HDE view.....  | 36        |
| 3.2.2    | ข้อมูลของภาพ HDE view.....  | 38        |
| 3.3      | วิธีการทดลอง .....  | 39        |
| 3.3.1    | การเก็บข้อมูล (Data collection).....  | 39        |
| 3.3.2    | การทำความสะอาดข้อมูล (Cleaning data).....   | 40        |
| 3.3.3    | เทคนิคการประมวลผลภาพเพิ่มลดความสว่างของภาพ (Image processing - adjust brightness) ..... | 40        |
| 3.3.4    | จัดประเภทของรูปภาพ (Categorize images).....   | 40        |

## สารบัญ (ต่อ)

หน้า

|          |  |           |
|----------|--|-----------|
| 3.3.5    | การออกแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์<br>ด้วยเทคนิคการเรียนรู้เชิงลึก .....     | 42        |
| 3.3.6    | สร้างโมเดล CNN (Create CNN model) .....  | 43        |
| 3.3.7    | การสร้างโมเดล YOLOv5 (Create YOLOv5 object detection) .....                                | 47        |
| 3.4      | สรุป.....  | 51        |
| <b>4</b> | <b>ผลการทดลอง และการวิเคราะห์ผลการทดลอง .....</b>  | <b>54</b> |
| 4.1      | บทนำ.....  | 54        |
| 4.2      | ผลการตรวจสอบภาพรวมด้วย CNN.....  | 54        |
| 4.3      | ผลการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วย YOLOv5.....                                   | 56        |
| 4.4      | ผลการทดสอบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (HDE view).....                          | 57        |
| 4.4.1    | ผลการทดสอบระบบ VI system กับข้อมูลชุดที่ 1.....  | 58        |
| 4.4.2    | ผลการทดสอบระบบ VI system กับข้อมูลชุดที่ 2.....  | 59        |
| 4.4.3    | ผลการทดสอบระบบ VI system กับข้อมูลชุดที่ 3.....  | 60        |
| 4.4.4    | ผลการทดสอบระบบ VI system กับข้อมูลชุดที่ 4.....  | 61        |
| 4.5      | สรุปผลการทดลอง.....  | 65        |
| <b>5</b> | <b>สรุปและข้อเสนอแนะ .....</b>   | <b>71</b> |
| 5.1      | สรุป.....  | 71        |
| 5.2      | ข้อเสนอแนะ.....  | 72        |
|          | รายการอ้างอิง.....   | 73        |
|          | ภาคผนวก  |           |
|          | ภาคผนวก ก. โค้ดโปรแกรมสำหรับการสร้างระบบตรวจสอบองค์ประกอบ<br>ฮาร์ดดิสก์ไดรฟ์ HDE view..... | 76        |
|          | ภาคผนวก ข. โค้ดระบบตรวจสอบองค์ประกอบฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิค<br>การเรียนรู้เชิงลึก.....   | 82        |
|          | ภาคผนวก ค. การใช้ระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ HDE view.....                     | 97        |
|          | ภาคผนวก ง. บทความที่ได้รับการตีพิมพ์เผยแพร่.....   | 101       |
|          | ประวัติผู้เขียน.....   | 110       |

## สารบัญตาราง

| ตารางที่ | หน้า  |
|----------|---|
| 1.1      | ตารางแสดงการตรวจสอบตำแหน่งผิดปกติของฮาร์ดดิสก์ไดรฟ์ Fail และ Overkill..... 2                |
| 2.1      | ตารางแสดงความแตกต่างระหว่าง computer vision กับ deep learning ..... 33                      |
| 2.2      | ตารางแสดงเกณฑ์การคัดเลือกระหว่าง computer vision กับ deep learning..... 34                  |
| 3.1      | ตารางแสดงการตรวจสอบตำแหน่งผิดปกติของฮาร์ดดิสก์ไดรฟ์ Fail และ Overkill..... 36               |
| 3.2      | ตารางเปรียบเทียบโมเดล CNN จากการฝึกสอนโมเดล ..... 44  |
| 3.3      | ตารางเปรียบเทียบการสร้างโมเดล CNN หลังจากครอปรูปภาพเพื่อฝึกสอนใหม่ ..... 47                 |
| 3.4      | ตารางแสดงผลการฝึกสอนโมเดล YOLOv5 ..... 50   |
| 4.1      | ตารางสรุปผลการทดสอบระบบตรวจสอบองค์ประกอบฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึก..... 70 |

## สารบัญรูป

| รูปที่ |  | หน้า |
|--------|--|------|
| 1.1    | ไลน์การผลิตฮาร์ดดิสก์ไดรฟ์ .....   | 2    |
| 1.2    | องค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive Element : HDE) .....               | 3    |
| 1.3    | การตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (HDE view) .....                          | 3    |
| 1.4    | จำนวนการตรวจสอบความผิดปกติขององค์ประกอบฮาร์ดดิสก์ไดรฟ์<br>Fail และ Overkill..... | 4    |
| 2.1    | ส่วนประกอบฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive : HDD).....                           | 9    |
| 2.2    | Pixel แต่งโม.....  | 10   |
| 2.3    | ขนาด Resolution ของภาพและวิดีโอ .....  | 10   |
| 2.4    | การครอบรูปภาพ.....   | 11   |
| 2.5    | เปรียบเทียบ AI, Machine learning และ Deep learning .....                         | 12   |
| 2.6    | ความแตกต่างระหว่าง Machine learning กับ Programming .....                        | 13   |
| 2.7    | ประเภทการเรียนรู้ของเครื่องจักร .....  | 13   |
| 2.8    | การเรียนรู้ของโครงข่ายประสาทของมนุษย์เทียบกับโครงข่ายประสาทเทียม .....           | 14   |
| 2.9    | โครงสร้างของโครงข่ายประสาทเทียม .....  | 15   |
| 2.10   | Activation function.....   | 16   |
| 2.11   | เครือข่ายการส่งข้อมูลไปข้างหน้าของโครงข่ายประสาทเทียม .....                      | 17   |
| 2.12   | แบบจำลองโครงข่ายประสาทเทียมแบบเครือข่ายการส่งข้อมูลย้อนกลับ .....                | 18   |
| 2.13   | สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) .....                       | 19   |
| 2.14   | ลักษณะตัวกรอง .....  | 20   |
| 2.15   | กำหนดเลื่อนตัวกรองเท่ากับ 1.....   | 21   |
| 2.16   | กำหนดเลื่อนตัวกรองเท่ากับ 2.....   | 21   |
| 2.17   | การทำ padding .....  | 22   |
| 2.18   | การทำ Max pooling และ Mean pooling .....   | 23   |
| 2.19   | ประเภทของ Object detection .....   | 24   |
| 2.20   | Sliding window.....  | 25   |

## สารบัญรูป (ต่อ)

| รูปที่ | หน้า  |
|--------|---|
| 2.21   | Area of IoU..... 25   |
| 2.22   | ตัวอย่างการซ้อนทับ..... 26  |
| 2.23   | ตาราง Confusion matrix..... 26  |
| 2.24   | โครงสร้างของ YOLOv3 ..... 28  |
| 2.25   | YOLOv3 เปรียบเทียบกับ Algorithm อื่นช่วงเวลานั้น ..... 28                                     |
| 2.26   | ประสิทธิภาพของ YOLOv5..... 29   |
| 2.27   | Algorithm ย่อยของ YOLOv5..... 29  |
| 2.28   | โครงสร้างสถาปัตยกรรมของ YOLOv5 Algorithm..... 30  |
| 2.29   | ตัวอย่างการตรวจจับวัตถุด้วย YOLOv5..... 31  |
| 3.1    | องค์ประกอบของฮาร์ดดิสก์ไดรฟ์มุมมองด้านบน (HDE view) ..... 36                                  |
| 3.2    | จำนวนภาพที่เกิดความผิดพลาดจริงเทียบกับไม่จริง (Overkill) ของ HDE view ..... 37                |
| 3.3    | ประเภทของภาพ HDE view ..... 38  |
| 3.4    | แผนผังการดำเนินการทดลอง VI system..... 39   |
| 3.5    | การปรับความสว่างของภาพ HDE view ..... 40  |
| 3.6    | Folder ชุดทดสอบระบบ VI system..... 42   |
| 3.7    | สถาปัตยกรรมของระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิค<br>การเรียนรู้เชิงลึก..... 43 |
| 3.8    | การแบ่งข้อมูลสำหรับสร้างโมเดล CNN..... 44   |
| 3.9    | ขอบบนและล่างของภาพ HDE view..... 45   |
| 3.10   | การครอบภาพ HDE view..... 45   |
| 3.11   | แผนผังการดำเนินการทดลอง VI system ใหม่..... 46  |
| 3.12   | สถาปัตยกรรม VI system แบบใหม่..... 46   |
| 3.13   | การสร้าง Label image HDE view ..... 48  |
| 3.14   | การแบ่งข้อมูลสำหรับสร้างโมเดล YOLOv5..... 48  |
| 3.15   | รหัส RS code และตำแหน่งตรวจสอบของภาพ HDE view..... 49   |
| 3.16   | แผนผังการฝึกสอนโมเดล YOLOv5 ..... 49  |
| 3.17   | กราฟแสดงการเปรียบเทียบเปอร์เซ็นต์ความถูกต้องของโมเดล YOLOv5..... 50                           |

## สารบัญรูป (ต่อ)

| รูปที่ | หน้า  |
|--------|---|
| 3.18   | ภาพตัวอย่างของการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์..... 51        |
| 4.1    | ภาพ HDE view Pass image จากโมเดล CNN..... 55                        |
| 4.2    | ภาพ HDE view Fail images จากโมเดล CNN ..... 55                      |
| 4.3    | ภาพ HDE view Pass image จาก YOLOv5..... 56                          |
| 4.4    | ภาพ HDE view Fail image จาก YOLOv5 ..... 57                         |
| 4.5    | Flow chart แสดงการทดสอบ VI system กับ Dataset1 ..... 58             |
| 4.6    | Confusion matrix สำหรับ VI system กับ Dataset1 ..... 58             |
| 4.7    | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset1 ..... 59 |
| 4.8    | Flow chart แสดงการทดสอบ VI system กับ Dataset2 ..... 60             |
| 4.9    | Confusion matrix สำหรับ VI system กับ Dataset2 ..... 60             |
| 4.10   | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset2..... 61  |
| 4.11   | Flow chart แสดงการทดสอบ VI system กับ Dataset3 ..... 62             |
| 4.12   | Confusion matrix สำหรับ VI system กับ Dataset3 ..... 62             |
| 4.13   | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset3..... 63  |
| 4.14   | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset3..... 64  |
| 4.15   | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset3..... 64  |
| 4.16   | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset3..... 65  |
| 4.17   | Flow chart แสดงการทดสอบ VI system กับ Dataset4..... 66              |
| 4.18   | Confusion matrix สำหรับ VI system กับ Dataset4 ..... 66             |
| 4.19   | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset4..... 67  |
| 4.20   | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset4..... 68  |
| 4.21   | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset4..... 68  |
| 4.22   | ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset4..... 69  |
| ค.1    | ข้อมูล HDE view สำหรับทดสอบระบบ ..... 98                            |
| ค.2    | Command Prompt สำหรับ run โปรแกรม..... 98                           |
| ค.3    | Run โปรแกรม..... 99   |
| ค.4    | ผลสรุปผลการทำงานระบบตรวจสอบองค์ประกอบฮาร์ดดิสก์ไดรฟ์ ..... 99       |

## สารบัญรูป (ต่อ)

| รูปที่ |                          | หน้า |
|--------|--------------------------|------|
| ค.5    | รูปจากโมเดล CNN.....     | 100  |
| ค.6    | รูปจากโมเดล YOLOv5 ..... | 100  |



# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของปัญหาการวิจัย

อุปกรณ์จัดเก็บข้อมูลดิจิทัล (Digital data) มีความสำคัญเป็นอย่างมากในยุคปัจจุบันและในปัจจุบันอุปกรณ์จัดเก็บข้อมูลดิจิทัลในมีหลากหลายชนิดมาก ฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive : HDD) เป็นหนึ่งในอุปกรณ์จัดเก็บข้อมูลดิจิทัลที่สำคัญเนื่องจากมีพื้นที่ในการจัดเก็บข้อมูลจำนวนมากต่อหนึ่งหน่วย ดังนั้น การผลิตฮาร์ดดิสก์ไดรฟ์ในปัจจุบันยังมีความจำเป็นอยู่แม้ว่าจะมีอุปกรณ์จัดเก็บข้อมูลดิจิทัลชนิดอื่นที่มีความรวดเร็วในการอ่านและเขียนข้อมูลสูงกว่าฮาร์ดดิสก์ไดรฟ์ โดยในปัจจุบันเป็นยุคของข้อมูลขนาดใหญ่ (Big data)

ในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์หนึ่งในขั้นตอนที่สำคัญคือการประกอบฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive Assembly : HDA) ดังรูปที่ 1.1 เป็นกระบวนการที่นำชิ้นส่วนต่างๆของฮาร์ดดิสก์ไดรฟ์มาประกอบรวมกันให้เป็นฮาร์ดดิสก์ไดรฟ์ ดังนั้นการตรวจสอบความถูกต้องของตำแหน่งองค์ประกอบต่างๆ ภายในฮาร์ดดิสก์ไดรฟ์ในกระบวนการผลิตจึงเป็นปัจจัยสำคัญอย่างหนึ่งเพื่อให้แน่ใจว่าองค์ประกอบภายในฮาร์ดดิสก์ไดรฟ์ไม่มีความผิดปกติ

ระบบการตรวจสอบความถูกต้องขององค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive Element : HDE) ดังรูปที่ 1.2 ในปัจจุบันคือระบบการมองเห็นของเครื่องจักร (Machine vision systems) โดยใช้เทคนิคต่างๆ ของการประมวลผลภาพ (Image processing) ซึ่งมีข้อจำกัดเกี่ยวกับขอบเขตการมองเห็น (Field Of View : FOV) และยังมีคามยืดหยุ่นในการใช้งานน้อยดังเช่นตัวอย่างรูปที่ 1.3 จะเห็นได้ว่าองค์ประกอบต่าง ๆ ของฮาร์ดดิสก์ไดรฟ์ไม่ได้ผิดปกติแต่ระบบกลับตรวจสอบว่าผิดปกติเนื่องจากมีส่วนที่อยู่นอกเหนือจากขอบเขตการมองเห็น ระบบจึงตรวจสอบว่าส่วนประกอบดังกล่าวผิดปกติ ดังนั้น ในงานวิจัยนี้จึงได้นำเสนอเกี่ยวกับเทคนิคการเรียนรู้เชิงลึก (Deep Learning : DL) ถือเป็นหนึ่งในกระบวนการเรียนรู้ของเครื่องจักร (Machine Learning : ML) โดยมีโครงสร้างและการประมวลผลคล้ายกับสมองของมนุษย์โดยเรียกว่าโครงข่ายประสาทเทียม (Artificial Neural Network : ANN) ในงานวิจัยนี้ จะนำเสนอการออกแบบระบบตรวจสอบองค์ประกอบต่างๆ ของฮาร์ดดิสก์ไดรฟ์ (Visual Inspection system : VI system) ด้วยโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network : CNN) กับโพลีเมอร์ชั้นห้าอัลกอริทึม (YOLOv5 Algorithm) ซึ่งเป็นหนึ่งในอัลกอริทึมสำคัญสำหรับการตรวจจับวัตถุอัตโนมัติมีประสิทธิภาพสูงและตรวจจับวัตถุได้อย่างรวดเร็วและมีความยืดหยุ่นสูงเพราะไม่ได้จำกัด



ขอบเขตการมองเห็นแต่จะเน้นไปที่การตรวจสอบว่ามีวัตถุที่ได้ทำการฝึกสอนโมเดลไว้อยู่ในภาพหรือไม่

งานวิจัยนี้ทำขึ้นมาเพื่อจัดการปัญหาการตรวจสอบองค์ประกอบฮาร์ดดิสก์ไดรฟ์ผิดพลาด เพราะจะทำให้เกิดปัญหาในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ได้ ซึ่งเกิดจากการตรวจสอบผิดพลาดต่างๆ ที่ไม่ได้มีอะไรผิดปกติเรียกการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ผิดพลาดนี้ว่า “Overkill” จากการตรวจสอบภาพทั้งหมด 739 ภาพของฮาร์ดดิสก์ไดรฟ์จากระบบเดิมแบ่งกลุ่มของภาพออกได้เป็น 3 ประเภท Good 418 ภาพ Fail 177 ภาพ และ Overkill 144 ภาพ ตารางที่ 1.1 แสดงการแยกประเภทตำแหน่งขององค์ประกอบฮาร์ดดิสก์ไดรฟ์ที่ตรวจสอบว่าผิดปกติทั้ง Fail และ Overkill และสรุปเป็นแผนภูมิรูปภาพในรูปที่ 1.4

ตารางที่ 1.1 ตารางแสดงการตรวจสอบตำแหน่งผิดปกติของฮาร์ดดิสก์ไดรฟ์ Fail และ Overkill

| No. | Parameter    | EC | RS code | G-code | Note                     | Failed | Overkill |
|-----|--------------|----|---------|--------|--------------------------|--------|----------|
| 1   | BASE1/BASE2  | 3  | RS03    | 3279   | Extra parts in MBA       | 0      | 0        |
| 2   | EXTRA        | 3  | RS03    | 3279   | Extra parts in MBA       | 0      | 0        |
| 3   | ExPartDisk   | 20 | RS20    | 3292   | Extra parts on media     | 141    | 1        |
| 4   | C-Filter     | 6  | RS06    | 3281   | C-Filter missing         | 28     | 2        |
| 5   | CLIP         | 7  | RS07    | 3282   | Comb not remove          | 3      | 2        |
| 6   | FPC          | 10 | RS10    | 3284   | FCC or screw problem     | 0      | 0        |
| 7   | FPCSCR1      | 10 | RS10    | 3284   | FCC or screw problem     | 0      | 22       |
| 8   | FPCSCR2      | 10 | RS10    | 3284   | FCC or screw problem     | 2      | 0        |
| 9   | HSA          | 11 | RS11    | 3285   | HSA damage or no HSA     | 2      | 8        |
| 10  | OUTERCS      | 14 | RS14    | 3287   | ODCS damage or missing   | 0      | 15       |
| 11  | Pivot        | 15 | RS15    | 3288   | Pivot abnormal or damage | 0      | 16       |
| 12  | RAMP         | 16 | RS16    | 3437   | Ramp missing or damage   | 1      | 6        |
| 13  | SPOILER      | 12 | RS12    | 3431   | Spoiler or screw missing | 1      | 0        |
| 14  | SpoilerScrew | 12 | RS12    | 3431   | Spoiler or screw missing | 0      | 59       |
| 15  | TC           | 17 | RS17    | 3290   | Top clamp missing        | 0      | 0        |
| 16  | TC_Center    | 17 | RS17    | 3290   | Top clamp missing        | 0      | 14       |
| 17  | VCM          | 18 | RS18    | 3291   | TVCM missing             | 1      | 0        |



รูปที่ 1.1 ไหล่การผลิตฮาร์ดดิสก์ไดรฟ์

| Camera | Parameter | EC           | RS | G-code | Note |                          |
|--------|-----------|--------------|----|--------|------|--------------------------|
| HDE    | 1         | BASE1/BASE2  | 3  | RS03   | 3279 | Extra parts in MBA       |
|        | 2         | EXTRA        | 3  | RS03   | 3279 | Extra parts in MBA       |
|        | 3         | ExPartDisk   | 20 | RS20   | 3292 | Extra parts on media     |
|        | 4         | C-Filter     | 6  | RS06   | 3281 | C-Filter missing         |
|        | 5         | CLIP         | 7  | RS07   | 3282 | Comb <u>not</u> remove   |
|        | 6         | FPC          | 10 | RS10   | 3284 | FCC or screw problem     |
|        | 7         | FPCSCR1      | 10 | RS10   | 3284 | FCC or screw problem     |
|        | 8         | FPCSCR2      | 10 | RS10   | 3284 | FCC or screw problem     |
|        | 9         | HSA          | 11 | RS11   | 3285 | HSA damage or no HSA     |
|        | 10        | OUTERCS      | 14 | RS14   | 3287 | ODCS damage or missing   |
|        | 11        | Pivot        | 15 | RS15   | 3288 | Pivot abnormal or damage |
|        | 12        | RAMP         | 16 | RS16   | 3437 | Ramp missing or damage   |
|        | 13        | SPOILER      | 12 | RS12   | 3431 | Spoiler or screw missing |
|        | 14        | SpoilerScrew | 12 | RS12   | 3431 | Spoiler or screw missing |
|        | 15        | TC           | 17 | RS17   | 3290 | Top clamp missing        |
|        | 16        | TC_Center    | 17 | RS17   | 3290 | Top clamp missing        |
|        | 17        | VCM          | 18 | RS18   | 3291 | TVCM missing             |

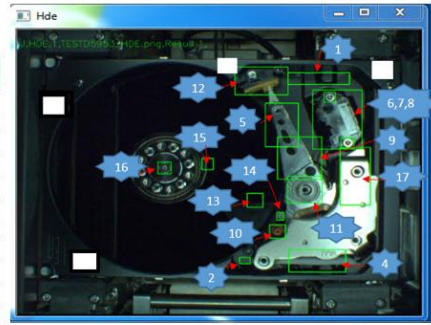


Figure 10: Pattern and Find Area for Ex-PartsChkCircle

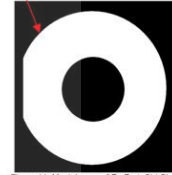
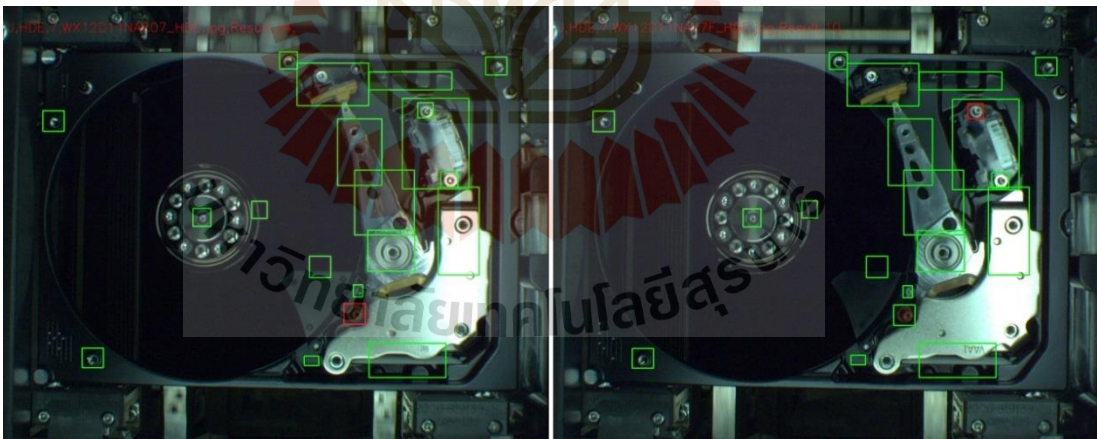


Figure 11: Mask Image of Ex-PartsChkCircle

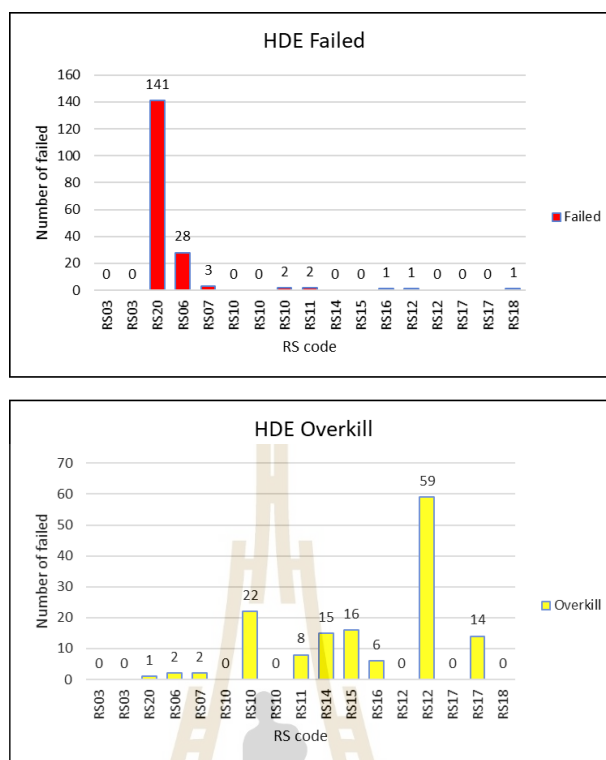
Brightness spec

| Brightness(lux) | Top view | Ramp view | Side view |
|-----------------|----------|-----------|-----------|
| Target          | 163      | 128       | 97        |
| Max             | 173      | 138       | 107       |
| Min             | 153      | 118       | 87        |

รูปที่ 1.2 องค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive Element : HDE)



รูปที่ 1.3 การตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (HDE view)



รูปที่ 1.4 จำนวนการตรวจสอบความผิดปกติขององค์ประกอบฮาร์ดดิสก์ไทรฟ์ Fail และ Overkill

## 1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษาเกี่ยวกับเทคนิคต่าง ๆ ของการประมวลผลภาพ (Image processing)

1.2.2 เพื่อศึกษาและออกแบบระบบการตรวจสอบองค์ประกอบต่าง ๆ ภายในฮาร์ดดิสก์ไทรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึก (Deep learning)

1.2.3 เพื่อปรับปรุงกระบวนการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไทรฟ์ให้มีความแม่นยำสูงและรวดเร็ว

1.2.4 ระบบตรวจสอบองค์ประกอบต่าง ๆ ของฮาร์ดดิสก์ไทรฟ์มีประสิทธิภาพในการตรวจสอบสูงกว่า 98% และมีความเร็วในการตรวจสอบแต่ละภาพไม่เกิน 2 วินาที

## 1.3 ขอบเขตของงานวิจัย

1.3.1 ระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไทรฟ์สำหรับ NEO automation line (Venice product)

1.3.2 VI system จะประกอบไปด้วย Convolution neural network สำหรับตรวจสอบภาพโดยรวม และ Deep learning (YOLOv5 algorithm model S) สำหรับตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์มุมมอง HDE view เท่านั้น

1.3.3 ประเมินประสิทธิภาพการทำงานของ VI system ด้วย Confusion matrix และ F1-Score

1.3.4 มีชุดทดสอบ VI system ทั้งหมด 4 ชุดแต่ละชุดประกอบไปด้วยภาพ HDE view (Pass = 375 ภาพ, Fail = 115 ภาพ)

1.3.5 ภาพต้นฉบับของ HDE view สำหรับออกแบบระบบ VI system และ test มี resolution = 1280x1024 pixel

1.3.6 VI system เป็นระบบต้นแบบสำหรับการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (Venice product) เท่านั้น

## 1.4 ระเบียบวิธีวิจัย

1.4.1 ค้นคว้าศึกษาองค์ความรู้ ทฤษฎี และเอกสารที่เกี่ยวข้องกับกระบวนการทางรูปภาพและการตรวจสอบวัตถุภายในภาพ รวมถึงการเรียนรู้ของเครื่องจักรและการเรียนรู้เชิงลึก

1.4.2 ออกแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์โดยใช้ Deep learning technique

1.4.3 เก็บข้อมูลรูปภาพ HDE view จากระบบเดิมและทำการคัดแยกรูปภาพดีหรือเสีย (Pass or Fail)

1.4.4 เตรียมรูปภาพที่ได้จากการคัดแยกรูปภาพ (Cleaning data) เพื่อใช้สำหรับสร้างโมเดลการเรียนรู้เชิงลึกทั้ง CNN และ Object detection

1.4.5 สร้างระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ HDE view ซึ่งจะประกอบไปด้วยโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) และ การตรวจสอบองค์ประกอบภายในภาพด้วยเทคนิคการเรียนรู้เชิงลึก (Deep learning YOLOv5 algorithm)

1.4.6 ออกแบบการทดสอบประสิทธิภาพของโมเดล และประเมินประสิทธิภาพของโมเดล

1.4.7 ปรับปรุงประสิทธิภาพของระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (VI system)

1.4.8 วิเคราะห์และสรุปผลลัพท์การทำงานของระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (VI system)

1.4.9 จัดทำเอกสารและรายงานการวิจัย

## 1.5 สถานที่ทำงานวิจัย

- 1.5.1 System and Control Lab มหาวิทยาลัยเทคโนโลยีสุรนารี
- 1.5.2 บริษัท เวสเทิร์น ดิจิตอล ประเทศไทย จำกัด

## 1.6 เครื่องมือที่ใช้ในการทำวิจัย

- 1.6.1 Python programming language สำหรับสร้าง Model deep learning และ image processing
- 1.6.2 YOLOv5 algorithm model S สำหรับสร้าง Object detection
- 1.6.3 รูปภาพของฮาร์ดดิสก์ไดรฟ์ HDE view จากเครื่อง Visual Inspection machine ของ Neo automation line (Venice product)
- 1.6.4 คอมพิวเตอร์ส่วนบุคคล และจอมอนิเตอร์

## 1.7 ประโยชน์ที่คาดว่าจะได้รับ

- 1.7.1 ได้องค์ความรู้เกี่ยวกับการประมวลผลทางรูปภาพและเทคนิคการเรียนรู้เชิงลึก
- 1.7.2 ได้องค์ความรู้เกี่ยวกับการออกแบบระบบตรวจสอบวัตถุ และสามารถนำไปประยุกต์ใช้ร่วมกับโปรเจกต์ต่าง ๆ ได้ในอนาคต
- 1.7.3 ได้ต้นแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์โดยใช้เทคนิคการเรียนรู้เชิงลึก
- 1.7.4 เพิ่มประสิทธิภาพในการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ HDE view
- 1.7.5 เพิ่มประสิทธิภาพกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ได้
- 1.7.6 เผยแพร่บทความงานวิจัยในงานประชุมวิชาการภายในประเทศ

## 1.8 การจัดทำรูปเล่มวิทยานิพนธ์

บทที่ 1 เป็นบทนำของเล่มวิทยานิพนธ์นี้ซึ่งจะกล่าวถึงที่มาและความสำคัญของปัญหา วัตถุประสงค์ รวมถึงขอบเขต และประโยชน์ที่คาดว่าจะได้รับจากงานวิจัยนี้

บทที่ 2 เป็นบทที่เกี่ยวกับทฤษฎีต่าง ๆ และงานวิจัยที่เกี่ยวข้องกับระบบตรวจสอบองค์ประกอบฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึก และการประมวลผลรูปภาพที่เกี่ยวข้องกับงานวิจัย

บทที่ 3 เป็นบทที่กล่าวถึงการเตรียมข้อมูลรูปภาพฮาร์ดดิสก์ไดรฟ์ HDE view และการออกแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึกโดยจะใช้ CNN และ YOLOv5 ในการสร้างระบบขึ้นมา

บทที่ 4 เป็นบทที่อธิบายเกี่ยวกับการทดสอบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ รวมถึงการประเมินประสิทธิภาพและการปรับปรุงประสิทธิภาพของระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ให้เป็นไปตามวัตถุประสงค์ที่กำหนดไว้

บทที่ 5 เป็นบทสรุปและข้อเสนอแนะ



## บทที่ 2

### ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

#### 2.1 ส่วนประกอบของฮาร์ดดิสก์ไดรฟ์

ฮาร์ดดิสก์ไดรฟ์ (Hard disk drive) เป็นอุปกรณ์อิเล็กทรอนิกส์ชนิดหนึ่งที่ทำหน้าที่ในการจัดเก็บข้อมูลดิจิทัล (Digital data) ต่าง ๆ ซึ่งมีอยู่หลายประเภทเช่น ไฟล์ภาพ วิดีโอ เอกสาร รูปภาพ เป็นต้น โดยจะใช้งานข้อมูลเหล่านั้น กับคอมพิวเตอร์ เซิร์ฟเวอร์ อินเทอร์เน็ต และในปัจจุบันฮาร์ดดิสก์ไดรฟ์ยังมีความสำคัญเป็นอย่างมากในยุคของข้อมูลขนาดใหญ่ (Big data) และอินเทอร์เน็ตของสรรพสิ่ง (Internet of Thing : IoT) ที่การสร้างและเคลื่อนที่ของข้อมูลดิจิทัลเกิดขึ้นรวดเร็วเป็นอย่างมากโดยที่ข้อมูลดิจิทัลจะถูกจัดเก็บในรูปของโดเมนแม่เหล็ก 0 และ 1 ซึ่งมาขนาดเล็กมาก ๆ ผ่านหัวอ่าน/เขียน (Head Gimbal Assembly : HGA) ข้อมูลดิจิทัลลงบนแผ่น Media

ส่วนประกอบของฮาร์ดดิสก์ไดรฟ์ มีส่วนประกอบต่าง ๆ ดังนี้ โดยแสดงดังรูปที่ 2.1

##### 1. จานแข็ง (Platters)

จานแข็งหรือเรียกอีกอย่างว่าแผ่นมีเดียสร้างขึ้นจากโลหะผสมโดยจะเคลือบด้วยสารที่มีคุณสมบัติทางแม่เหล็กเนื่องจากการบันทึกข้อมูลลงบนแผ่นมีเดียนั้นจะใช้หลักการเหนี่ยวนำแม่เหล็ก โดยจะเปรียบข้อมูล 0 และ 1 ให้เป็นขั้วเหนือและขั้วใต้

##### 2. มอเตอร์หมุนแผ่นแม่เหล็ก (Spindle Motor)

ทำหน้าที่หมุนแผ่นมีเดียด้วยความเร็วมาตรฐานของฮาร์ดดิสก์ไดรฟ์ทั่วไปในปัจจุบันจะมีความเร็วรอบอยู่ที่ 7200 RPM (Rotations Per Minute : จำนวนรอบต่อนาที) แต่สำหรับฮาร์ดดิสก์ไดรฟ์ที่มีประสิทธิภาพสูงกว่ารุ่นทั่วไปจะมีความเร็วรอบอยู่ที่ประมาณ 10000 หรือ 15000 RPM

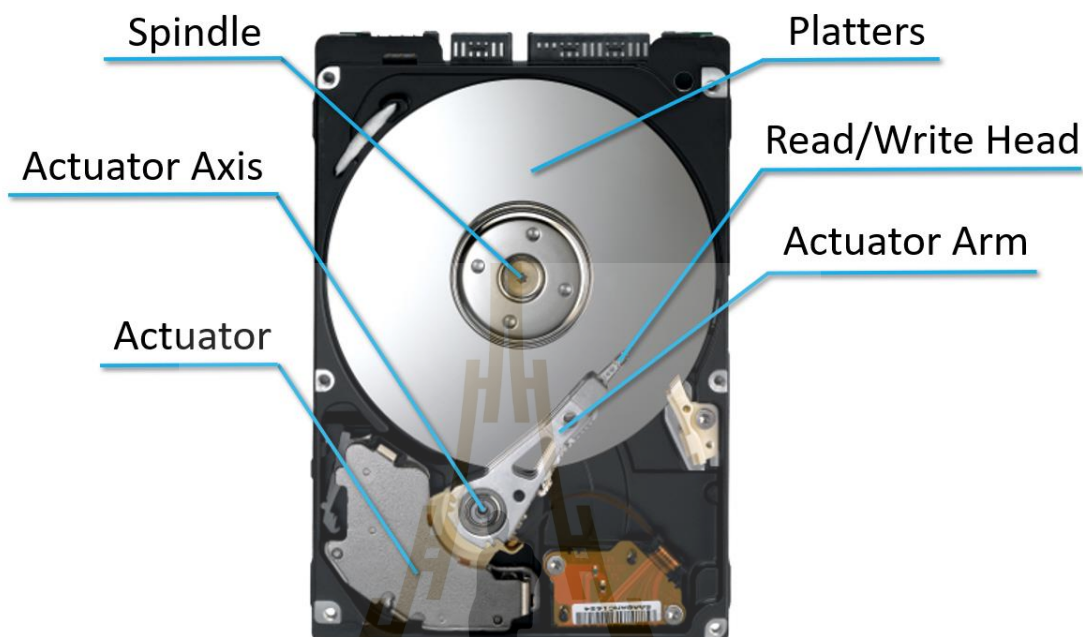
##### 3. แขนหัวอ่าน/เขียนข้อมูล (Actuator arm)

เป็นส่วนแขนสำหรับอ่านและเขียนข้อมูลดิจิทัลลงบนแผ่นมีเดีย ซึ่งส่วนปลายของแขนอ่าน/เขียนข้อมูลจะประกอบไปด้วยหัวอ่าน/เขียนข้อมูลเรียงซ้อนกัน

##### 4. หัวอ่าน/เขียน (Head Gimbal Assembly : HGA)

เป็นส่วนประกอบสำคัญของฮาร์ดดิสก์ไดรฟ์ซึ่งประกอบไปด้วย Suspension และ Slider ที่สร้างขึ้นจากแผ่น Wafer ซึ่งประกอบไปด้วยธาตุต่าง ๆ ที่มีคุณสมบัติทางแม่เหล็กโดยที่หัวอ่าน/เขียนจะเป็นส่วนที่ใช้ในการสั่งการจดบันทึกหรืออ่านข้อมูลจากแผ่น Media โดยได้รับสัญญาณไฟฟ้ามาจากตัวคอนโทรลเลอร์ก่อนเกิดการเหนี่ยวนำแม่เหล็ก

5. เคส (Case) หรือตัวกล่องสี่เหลี่ยมเป็นชิ้นส่วนที่ใช้ในการรวมชิ้นส่วนต่าง ๆ มาประกอบเป็นฮาร์ดดิสก์ไดรฟ์



รูปที่ 2.1 ส่วนประกอบฮาร์ดดิสก์ไดรฟ์ (Hard Disk Drive : HDD)

## 2.2 การประมวลผลภาพ (Image processing)

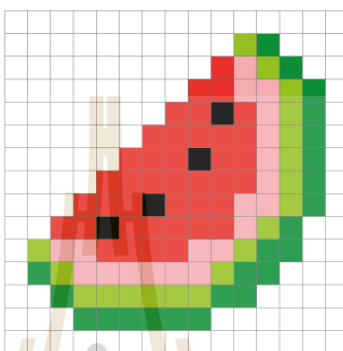
เป็นการจัดการหรือการประมวลผลรูปภาพ หรือวิดีโอให้เป็นข้อมูลในรูปแบบดิจิทัล โดยประมวลผลผ่านคอมพิวเตอร์ไม่ว่าจะเป็นการใช้โปรแกรมต่าง ๆ เพื่อจัดการกับรูปภาพหรือวิดีโอ นั้น ให้ได้ข้อมูลที่ต้องการทั้งในเชิงคุณภาพหรือปริมาณ (ขนาด รูปร่าง) รวมไปถึงคุณลักษณะเด่นของภาพ หรือวิดีโอ เพื่อนำข้อมูลที่ได้นำไปประยุกต์ใช้งานสำหรับงานด้านต่าง ๆ เช่น งานด้านภาพยนตร์ งานด้านตกแต่งรูปภาพ งานด้านอุตสาหกรรมที่มีการตรวจเช็คผลิตภัณฑ์ด้วยกล้อง ระบบตรวจสอบข้อมูลสินค้าด้วยบาร์โค้ด คิวอาร์โค้ด เป็นต้น ที่กล่าวมาทั้งหมดนั้นใช้งานโปรแกรมด้านการประมวลผลภาพทั้งสิ้น

### 2.2.1 พิกเซล (Pixel)

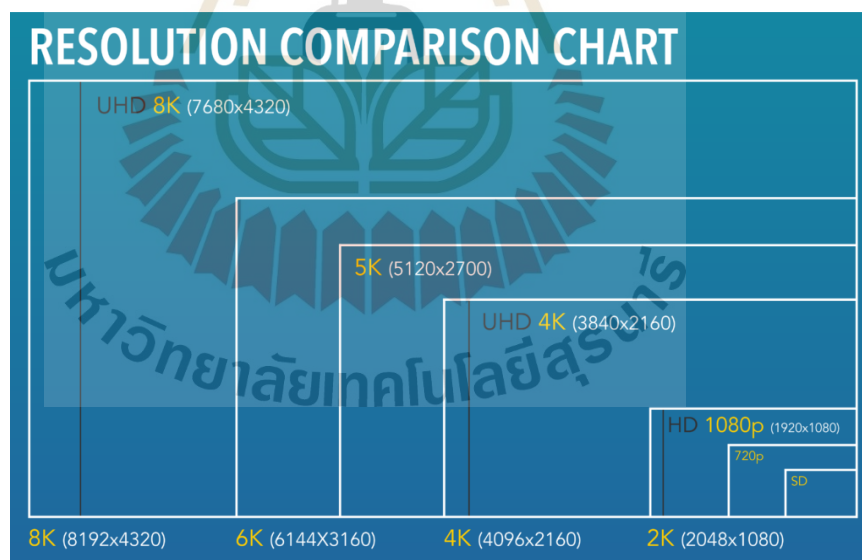
พิกเซลมาจากคำว่า พิกเจอร์ (Picture) รวมกับ เอเลเมนต์ (Element) เปรียบเสมือนจุดเล็ก ๆ ของภาพดิจิทัลซึ่งแต่ละจุดพิกเซล นั้น จะมีค่าสีที่แตกต่างกันเมื่อพิกเซลมารวมกันตัวกันเป็นจำนวนมาก ๆ ก็จะเกิดเป็นรูปภาพดิจิทัลขึ้นมาเช่นตัวอย่างรูปที่ 2.2 นั้น เป็นภาพแต่งโมที่ถูกแบ่งเป็นตารางในแต่ละช่องของตารางคือพิกเซลจะมีค่าของสีอยู่หนึ่งค่า การแสดงผลด้าน



กราฟฟิก เช่น เส้น จุด สีของภาพ อื่น ๆ ล้วนประกอบไปด้วยพิกเซลทั้งสิ้นยิ่งภาพนั้นมีความละเอียดของพิกเซลมากเท่าไรความชัดเจนของภาพก็จะยิ่งมากขึ้นเท่านั้น แต่ก็แลกมาด้วยขนาดข้อมูลที่ใหญ่ขึ้นด้วยเนื่องจากพิกเซลนั้นจะอยู่ในรูปแบบของเมตริกซ์ที่มีจำนวนคอลัมน์ (Column) และแถว (Row) ในปัจจุบันจะมีการแบ่งความละเอียดของภาพออกเป็น SD 720p 1080p 2k 4k เป็นต้นตามรูปภาพที่ 2.3 บอกรายละเอียดเกี่ยวกับขนาดของพิกเซลของรูปภาพและวิดีโอที่ใช้งานอยู่ในปัจจุบัน



รูปที่ 2.2 Pixel แต่งโม

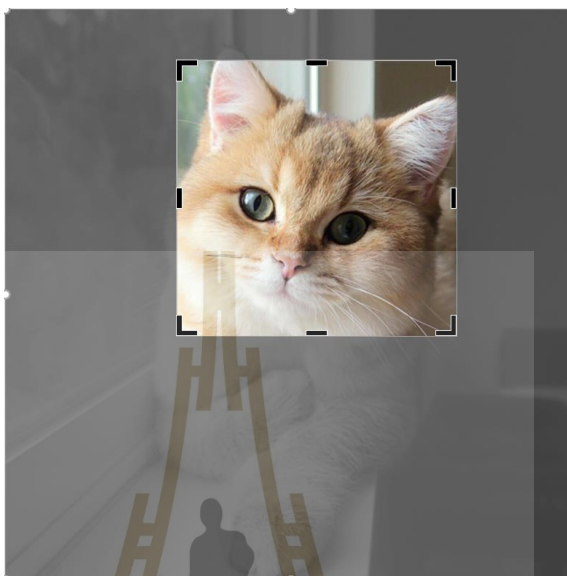


รูปที่ 2.3 ขนาด Resolution ของภาพและวิดีโอ (อ้างอิง [www.napavalleytv.org/](http://www.napavalleytv.org/))

## 2.2.2 การครอบภาพ (Crop image)

เป็นหนึ่งในเครื่องมือของการประมวลผลภาพซึ่งการครอบภาพนั้นใช้งานสำหรับตัดส่วนที่ไม่ต้องการของภาพออกโดยการตัดพิกเซลออกไม่ว่าจะเป็นจำนวนคอลัมน์ (Column) หรือแถว

(Row) ขึ้นอยู่กับการใช้งานโดยการครอบภาพนั้นส่วนมากจะนิยมครอบเป็นแบบสี่เหลี่ยมในอัตราส่วนต่าง ๆ ไม่ว่าจะเป็น 1:1 16:9 4:3 21:9 เป็นต้นขึ้นอยู่กับการใช้งานเช่นรูปที่ 2.4



รูปที่ 2.4 การครอบรูปภาพ

### 2.3 ปัญญาประดิษฐ์ (Artificial Intelligence : AI)

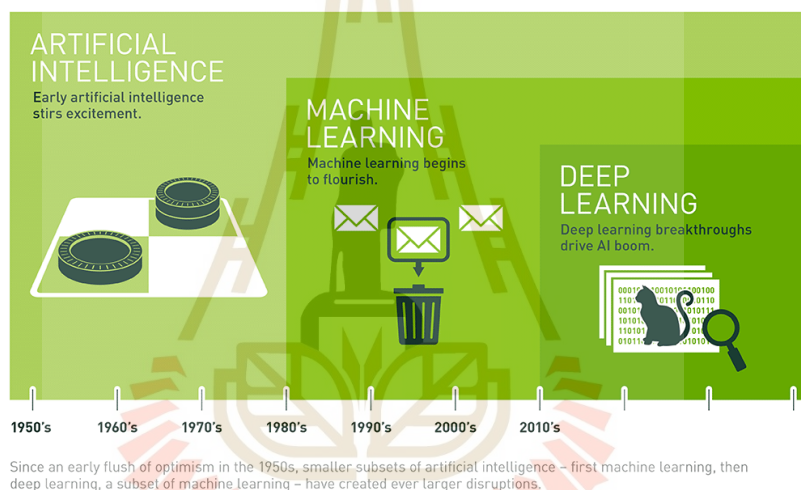
ปัญญาประดิษฐ์ คือ โปรแกรมที่ถูกพัฒนาให้มีความสามารถในการคิด วิเคราะห์ที่มีความฉลาดเลียนแบบสมองมนุษย์ โดยที่ปัญญาประดิษฐ์สามารถที่จะประมวลผลข้อมูลที่มีขนาดใหญ่และสามารถที่จะคิดอย่างมีเหตุผลและการกระทำคล้ายกับมนุษย์ได้โดยเปรียบได้ว่าปัญญาประดิษฐ์นั้นจะนำผลลัพธ์ที่ได้จาก Machine Learning มาแสดงผลหรือนำไปใช้งานในรูปแบบต่าง ๆ โดยที่ปัญญาประดิษฐ์จะเป็นภาพรวมกว้าง ๆ ของโปรแกรมอัจฉริยะโดยมีซบเซตเป็น การเรียนรู้ของเครื่องจักร (Machine Learning : ML) และการเรียนรู้เชิงลึก (Deep Learning : DL) ดังรูปที่ 2.5 ซึ่งในปัจจุบันนั้น AI มีความสำคัญเป็นอย่างมากในวงการเทคโนโลยีเช่น สมาร์ทโฟน หลาย ๆ รุ่นได้มีโปรแกรม AI ช่วยในการตัดสินใจการทำงานร่วมกับหน่วยประมวลผลกลาง (Central Processing Unit : CPU) ของสมาร์ทโฟน หรือในส่วนของแอปพลิเคชัน (Application) ก็มี AI ช่วยในการทำงานด้วยเช่นกัน เช่น Face ID ของ Iphone, การสแกนนิ้วมือ, การปรับปรุงภาพด้วย AI ด้วยการวิเคราะห์รูปภาพ หรือแสง เป็นต้น

AI ถูกจำแนกเป็น 3 ระดับตามความสามารถหรือความฉลาด ดังนี้

**2.3.1 ปัญญาประดิษฐ์เชิงแคบ (Narrow AI)** คือ AI ที่มีความสามารถเฉพาะทางได้ดีกว่ามนุษย์

**2.3.2 ปัญญาประดิษฐ์ทั่วไป (General AI)** คือ AI ที่มีความสามารถระดับเดียวกับมนุษย์สามารถทำทุก ๆ อย่างที่มนุษย์ทำได้และมีประสิทธิภาพที่ใกล้เคียงกับมนุษย์

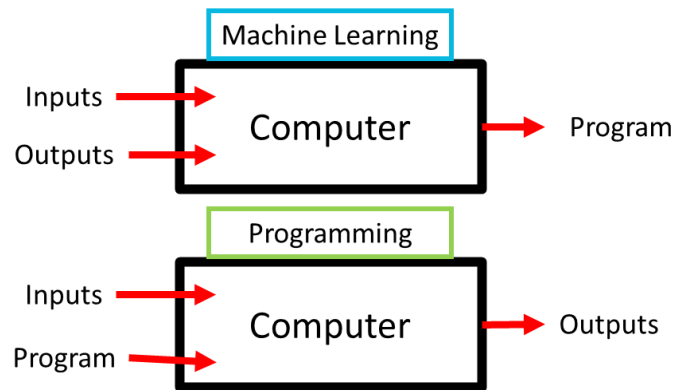
**2.3.3 ปัญญาประดิษฐ์แบบเข้ม (Strong AI)** คือ AI ที่มีความสามารถเหมือนมนุษย์ในหลายๆ ด้าน พัฒนาให้เครื่องจักรสามารถเรียนรู้และทำนายองค์ความรู้ได้ สามารถจำลองการทำงานต่าง ๆ เลียนแบบพฤติกรรมของมนุษย์ได้ โดยเน้นแนวคิดตามแบบสมองมนุษย์ที่มีการวางแผนการเรียนรู้ การให้เหตุผล การตัดสินใจ การแก้ปัญหา



รูปที่ 2.5 เปรียบเทียบ AI, Machine learning และ Deep learning  
(อ้างอิง <https://blogs.nvidia.com>)

## 2.4 การเรียนรู้ของเครื่องจักร (Machine Learning : ML)

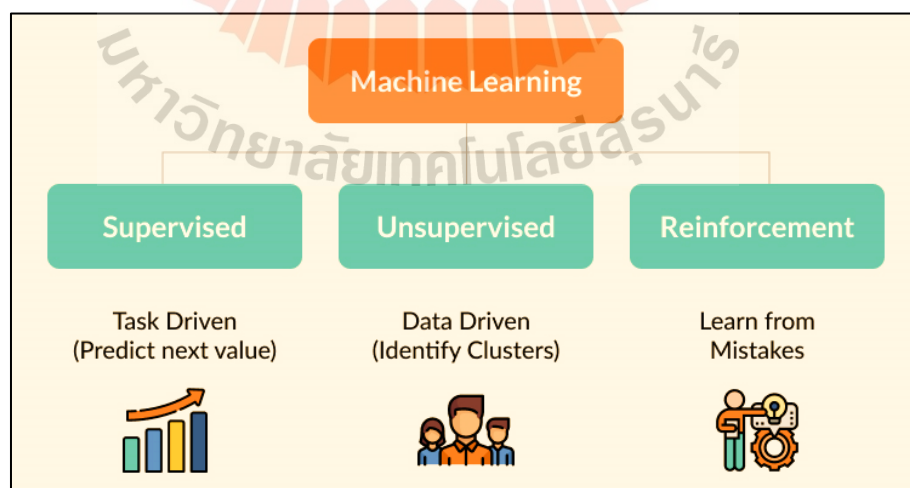
การเรียนรู้ของเครื่องจักร หรือ ML นั้น ถูกใช้งานเหมือนกับสมองของ AI หรืออีกอย่างคือ AI นั้นใช้ ML สำหรับการเรียนรู้ให้ฉลาดขึ้น การที่ AI จะฉลาดหรือทำงานได้เก่งขึ้นนั้นขึ้นอยู่กับการเรียนรู้ของเครื่องจักร นั้น ซึ่งจะแตกต่างกับการที่มนุษย์เขียนแล้วส่งโปรแกรมขึ้นมาด้วยเงื่อนไขที่กำหนดไว้โปรแกรมจะไม่สามารถเรียนรู้เองหรือเรียนรู้เพิ่มเติมได้แบบที่ AI ทำ โดยความแตกต่างระหว่างการเรียนรู้ของเครื่องจักรเทียบกับโปรแกรมทั่วไปดังรูปที่ 2.6



รูปที่ 2.6 ความแตกต่างระหว่าง Machine learning กับ Programming

ML เรียนรู้จากข้อมูลที่ได้ส่งเข้าไปฝึกสอน (Training input) แล้วจดจำไว้และส่งผลลัพธ์ออกมาเป็นตัวเลขหรือโค้ดเพื่อส่งต่อไปให้ AI นำไปแสดงการกระทำโดยที่ ML นั้น สามารถนำไปใช้งานได้หลายรูปแบบโดยจะอาศัยกลไกของโปรแกรมหรือเรียกว่าอัลกอริทึม (Algorithm) ที่มีให้เลือกใช้งานหลายแบบซึ่งถูกออกแบบโดยนักวิทยาศาสตร์ข้อมูล (Data scientist) เป็นผู้ออกแบบอัลกอริทึม การเรียนรู้ของเครื่องจักรสามารถแบ่งออกได้เป็น 3 ประเภทการเรียนรู้ดังรูปที่ 2.7

1. การเรียนรู้แบบมีข้อมูลสอน (Supervised Learning)
2. การเรียนรู้แบบไม่มีข้อมูลสอน (Unsupervised Learning)
3. การเรียนรู้แบบตามสภาพแวดล้อม (Reinforcement Learning)

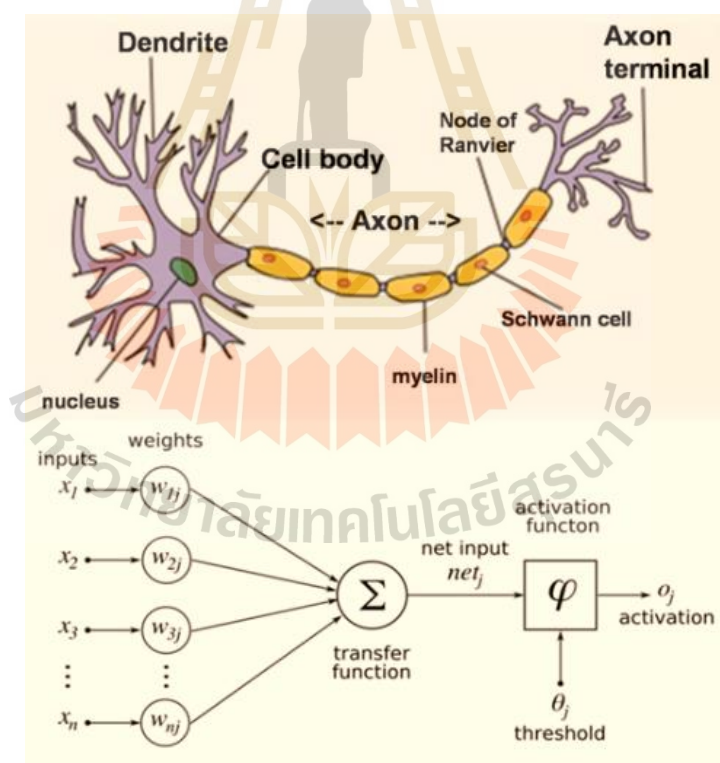


รูปที่ 2.7 ประเภทการเรียนรู้ของเครื่องจักร (อ้างอิง [www.n-ix.com](http://www.n-ix.com))

## 2.5 โครงข่ายประสาทเทียม (Artificial Neural Networks : ANNs)

โครงข่ายประสาทเทียมเป็นโมเดลทางคณิตศาสตร์ด้วยการคำนวณแบบคอนเนกชันนิสต์ (Connectionist) ใช้สำหรับประมวลผลเพื่อจำลองการทำงานของเครือข่ายประสาทของสมองมนุษย์ ให้สามารถสร้างความสามารถในการเรียนรู้แบบจดจำรูปแบบ (Pattern recognition) ให้เหมือนกับในสมองของมนุษย์

แนวคิดในการเรียนแบบโครงข่ายประสาทของมนุษย์นั้นเริ่มมาจากการศึกษาโครงข่ายไฟฟ้าชีวภาพ (Bioelectric Network) ในสมองสำหรับการรับรู้สิ่งต่าง ๆ ได้นั้นจะต้องมี Dendrite เปรียบเหมือน input layer รับข้อมูลเข้ามามากจากนั้นจะใช้ Soma เปรียบเหมือน hidden layer เป็นตัวประมวลผลข้อมูลที่ได้รับเข้ามาและ Axon จะทำหน้าที่แปลงข้อมูลให้เป็นผลลัพธ์ออกมาจากนั้นจะแสดงผลลัพธ์นั้นด้วย Synapses เปรียบเหมือน output layer ซึ่งโครงข่ายประสาทเทียมจะสามารถทำให้คอมพิวเตอร์รู้จักการเรียนรู้ จดจำ และตัดสินใจคล้ายกับโครงข่ายประสาทของสมองมนุษย์ ดังตัวอย่างรูปที่ 2.8



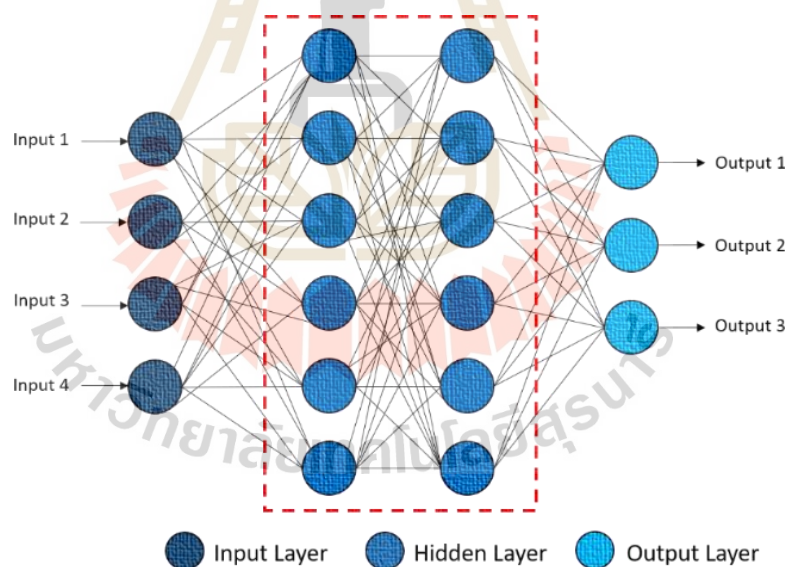
รูปที่ 2.8 การเรียนรู้ของโครงข่ายประสาทของมนุษย์เทียบกับโครงข่ายประสาทเทียม

### 2.5.1 ส่วนประกอบของโครงข่ายประสาท (Neural Network)

1. Input layer คือ layer ชั้นแรกของการรับข้อมูลเข้าสู่โครงข่ายประสาทเทียม โดยจะมีโหนดรับข้อมูลตามจำนวน input ที่มีเช่น หากจะต้องทำการคัดแยกผลไม้ข้อมูลที่เตรียมมาก็จะต้องมีภาพของผลไม้ต่าง ๆ เป็น input ที่ป้อนเข้าสู่โครงข่ายประสาทเทียมต่อไปหรืออาจเรียกว่า Feature

2. Hidden layer คือ layer ชั้นกลางซึ่งอาจจะมีมากกว่าหนึ่งชั้นโดยที่ hidden layer นั้น จะมีผลต่อประสิทธิภาพการเรียนรู้ของโมเดลยิ่งมี hidden layer มากความซับซ้อนของโมเดลก็จะมากขึ้นตามไปด้วยแน่นอนว่าจะส่งผลต่อการทำงานด้วยโดยการใช้งานนั้น ขึ้นอยู่กับผู้สร้างโมเดลว่าต้องการใช้จำนวนเท่าไร

3. Output layer คือ layer สุดท้ายของโครงข่ายประสาทเทียมซึ่งทำหน้าที่ในการเอาข้อมูลที่ได้จากการคำนวณหรือสังเคราะห์มาแล้วจาก hidden layer มาแสดงเป็นผลลัพธ์ที่ต้องการโดยที่โหนดของ output layer นั้น ขึ้นอยู่กับรูปแบบ output ที่ผู้สร้างต้องการโดยโครงสร้างภาพรวมของโครงข่ายประสาทเทียมแสดงดังรูปที่ 2.9

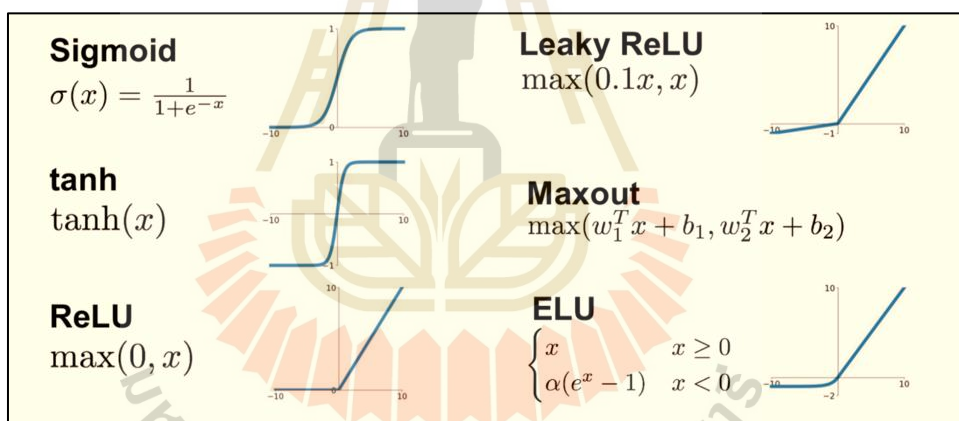


รูปที่ 2.9 โครงสร้างของโครงข่ายประสาทเทียม

### 2.5.2 ส่วนประกอบของโครงข่ายประสาทเทียม (ANN components)

1. ข้อมูลขาเข้า (Input) เป็นข้อมูลที่เป็นตัวเลข หากเป็นข้อมูลเชิงคุณภาพต้องแปลงให้อยู่ในรูปเชิงปริมาณที่โครงข่ายประสาทเทียมยอมรับได้

2. ข้อมูลส่งออก (Output) คือ ผลลัพธ์ที่เกิดขึ้นจริง (actual output) จากกระบวนการเรียนรู้ของโครงข่ายประสาทเทียม
3. ค่าน้ำหนัก (weight) คือสิ่งที่ได้จากการเรียนรู้ของโครงข่ายประสาทเทียม หรือเรียกอีกอย่างว่า ความรู้ (Knowledge) ค่านี้จะถูกเก็บเป็นทักษะเพื่อใช้ในการจดจำข้อมูลอื่น ๆ ที่อยู่ในรูปแบบเดียวกัน
4. ฟังก์ชันผลรวม (Summation function : S) เป็นผลรวมของข้อมูลป้อนเข้า โดยจะนำอินพุตคูณกับค่าน้ำหนักของแต่ละขา ผลที่ได้จากอินพุตทุก ๆ ขาของ neuron จะเอมารวมกัน
5. ฟังก์ชันการแปลง (Transfer function) เป็นการคำนวณการจำลองการทำงานของโครงข่ายประสาทเทียม ซึ่ง activation function ที่นิยมนำมาใช้คือ ซิกมอยด์ฟังก์ชัน (sigmoid function) ไฮเปอร์โบลิกแทนเจนต์ฟังก์ชัน (hyperbolic tangent function) และ Rectified Linear Unit (ReLU) แสดงให้เห็นในรูปที่ 2.10



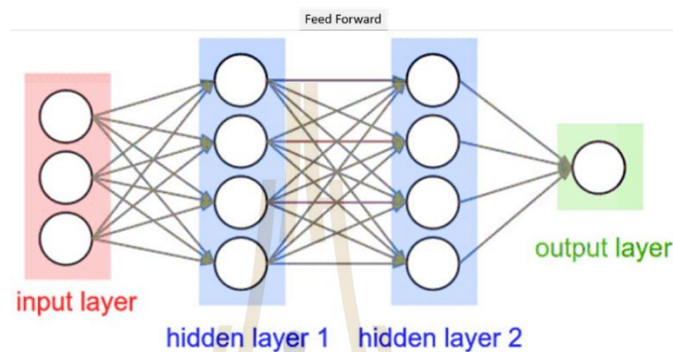
รูปที่ 2.10 Activation function

### 2.5.3 หลักการทำงานของโครงข่ายประสาทเทียม

#### 2.5.3.1 เครือข่ายการส่งข้อมูลไปข้างหน้า (Feed forward network)

หลักการทำงานของโครงข่ายประสาทเทียมแบบเครือข่ายการส่งข้อมูลไปข้างหน้าคือ เมื่อมีข้อมูลเข้ามาในเครือข่ายจะนำ input นั้น มาคูณกับ weight ของแต่ละขาซึ่งผลที่ได้จากทุก ๆ input ของแต่ละ neuron ก็จะเอมารวมกันแล้วก็เอามาเทียบกับ threshold ที่กำหนดไว้ ถ้าผลรวมมีค่ามากกว่า threshold แล้ว neuron ก็จะส่ง output ออกไป output นี้ก็จะถูกส่งไปยัง input ของ neuron ต่อไปที่เชื่อมกันใน network ถ้าค่าน้อยกว่า threshold ก็จะไม่เกิด output ขึ้น

สิ่งสำคัญคือต้องทราบค่า weight และ threshold สำหรับเป้าหมายที่เราต้องการเพื่อให้คอมพิวเตอร์ รู้จำ ซึ่งเป็นค่าที่ไม่แน่นอน แต่สามารถกำหนดให้คอมพิวเตอร์ปรับค่าเหล่านั้นได้โดยการสอนให้มัน รู้จักรูปแบบของสิ่งที่เราต้องการให้คอมพิวเตอร์รู้จำ ซึ่งปกติ Neural Network จะเรียกว่า Feed Forward คือการไปข้างหน้า กล่าวคือ Neural Network จะรับข้อมูลจาก Input Layer สร้างน้ำหนัก (W) แล้วประมวลผลผ่าน Hidden Layer แล้วออกที่ Output ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 เครือข่ายการส่งข้อมูลไปข้างหน้าของโครงข่ายประสาทเทียม

การคำนวณหาค่า output หาได้จากสมการที่ 2.1

$$y = f(Wx + b) \quad (2.1)$$

โดยที่

$$x^T = [x_1 \ x_2 \ \dots \ x_s] \quad \text{คือ Input matrix}$$

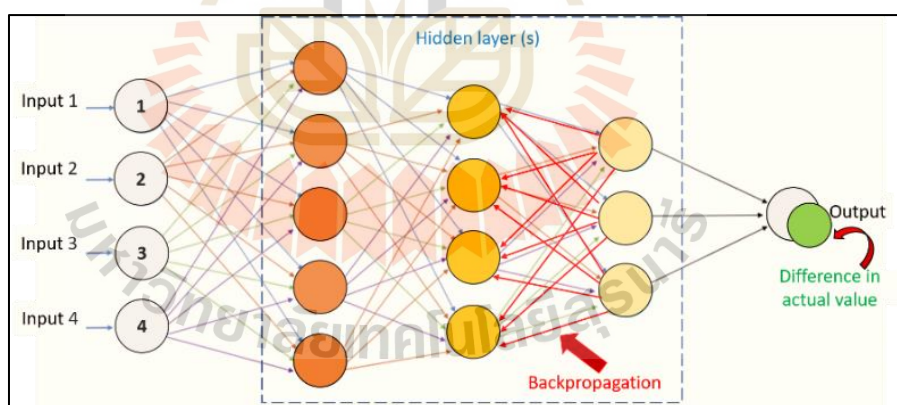
$$b^T = [b_1 \ b_2 \ \dots \ b_R] \quad \text{คือ Bias matrix}$$

$$W = \begin{pmatrix} W_{11} & K & W_{1R} \\ M & O & M \\ W_{S1} & L & W_{SR} \end{pmatrix} \quad \text{คือ Weight matrix}$$



### 2.5.3.2 เครือข่ายการส่งข้อมูลย้อนกลับ (Back-propagation network)

หลักการทำงานของโครงข่ายประสาทเทียมแบบเครือข่ายการส่งข้อมูลย้อนกลับคือ เมื่อมีข้อมูล input เข้ามา network ก็จะเอา input นั้น มาคูณกับ weight ของแต่ละขาของเครือข่ายประสาทเทียมผลที่ได้จาก input ทุก ๆ ขาของ neuron จะเอามารวมกันแล้วก็เอามาเทียบกับ threshold ที่กำหนดไว้ ถ้าผลรวมมีค่ามากกว่า threshold แล้ว neuron ก็จะส่ง output ออกไป output นี้ก็จะถูกส่งไปยัง input ของ neuron อื่น ๆ ที่เชื่อมกันในโครงข่ายประสาทเทียม ซึ่งเป็นกระบวนการย้อนกลับของการรู้จำในการฝึก feed-forward Neural Networks แต่ถ้าหากให้มันย้อนกลับจาก output กลับมาที่ input จะเรียกว่า "back-propagation" จะมีการใช้อัลกอริทึมแบบ back-propagation เพื่อใช้ในการปรับปรุงน้ำหนักคะแนนของเครือข่าย (Network Weight) หลังจากใส่รูปแบบข้อมูลสำหรับฝึกให้แก่เครือข่ายในแต่ละครั้งแล้ว ค่าที่ได้รับ (output) จากเครือข่ายจะถูกนำไปเปรียบเทียบกับผลที่คาดหวัง แล้วทำการคำนวณหาค่าความผิดพลาด (Error Estimation) ซึ่งค่าความผิดพลาดนี้จะถูกส่งกลับเข้าสู่เครือข่ายเพื่อใช้แก้ไขค่าน้ำหนักคะแนนต่อไป โครงข่ายประสาทเทียมที่มีการสร้างโมเดลแบบการส่งข้อมูลย้อนกลับคือจะมีการนำค่าความผิดพลาดของ output ของโมเดลกับ output จริงย้อนกลับเพื่อทำการปรับค่าถ่วงน้ำหนักใหม่เพื่อให้โมเดลเกิดความผิดพลาดน้อยที่สุด แสดงให้เห็นดังรูปที่ 2.12



รูปที่ 2.12 แบบจำลองโครงข่ายประสาทเทียมแบบเครือข่ายการส่งข้อมูลย้อนกลับ

สมการค่าความผิดพลาดของการสร้างโมเดลสามารถได้จากค่าเฉลี่ยกำลังสอง (Mean Square Error, MSE) ดังสมการที่ 2.2

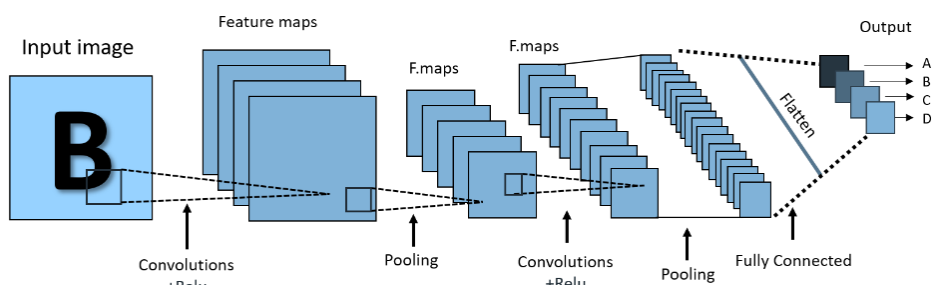
$$Error = \frac{1}{2} (y_{target} - y_{output})^2 \quad (2.2)$$

## 2.6 การเรียนรู้เชิงลึก (Deep Learning : DL)

การเรียนรู้เชิงลึกหรือ deep learning นั้น เป็นส่วนหนึ่งของการเรียนรู้ของเครื่องจักรโดยที่ deep learning นั้นจะเน้นการเรียนรู้เชิงลึกด้วยข้อมูลที่มีอยู่เป็นจำนวนมากเพื่อสร้างแบบจำลองโครงข่ายประสาทเทียมขึ้นมาเพื่อให้สามารถทำนายหรือหาผลลัพธ์ได้อย่างแม่นยำ deep learning นั้นจะใช้อัลกอริทึมที่เรียกว่า neural network อัลกอริทึมโดยที่ Deep Learning นั้นคือการที่มีโครงข่ายประสาทเทียมมากกว่า 2 layer ขึ้นไปจึงจะเรียกว่าการเรียนรู้เชิงลึกหากโครงข่ายประสาทเทียมมี hidden layer เพียงหนึ่งชั้นจะเรียกว่า โครงข่ายประสาทเทียมแบบง่าย (Single layer perceptron network) สำหรับ deep learning นั้นเรียกว่า โครงข่ายประสาทเทียมแบบซับซ้อน (Multi-layer perceptron network) จะมี hidden layer ตั้งแต่สองชั้นขึ้นไปดังที่แสดงในรูปที่ 2.9 เพื่อให้การประมวลผลเพื่อหาผลลัพธ์ได้ซับซ้อนมากยิ่งขึ้นละเอียดมากขึ้น สามารถแยกแยะข้อมูลที่ต้องการสอนให้เครื่องจักรรู้ได้แม่นยำยิ่งขึ้น

## 2.7 แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network : CNN)

โครงข่ายประสาทเทียมแบบคอนโวลูชันเป็นโครงข่ายประสาทเทียมที่จำลองการมองเห็นของมนุษย์ที่มองเห็นเป็นพื้นที่ย่อย ๆ และนำกลุ่มพื้นที่ย่อย ๆ ที่ CNN มองนั้นมารวมเข้าด้วยกันโดยที่จะมีประสิทธิภาพในการจำแนกรูปภาพ โดยในปัจจุบันมีการใช้งานอย่างแพร่หลายและยังมีแนวคิดเกี่ยวกับโครงข่ายประสาทเทียมแบบคอนโวลูชันย่อยลงไปอีกโดยจะแยกย่อยตามการใช้งานของแต่ละโมเดลตามที่ผู้ใช้งานต้องการจึงมีการออกแบบสถาปัตยกรรมสำหรับเป้าหมายงานต่าง ๆ ตามแต่วัตถุประสงค์เพื่อดึงลักษณะเด่นของภาพที่ผู้ใช้งานสนใจออกมาเรียกว่า Feature extraction โดยสถาปัตยกรรมของโครงข่ายประสาทเทียมแบบคอนโวลูชันแสดงดังรูปที่ 2.13



รูปที่ 2.13 สถาปัตยกรรมของโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN)

## 2.7.1 หลักการทำงานของโครงข่ายประสาทเทียมแบบคอนโวลูชัน

หลักการทำงานของ CNN จะมีทั้งหมด 6 ขั้นตอนดังนี้

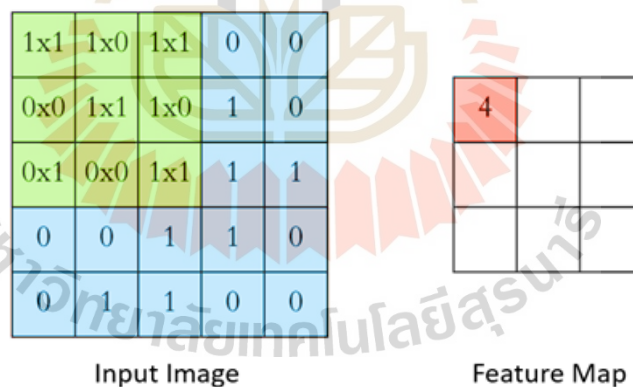
### 1. Convolutional layer

#### 1.1 การดึงคุณลักษณะเด่น (Feature Extraction)

การดึงคุณลักษณะเด่นมีแนวคิดการคำนวณมาจากหลักการคอนโวลูชันเชิงพื้นที่ (Spatial convolution) โดยที่การทำงานเพื่อดึงคุณลักษณะเด่นนั้นเริ่มจากการกำหนดค่าตัวกรอง (filter) หรือเรียกอีกอย่างว่าเคอร์เนล (kernel) ทำหน้าที่ช่วยดึงคุณลักษณะที่สำคัญของภาพนั้นออกมา หนึ่งตัวกรองจะสามารถดึงคุณลักษณะเด่นออกมาได้เพียงหนึ่งอย่างเท่านั้น ดังนั้นจำเป็นต้องมีตัวกรองหลายตัวเพื่อหาคุณลักษณะเด่นจากพื้นที่ทั้งหมด

#### 1.2 ลักษณะของตัวกรอง (Filter)

ในภาพดิจิทัลตัวกรองจะเป็นตารางสองมิติที่มีขนาดตามพื้นที่ย่อย ๆ ที่อยากพิจารณา โดยใช้ตัวกรอง (filter) หรือ เคอร์เนล (kernel) วางทับลงในพิกเซลแรกของภาพข้อมูลเข้า จากนั้น จะถูกเลื่อนไปทับบนพิกเซลอื่นในภาพทีละพิกเซลจนครบทุกพิกเซลในภาพ เมื่อเลื่อนตัวกรองไปเรื่อย ๆ จนครบทุกพิกเซลที่สามารถเลื่อนได้ในภาพแล้ว จะได้เป็นผังคุณลักษณะ (feature map) แสดงให้เห็นดังรูปที่ 2.14



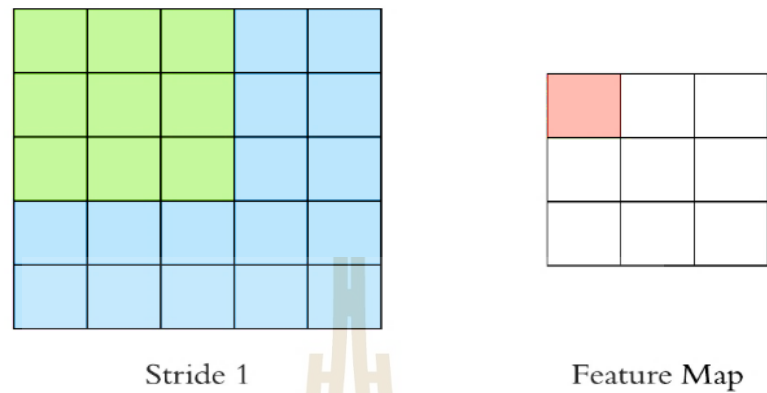
รูปที่ 2.14 ลักษณะตัวกรอง

(ภาพจาก <https://blog.datawow.io/>)

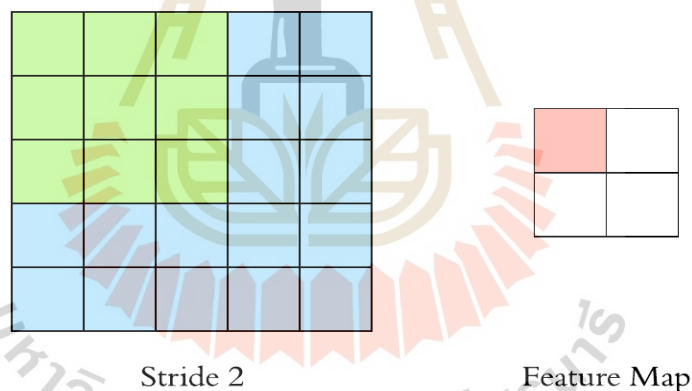
#### 1.3 Stride

คือการกำหนดค่าเพื่อให้ตัวกรองเลื่อนไปบนรูปภาพที่สนใจด้วยค่า step ที่กำหนดขึ้น (กำหนด Stride เท่ากับ 1) แสดงให้เห็นดังรูปที่ 2.15 และสามารถกำหนดค่าของ Stride

ให้มากขึ้นได้ แต่อาจจะทำให้ฟังก์ชันลักษณะ (feature map) มีขนาดที่เล็กลง แสดงให้เห็นดังรูปที่ 2.16 เพื่อให้การคำนวณคุณลักษณะมีพื้นที่ทับซ้อนกันน้อยลง



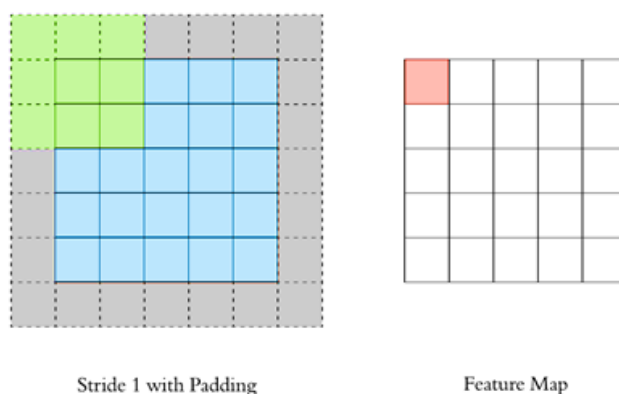
รูปที่ 2.15 กำหนดเลื่อนตัวกรองเท่ากับ 1  
(ภาพจาก <https://blog.datawow.io/>)



รูปที่ 2.16 กำหนดเลื่อนตัวกรองเท่ากับ 2  
(ภาพจาก <https://blog.datawow.io/>)

#### 1.4 Padding

Padding เป็นการเพิ่มพื้นที่โดยรอบของรูปภาพที่กำลังประมวลผลขึ้น เพื่อให้การดึงคุณลักษณะเด่นของภาพนั้นออกมาได้อย่างถูกต้องกรณีที่มีวัตถุที่สนใจอยู่บริเวณขอบของภาพโดยจะเพิ่มพื้นที่พิกเซลดังแสดงในรูปที่ 2.17 จะเห็นพื้นที่สีเทาขอบ ๆ รูปภาพเรียกว่าการสร้าง Padding โดยอาจจะกำหนดเป็น 0 หรือค่าต่าง ๆ เท่า ๆ กันเข้าไปเพื่อให้ Feature map นั้นยังมีขนาดเท่ากับ Input



รูปที่ 2.17 การทำ padding

(ภาพจาก <https://blog.datawow.io/>)

## 2. Detector stage

คือการรับข้อมูล output จาก convolutional layer แล้วแปลงค่า output ที่รับมาให้อยู่ในรูปของ non-linear โดยการใช้ activation function เข้ามาช่วยเช่น Rectified Linear Units (ReLU) ซึ่งทำให้โมเดลง่ายต่อการคำนวณและมีผลลัพธ์ที่มีประสิทธิภาพ

## 3. Pooling layer

เป็น layer ที่ทำหน้าที่ในการปรับปริมาณและขนาดของข้อมูลที่ได้รับมาจาก Convolutional layer ให้น้อยลงก่อนส่งไปยัง layer ถัดไปโดยยังคงลักษณะเด่นของรูปภาพไว้ได้อย่างครบถ้วนโดยการทำให้ Pooling นั้นมีความนิยมอยู่สองรูปแบบคือ Max pooling และ Average pooling ซึ่งจะนิยมเรียกกันว่า pool size คือ ค่าคุณลักษณะที่เด่นที่สุดจะถูกเก็บไว้จำนวนต่อ โดยที่รายละเอียดของ input ยังครบถ้วนเหมือนเดิม แต่อาจได้ output ที่ได้จะมีขนาดเล็กลง ซึ่งขึ้นอยู่กับ stride ที่กำหนด สามารถประมวลผลได้รวดเร็วขึ้น และแก้ปัญหา overfitting ได้ ดังรูปที่ 2.18

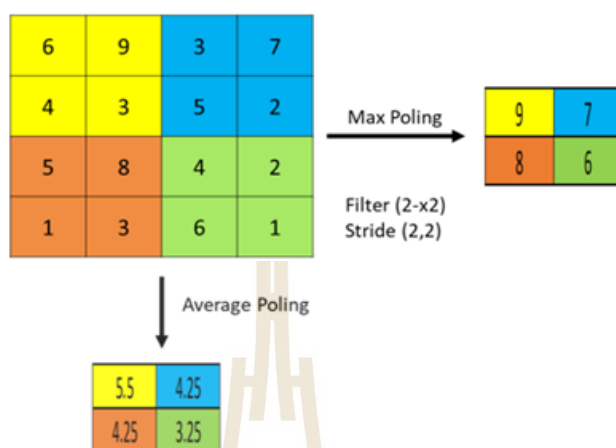
### 3.1 Max Pooling

เป็นตัวกรองแบบหนึ่งที่ทำค่าสูงสุดในบริเวณที่ตัวกรองทาบอยู่มาเป็นผลลัพธ์ โดยจะต้องเตรียมตัวกรองในลักษณะเดียวกับการทำ Feature Extraction ของ CNN มาทาบบนข้อมูลแล้วเลือกค่าที่สูงที่สุดบนตัวกรองนั้น มาเป็นผลลัพธ์ใหม่ และจะเลื่อนตัวกรองไปตาม Stride ที่กำหนด

### 3.2 Mean Pooling

เป็นตัวกรองแบบหนึ่งที่ทำค่าเฉลี่ยในบริเวณที่ตัวกรองทาบอยู่มาเป็นผลลัพธ์ โดยจะต้องเตรียมตัวกรองในลักษณะเดียวกับการทำ Feature Extraction ของ CNN มาทาบ

บนข้อมูลแล้วนำค่าเฉลี่ยของตัวกรองนั้นมาเป็นผลลัพธ์ใหม่ และจะเลื่อนตัวกรองไปตาม Stride ที่กำหนด



รูปที่ 2.18 การทำ Max pooling และ Mean pooling

#### 4. Flatten

คือการทำให้ข้อมูลของ pool size ซึ่งอยู่ในรูปแบบ 3 มิติมาแปลงให้เป็น 1 มิติ เพื่อสร้างเป็น input ที่จะส่งให้กับโครงข่ายประสาทเทียม (Artificial Neural Network) โดยจะประกอบไปด้วย input layer, hidden layer, output layer

#### 5. Fully connected layer

รับข้อมูลจาก Flatten มาเพื่อสร้างเป็นโครงข่ายประสาทเทียมแบบซับซ้อน (Multi-layer perceptron network) เพื่อประมวลผลให้สามารถจำแนกหรือทำนายได้ว่าข้อมูล input ที่รับเข้ามานั้นคืออะไรจากนั้นจะส่งข้อมูลที่ไปยัง output layer ต่อไป

#### 6. Classification

ในขั้นตอนนี้จะเป็นการจำแนกหรือจัดกลุ่มหลังจากผ่านกระบวนการต่าง ๆ โดยจะใช้ Softmax activation function ในการวิเคราะห์ข้อมูล เพื่อที่จะจำแนกว่า output คืออะไร

## 2.8 การตรวจจับวัตถุ (Object detection)

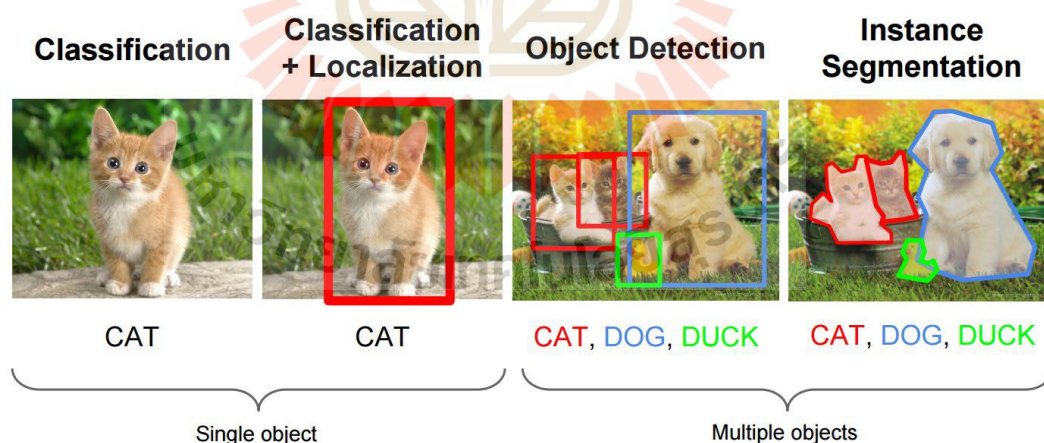
การตรวจจับวัตถุ คือการที่จะให้คอมพิวเตอร์มีความสามารถในการจำแนกวัตถุที่อยู่ในภาพออกมาแล้วทำนายว่าวัตถุนั้นเป็น Class อะไรต้องอาศัยหลักการของ Object classification และการที่จะให้คอมพิวเตอร์หาตำแหน่งของวัตถุที่อยู่ในภาพว่าอยู่ตำแหน่งไหนของภาพต้องอาศัยหลักการของ Object detection ดังแสดงในรูปที่ 2.19 สำหรับการทำให้ Object classification จะ

สรุปได้ดังนี้ Input image>Feature extractor>Classifier>Output Class โดยมีขั้นตอนดังนี้ นำเข้าข้อมูลเพื่อสกัดหาคุณลักษณะที่สำคัญของรูปภาพ แล้วทำการ Classifier เพื่อให้ได้ Output ของ class ขั้นตอนถัดมาเข้าสู่กระบวนการ Object detection

Algorithm ที่เป็นที่ยอมรับสำหรับการตรวจจับวัตถุภายในภาพมีทั้งหมด 8 algorithm ดังนี้

1. Fast R-CNN
2. Faster R-CNN
3. Histogram of Oriented Gradients (HOG)
4. Region-based Convolutional Neural Networks (R-CNN)
5. Region-based Fully Convolutional Network (R-FCN)
6. Single Shot Detector (SSD)
7. Spatial Pyramid Pooling (SPP-net)
8. You Only Look Once (YOLO)

โดยที่ทุกอัลกอริทึมที่กล่าวมานั้นล้วนแล้วแต่เป็นอัลกอริทึมด้าน object detection ทั้งสิ้น สำหรับงานวิจัยนี้จะใช้งาน YOLO Algorithm สำหรับตรวจจับวัตถุภายในภาพเนื่องจากมีความรวดเร็วในการตรวจจับและมีความแม่นยำที่สูง

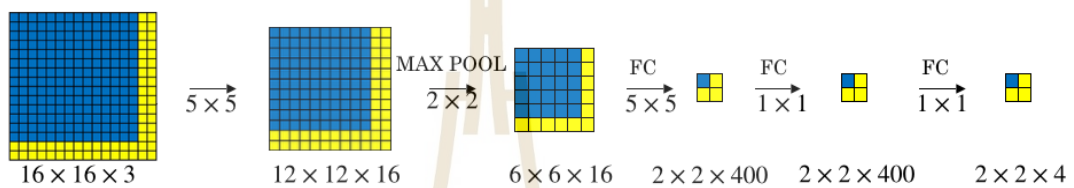


รูปที่ 2.19 ประเภทของ Object detection

(อ้างอิง [http://cs231n.stanford.edu/slides/winter1516\\_lecture8.pdf](http://cs231n.stanford.edu/slides/winter1516_lecture8.pdf))

### 2.8.1 Sliding Window

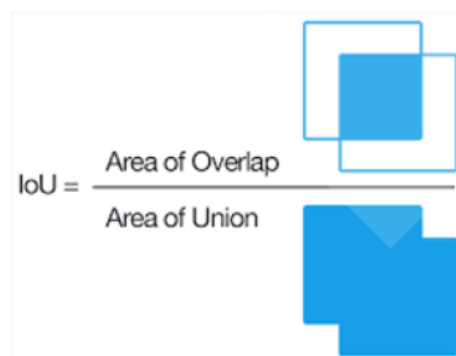
เป็นวิธีการทำ Object Detection ที่นิยมมากในสมัย 10 ปีที่ผ่านมามีการทำงานจะนำเอาผลลัพธ์จากการทำ Object Classification ดังนี้ ทำการตีกรอบแล้วขยับไปเรื่อย ๆ จากนั้นภาพที่ได้จากการขยับเข้าไปทำ Feature Extractor และทำการ Classifier ว่าตรงกับ Class อาจจะมีการขยับทีละ 1 pixel ไปเรื่อย ๆ จนหมด จะเกิดปัญหาขึ้นเพราะถ้าเรากำหนดกรอบ ขนาด (32X32) และมีภาพขนาด (512X512) เวลาทำ Sliding Window จะต้องใช้พลังงานของเครื่องมหาศาลถึงจะเสร็จ ซึ่งในปัจจุบันมีการแก้ปัญหาแล้ว มีการย่อขยายรูปทุกครั้งที่มีการทำงานเสร็จดังรูปที่ 2.20



รูปที่ 2.20 Sliding window  
(อ้างอิง <https://medium.com>)

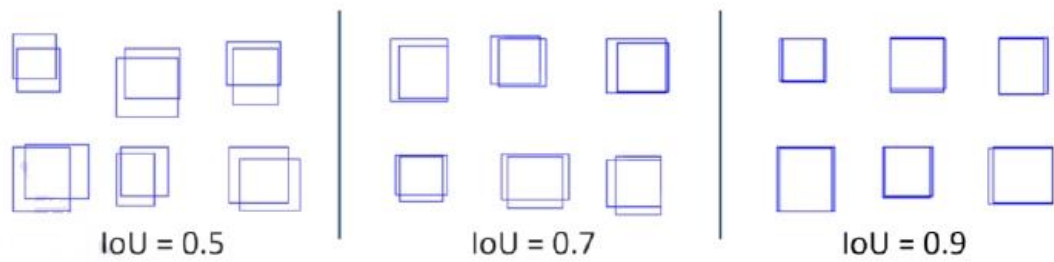
### 2.8.2 การวัดผล

Intersection over Union (IoU) เป็นการวัดประสิทธิภาพของของโมเดลเป็นที่นิยมใน การทำ Object detection ซึ่งหาได้จาก อัตราส่วนระหว่าง area ที่เป็น intersection ของ 2 bounding box ทหารด้วย area รวมของกรอบทั้งสอง หรืออาจจะเรียกว่า ดัชนี Jaccard เป็นหลักวิธีการในการหาจำนวนเปอร์เซ็นต์ที่ทับซ้อนกันระหว่างผลเฉลย (Ground Truth) และผลจากการทำนาย (Predict) ถ้าได้ค่า IoU ที่มากกว่า 0.5 ถึงว่ายอมรับได้ ตามรูปที่ 2.21 และ รูปที่ 2.22



รูปที่ 2.21 Area of IoU





รูปที่ 2.22 ตัวอย่างการซ้อนทับ

ในการวัดผลหรือประเมินประสิทธิภาพของโมเดลนั้น สำหรับโครงข่ายประสาทเทียมแบบคอนโวลูชัน หรือ การตรวจจับวัตถุภายในภาพนั้น สามารถใช้การประเมินแบบ Confusion matrix หรือค่า F1-Score ในการประเมินโดยจะมีตารางแสดงการคำนวณดังรูปที่ 2.23

|              |          | Predicted Class                            |  |  |
|--------------|----------|--|--|--|
|              |          | Positive                                   | Negative   |  |
| Actual Class | Positive | True Positive (TP)                         | False Negative (FN)<br>Type II Error                       | <b>Sensitivity</b><br>$\frac{TP}{(TP + FN)}$             |
|              | Negative | False Positive (FP)<br>Type I Error        | True Negative (TN)   | <b>Specificity</b><br>$\frac{TN}{(TN + FP)}$             |
|              |          | <b>Precision</b><br>$\frac{TP}{(TP + FP)}$ | <b>Negative Predictive Value</b><br>$\frac{TN}{(TN + FN)}$ | <b>Accuracy</b><br>$\frac{TP + TN}{(TP + TN + FP + FN)}$ |

รูปที่ 2.23 ตาราง Confusion matrix

|                     |     |  |
|---------------------|-----|--|
| True Positive (TP)  | คือ | สิ่งที่โปรแกรมทำนายว่าจริงและความเป็นจริงคือจริง                       |
| True Negative (TN)  | คือ | สิ่งที่โปรแกรมทำนายว่าไม่จริงและความเป็นจริงคือไม่จริง                 |
| False Positive (FP) | คือ | สิ่งที่โปรแกรมทำนายว่าจริงแต่ความเป็นจริงคือไม่จริง                    |
| False Negative (FN) | คือ | สิ่งที่โปรแกรมทำนายว่าไม่จริงแต่ความเป็นจริงคือจริง                    |
| Accuracy            | คือ | ค่าความถูกต้องที่ทำนายได้ตรงกับสิ่งที่เกิดขึ้นจริงหาได้จากสมการที่ 2.3 |

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (2.3)$$

Precision เป็นการเปรียบเทียบการทำนายที่ถูกต้องว่าจริง และก่เกิดขึ้นจริง (TP) กับ การทำนายว่าจริง แต่สิ่งที่เกิดขึ้นคือ ไม่จริง (FP)

Precision หาได้จากสมการที่ 2.4

$$Precision = \frac{TP}{TP + FP} \quad (2.4)$$

Recall คือค่าความถูกต้องของการทำนายว่าจะเป็นจริง เทียบกับจำนวนครั้งของเหตุการณ์ทั้งทำนายและเกิดขึ้นว่า เป็นจริง

Recall หาได้จากสมการที่ 2.5

$$Recall = \frac{TP}{TP + FN} \quad (2.5)$$

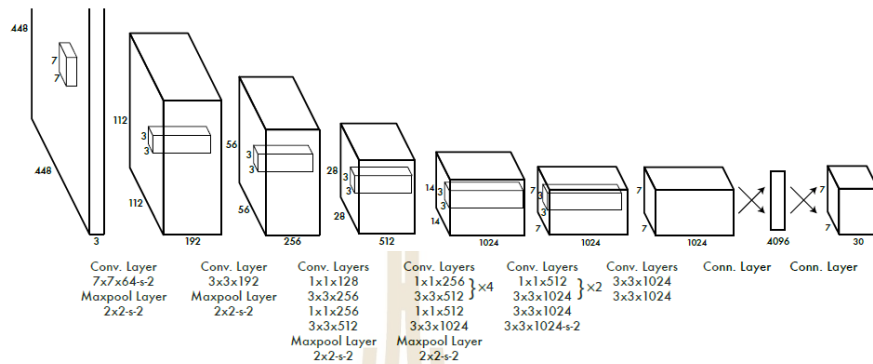
F1-Score คือค่าเฉลี่ยแบบ harmonic mean ระหว่าง Precision และ Recall เพื่อใช้สำหรับวัดประสิทธิภาพของโมเดลแบบ single matrix

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \quad (2.6)$$

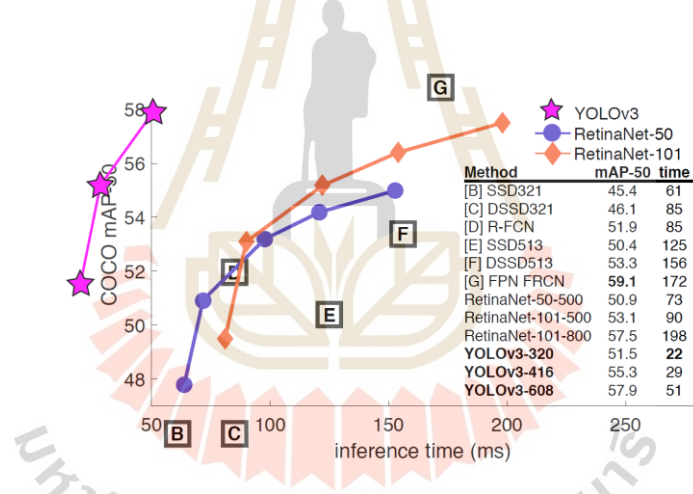
## 2.9 YOLOv5 (You Only Look Once Ver.5)

YOLO หรือ You Only Look Once คือหนึ่งใน Algorithm ที่พัฒนามาจาก CNN ซึ่งเป็นหนึ่งใน Algorithm ของ Deep learning มีการปรับปรุงประสิทธิภาพให้มีการทำงานที่รวดเร็วและยังคงความแม่นยำโดยในปัจจุบัน YOLO ถือได้ว่าเป็น Algorithm ของ Deep learning ที่ทำงานได้เร็วที่สุดแล้วดังรูปที่ 2.25 ที่แสดงให้เห็นถึงความรวดเร็วในการตรวจจับองค์ประกอบภายในภาพ

เทียบกับอัลกอริทึมอื่น ๆ และสามารถตรวจจับวัตถุได้แบบ Real-time โดยมีโครงสร้างของ Algorithm ดังรูปที่ 2.24



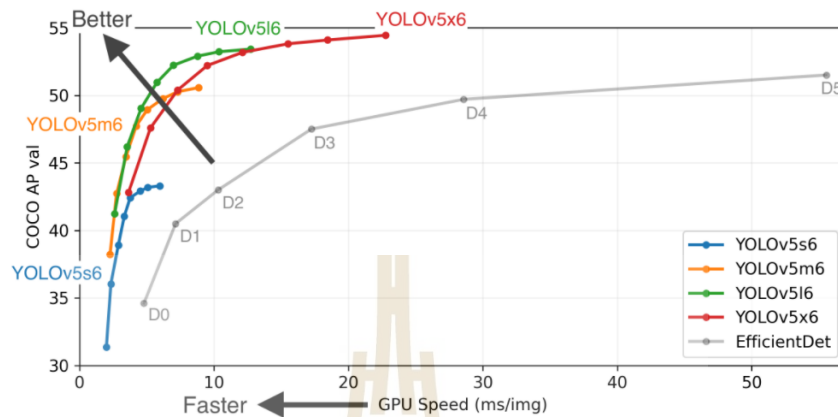
รูปที่ 2.24 โครงสร้างของ YOLOv3



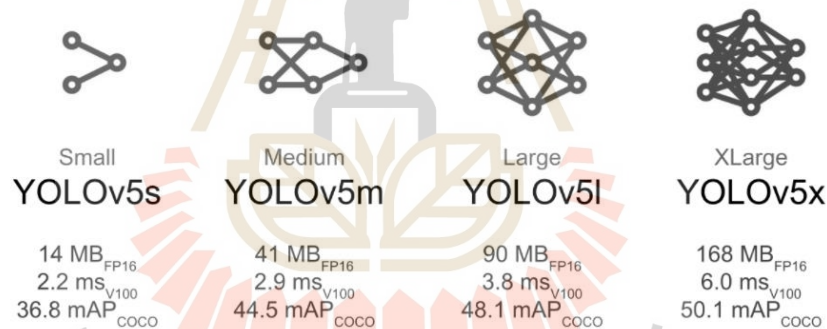
รูปที่ 2.25 YOLOv3 เปรียบเทียบกับ Algorithm อื่นช่วงเวลานั้น

ในปัจจุบัน YOLO ได้พัฒนามาถึงเวอร์ชัน 5 แล้วซึ่งพัฒนามาจาก YOLO เวอร์ชัน 4 ที่เปิดตัวและตีพิมพ์ได้ไม่นานYOLOv5 นั้น สามารถนำมาประยุกต์ใช้งานได้ง่ายกว่า YOLOv4 และมีการปรับปรุงประสิทธิภาพการทำงานให้ดีขึ้นจากเวอร์ชันเดิมและยังคงรักษาการทำงานที่มีเอกลักษณ์ด้านความเร็วในการประมวลผลตรวจจับวัตถุแบบ Real-time เหตุผลที่ YOLO นั้นมีความรวดเร็วในการทำงานมากกว่าโมเดลประเภท Region based อื่น ๆ เนื่องจากโครงสร้างโมเดลของ YOLO นั้นเป็นแบบ pass through image ภายในรอบเดียวโดยผ่าน CNN เรื่อย ๆ และพยายามทำนายหาตำแหน่งของวัตถุที่เราสนใจภายในภาพและประเภทของวัตถุจาก feature map โดยที่ YOLOv5 นั้นจะมี

Algorithm ให้เลือกใช้ให้เหมาะสมกับการใช้งานของโปรเจค โมเดลขนาดเล็กรหัส S ไปจนถึงโมเดลขนาดใหญ่รหัส X โดยเปรียบเทียบประสิทธิภาพในแต่ละโมเดลดังรูปที่ 2.26 และ 2.27



รูปที่ 2.26 ประสิทธิภาพของ YOLOv5



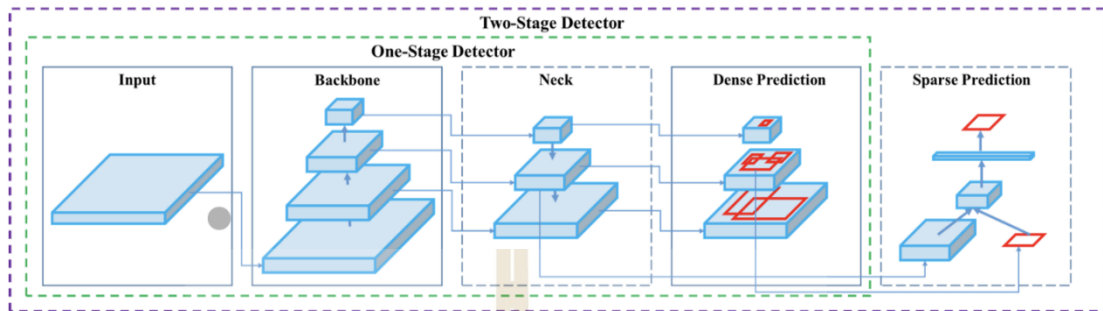
รูปที่ 2.27 Algorithm ย่อยของ YOLOv5

### 2.9.1 หลักการทำงานของ YOLOv5

สำหรับหลักการทำงานของ YOLOv5 นั้น เริ่มจากมีข้อมูล input เข้ามาโมเดล YOLOv5 ก็จะทำการแบ่ง grid cell ออกเป็นช่องเล็ก ๆ  $n \times n$  grid โดยที่สถาปัตยกรรมของ YOLOv5 นั้นจะประกอบไปด้วย 3 ส่วนประกอบหลักดังแสดงในรูปที่ 2.28

1. Backbone คือ ส่วนที่รับข้อมูล input เข้ามาแล้วทำการสกัดเอาคุณลักษณะเด่นของรูปภาพนั้นออกมาเพื่อที่จะส่งต่อไปยังส่วนถัดไปของโมเดล
2. Neck คือส่วนที่รับข้อมูล input มาจาก Backbone แล้วนำคุณลักษณะเด่นที่ถูกสกัดมาสร้าง feature map ต่อไป

3. Head ในส่วนสุดท้ายนี้จะทำหน้าที่ตีกรอบไปยังตำแหน่งของวัตถุที่เราสนใจ แล้วทำการทำนายว่าวัตถุนั้นอยู่ใน class ใดหลังจากนั้นจะส่งไปยัง output layer สุดท้ายต่อไป



รูปที่ 2.28 โครงสร้างสถาปัตยกรรมของ YOLOv5 Algorithm

ตัวอย่างภาพการตรวจจับวัตถุโดยใช้ YOLOv5 Algorithm ดังรูปที่ 2.29



รูปที่ 2.29 ตัวอย่างการตรวจจับวัตถุด้วย YOLOv5

## 2.10 ปรัชศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

### 2.10.1 งานวิจัยเกี่ยวกับโครงข่ายประสาทเทียม

ปี ค.ศ. 2019 Wang และคณะ ได้ทำการจำแนกและระบุประเภทของภาพเบลอสี่ประเภท โดยการศึกษาและออกแบบตัวดำแนก SFA+SFGN สำหรับการทดสอบประสิทธิภาพของตัวจำแนก ensemble โดยการทดสอบและเปรียบเทียบกับ GoogleNet, Alexnet และการจำแนกภาพเบลอวิธีอื่น ๆ ด้วยชุดข้อมูลภาพเบลอที่สร้างขึ้นเองและภาพเบลอที่เกิดขึ้นเองใน BHFID จากการทดลองเผยให้เห็นถึงประสิทธิภาพที่ดีกว่าตัวของจำแนกแบบ ensemble เมื่อเปรียบเทียบกับ

GoogleNet และ Alexnet ประสิทธิภาพดีพอ ๆ กันแต่ใช้เวลาเฉลี่ยในการทำนายภาพ 0.159s ดังนั้น ตัวจำแนกประเภท ensemble ให้ประสิทธิภาพที่ดีกว่าช่วยให้การใช้งานโครงข่ายประสาทเทียมแบบซับซ้อนได้สะดวกกว่า

ปี ค.ศ. 2019 Dung และทีมงาน มุ่งงานวิจัยที่ต้องศึกษาเกี่ยวกับการตรวจสอบรอยแตกของโครงสร้างคอนกรีตแบบอัตโนมัติ ซึ่งเป็นงานที่สำคัญมากสำหรับวิศวกรรมโยธา วิธีที่ใช้สำหรับตรวจสอบคือโครงข่ายประสาทเทียมแบบ Fully Convolution Neural Network (FCN) ในการประเมินและจัดประเภทของภาพบนชุดข้อมูลคอนกรีตที่มีรอยแตกซึ่งมาขนาดภาพ 227 x 227 พิกเซล โดยจะแบ่งข้อมูลที่เป็นภาพร้าวของคอนกรีตออกมาฝึกสอนให้ได้ประสิทธิภาพในการทำงานของโครงข่ายประสาทเทียมที่ดี และโครงข่ายประสาทเทียม FCN แบบ encoder-decoder ที่ใช้ VGG16 ถูกสอนทั้งหมด 500 ครั้ง โดยที่ความแม่นยำของโครงข่ายประสาทเทียม FCN มีความแม่นยำเฉลี่ยอยู่ที่ 90% โดยใช้ภาพรอยร้าวของคอนกรีตที่ได้จากวิดีโอพบว่ามีการตรวจพบรอยแตกที่สมเหตุสมผล และความหนาแน่นของรอยแตกก็ถูกประเมินอย่างแม่นยำเช่นกันด้วยโครงข่าย FCN encoder-decoder ที่ใช้ VGG16

ปี ค.ศ. 2019 Xingyuan. Bu และทีมงาน ได้เสนอให้มีการศึกษาเกี่ยวกับชั้นของ convolutional ว่ามีประโยชน์และความเหมาะสมในการบ่งชี้ที่มีความสามารถในการจำแนกพื้นผิวภายใต้กรอบการเรียนรู้แบบ end-to-end โดยจะใช้เทคนิคการฝึกสอนโมเดลแบบ back-propagation network เพื่อให้ง่ายต่อการแก้ไขชั้นของ convolutional ในแต่ละชั้นได้ จึงทำให้การเรียนรู้ของแต่ละชั้นนั้นมีความฉลาดมากยิ่งขึ้น โดยจะแสดงผลลัพธ์ที่มีความแม่นยำมากยิ่งขึ้นสำหรับการจำแนกพื้นผิว

### 2.10.2 งานวิจัยที่เกี่ยวข้องกับการตรวจจับวัตถุภายในภาพ

ในปี ค.ศ 2016 Joseph R. และคณะ ได้นำเสนออัลกอริทึมใหม่สำหรับการตรวจจับวัตถุ (Object detection) โดยให้ชื่อว่า “You Only Look Once : YOLO” โดยมีความแตกต่างจากอัลกอริทึมอื่น ๆ ในยุคนั้น คือ แทนที่จะใช้ Neural network 2 ชุดคือ การหา Feature map และการทำ Classification สำหรับแก้ปัญหาที่นั้น ๆ แต่สำหรับ YOLO นั้น จะเป็นแบบ Unified architecture neural network เพียงชุดเดียวสำหรับการแก้ปัญหาการตรวจจับวัตถุซึ่งสามารถทำงานเร็วขึ้นมากเมื่อเปรียบเทียบกับ R-CNN และ Fast R-CNN

ในปี ค.ศ. 2018 Joseph R. และ Ali F. ได้พัฒนา YOLO ขึ้นมาเป็นเวอร์ชันที่ 3 (YOLOv3) ซึ่งมีการปรับการทำนาย class ในส่วนของ Bounding box จากเดิมใช้ Linear regression เปลี่ยนมาเป็น Logistic regression และได้ทำการเอา Softmax layer ออกเพื่อให้เกิด bounding box overlap ได้และได้เพิ่มส่วนของ Multi-scale detection เพื่อทำให้ภาพที่มีขนาดต่างกันมีความแม่นยำในการตรวจจับวัตถุ

ในปี ค.ศ. 2020 Alexey Bochkovskiy, Chien-Yao Wang และ Hong-Yuan Mark Liao ได้นำ YOLOv3 ของ Joseph R. และ Ali F. ซึ่งถอนตัวออกจากการพัฒนา object detection มาพัฒนาต่อโดยได้เผยแพร่ YOLOv4 โดยทำการพัฒนา YOLOv3 ด้วยการประยุกต์เทคนิคต่างๆ เข้าไปเพิ่มเติม เช่น Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Selfadversarial-training (SAT) และทำการปรับเปลี่ยน activation function จากเดิมใช้ leaky relu เปลี่ยนมาเป็น mish จึงทำให้ประสิทธิภาพความแม่นยำของ YOLOv4 นั้นสูงขึ้นจาก YOLOv3

และในไม่กี่เดือนต่อมา บริษัท Ultralytics ได้พัฒนา YOLO จากเวอร์ชัน 4 มาเป็น YOLOv5 ขึ้นมาทำให้การใช้งาน YOLO นั้นง่ายขึ้นและปรับปรุงประสิทธิภาพให้ดียิ่งขึ้นและได้แบ่ง model algorithm สำหรับสร้าง object detection model ให้เหมาะสมกับการใช้งานได้แก่ model s, model m, model l และ model x ตามลำดับซึ่งหมายถึงขนาดของ model

ระบบการตรวจจับองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ในปัจจุบันคือระบบ computer vision ซึ่งมีการใช้งานอย่างแพร่หลายในอุตสาหกรรมต่าง ๆ และสามารถทำงานได้เป็นอย่างดีแต่ ณ ปัจจุบันระบบการตรวจจับวัตถุที่กำลังนิยมและเป็นเทคโนโลยีใหม่คือการตรวจจับองค์ประกอบภายในภาพด้วยเทคนิคการเรียนรู้เชิงลึกซึ่งมีความแตกต่างจาก ระบบ computer vision ดังตารางที่ 2.1 และการเลือกใช้งานของทั้ง 2 เทคนิคแสดงในตารางที่ 2.2

ตารางที่ 2.1 ตารางแสดงความแตกต่างระหว่าง computer vision กับ deep learning

| Differences                                  | Deep learning | Traditional computer vision |
|--|---------------|-----------------------------|
| Manual feature extraction required?          | ✗             | ✓                           |
| Training computationally resource-heavy?     | ✓             | ✗                           |
| Requires labeled or classification datasets? | ✓             | ✗                           |
| Black-box models?                            | ✓             | ✗                           |
| Easy to deploy even on microprocessors?      | ✗             | ✓                           |
| Yield high accuracy rates                    | ✓             | ✗                           |



ตารางที่ 2.2 ตารางแสดงเกณฑ์การคัดเลือกระหว่าง computer vision กับ deep learning

| Selection criteria     | Deep learning | Traditional computer vision |
|------------------------|---------------|-----------------------------|
| Training dataset       | Large         | Small                       |
| Computing power        | High          | Low                         |
| Training time          | Long          | Short                       |
| Annotation time        | Long          | Short                       |
| Feature engineering    | Unnecessary   | Required                    |
| Deployment flexibility | Low           | High                        |
| Expenditure (BOM)      | High          | Low                         |

### 2.11 สรุป

จากการศึกษาทฤษฎีความรู้และปรัทัศน์วรรณกรรมที่เกี่ยวข้องกับงานวิจัย เพื่อที่จะพัฒนาระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ให้มีประสิทธิภาพและความถูกต้องแม่นยำให้มากยิ่งขึ้น โดยจะเห็นจากงานวิจัยอื่น ๆ เกี่ยวกับระบบตรวจจับวัตถุภายในภาพนั้น มีการใช้งานอย่างแพร่หลายในปัจจุบัน ดังนั้น การพัฒนาระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์นั้นมีความสำคัญเป็นอย่างมากและทำงานได้อย่างรวดเร็วโดยทางผู้วิจัยได้เสนอให้ใช้เครื่องมือในการตรวจสอบคือการเรียนรู้เชิงลึก (deep learning) โดยจะใช้อัลกอริทึมสองอย่างร่วมกันคือ โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution neural network) สำหรับการตรวจสอบเบื้องต้นโดยดูเฉลี่ยทั้งภาพของฮาร์ดดิสก์ไดรฟ์ (HDE view) และอีกหนึ่งอัลกอริทึมที่สำคัญสำหรับ VI system ก็คือ YOLOv5 algorithm ใช้สำหรับระบุ class องค์ประกอบต่าง ๆ ภายในภาพว่าถูกต้องหรือไม่โดยสุดท้ายแล้วการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์นั้นจะจำแนกว่าภาพที่ถ่ายมานั้นผ่าน (pass) หรือไม่ผ่าน (fail) มีชิ้นส่วนของฮาร์ดดิสก์ไดรฟ์ตำแหน่งไหนผิดปกติหรือไม่ สุดท้ายก็เพื่อเพิ่มประสิทธิภาพการผลิตฮาร์ดดิสก์ไดรฟ์ให้ได้จำนวนตามที่ตั้งเป้าไว้

## บทที่ 3

### วิธีการดำเนินงานวิจัย

#### 3.1 บทนำ

งานวิจัยนี้มีเป้าหมายในการออกแบบ วิจัยและพัฒนาาระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ในมุมมองด้านบน (Visual Inspection system HDE view) ในกระบวนการประกอบฮาร์ดดิสก์ไดรฟ์ของไลน์การผลิตแบบอัตโนมัติโดยจะใช้เทคนิคการเรียนรู้เชิงลึก (deep learning technique) ในการพัฒนาระบบ VI system ตัวต้นแบบเพื่อต่อยอดและนำไปใช้งานแทนระบบตรวจสอบเดิมที่ใช้การมองเห็นของเครื่องจักร (Machine vision) ในการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์โดยทำการกำหนดขอบเขตการมองเห็น (Field of view) ของแต่ละตำแหน่งชิ้นงาน และตรวจสอบว่าชิ้นงานอยู่ในขอบเขตการมองเห็นหรือไม่จึงเกิดปัญหาที่ระบบมองว่าชิ้นงานไม่ได้อยู่ใน FOV เกิดปัญหา Overkill ขึ้นคือระบบมองว่าชิ้นงานอยู่ในตำแหน่งที่ไม่ถูกต้องทั้ง ๆ ที่ความจริงแล้วฮาร์ดดิสก์ไดรฟ์ไม่ได้เกิดข้อผิดพลาดในการประกอบชิ้นงานเรียกว่า Overkill

สำหรับการออกแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึกนั้นจะแบ่งออกเป็นสองส่วนคือ 1. โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution neural network) เป็นคั้งแรกของระบบใช้สำหรับตรวจสอบภาพรวมของภาพ HDE view ว่ามีความถูกต้องหรือไม่โดยจะเน้นไปที่การมองภาพรวมทั้งหมดก่อนและทำการคัดแยกผ่านหรือไม่ผ่าน (pass or fail) หลังจากนั้น หากภาพที่ผ่านจุดคัดกรองแรกไปได้จะถูกส่งข้อมูลไปยังตำแหน่งตรวจสอบที่สองคือ object detection (YOLOv5 algorithm) ทำหน้าที่ในการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ทั้งหมด 16 ตำแหน่งว่ามีความถูกต้องหรือไม่โดยจะแบ่งเป็นช่องตารางในการตรวจสอบแล้วแบ่งเป็น class ตามที่ได้ระบุไว้หลังจากนั้นก็ทำการคัดแยกผ่านหรือไม่ผ่าน อีกครั้งหากผ่านก็ถือว่าฮาร์ดดิสก์ไดรฟ์ นั้น Pass หากไม่ผ่านก็จะแจ้งว่า Fail

จากที่กล่าวมาข้างต้นระบบเดิมขาดความยืดหยุ่นในการตรวจสอบ ดังนั้น ผู้วิจัยจึงเสนอแนวทางการพัฒนาระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ให้มีประสิทธิภาพมากขึ้นและยืดหยุ่นสามารถฝึกสอน (training) และฝึกสอนโมเดลใหม่ (Re-training) ทำให้การตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์มีความแม่นยำมากยิ่งขึ้น สำหรับงานวิจัยนี้จะมุ่งเน้นที่มุมมองด้านบน (HDE view) ของเครื่องจักร VI เท่านั้น โดยจะมีการตรวจสอบองค์ประกอบภายในภาพทั้งหมด 17 ตำแหน่งดังที่แสดงในรูปที่ 3.1

| Camera | Parameter | EC           | RS | G-code | Note |                          |
|--------|-----------|--------------|----|--------|------|--------------------------|
| HDE    | 1         | BASE1/BASE2  | 3  | RS03   | 3279 | Extra parts in MBA       |
|        | 2         | EXTRA        | 3  | RS03   | 3279 | Extra parts in MBA       |
|        | 3         | ExPartDisk   | 20 | RS20   | 3292 | Extra parts on media     |
|        | 4         | C-Filter     | 6  | RS06   | 3281 | C-Filter missing         |
|        | 5         | CLIP         | 7  | RS07   | 3282 | Comb not remove          |
|        | 6         | FPC          | 10 | RS10   | 3284 | FCC or screw problem     |
|        | 7         | FPCSCR1      | 10 | RS10   | 3284 | FCC or screw problem     |
|        | 8         | FPCSCR2      | 10 | RS10   | 3284 | FCC or screw problem     |
|        | 9         | HSA          | 11 | RS11   | 3285 | HSA damage or no HSA     |
|        | 10        | OUTERCS      | 14 | RS14   | 3287 | ODCS damage or missing   |
|        | 11        | Pivot        | 15 | RS15   | 3288 | Pivot abnormal or damage |
|        | 12        | RAMP         | 16 | RS16   | 3437 | Ramp missing or damage   |
|        | 13        | SPOILER      | 12 | RS12   | 3431 | Spoiler or screw missing |
|        | 14        | SpoilerScrew | 12 | RS12   | 3431 | Spoiler or screw missing |
|        | 15        | TC           | 17 | RS17   | 3290 | Top clamp missing        |
|        | 16        | TC_Center    | 17 | RS17   | 3290 | Top clamp missing        |
|        | 17        | VCM          | 18 | RS18   | 3291 | TVCM missing             |

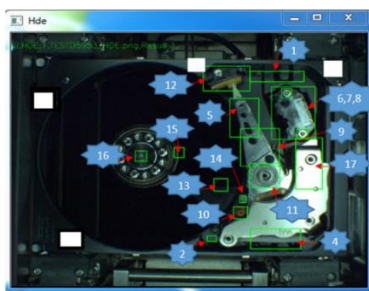


Figure 10. Pattern and Find Area for Ex-PartDisk

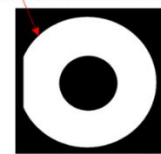


Figure 11. Mask Image of Ex-PartDisk

### รูปที่ 3.1 องค์ประกอบของฮาร์ดดิสก์ไดรฟ์มุมมองด้านบน (HDE view)

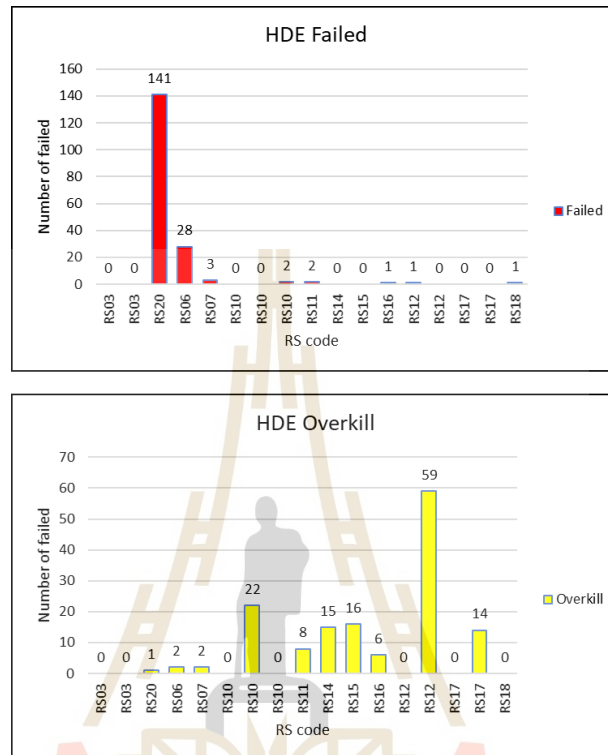
## 3.2 วิเคราะห์ข้อมูลและการเก็บรวบรวมข้อมูลภาพ HDE view

สำหรับแผนการดำเนินงานวิจัยขั้นตอนแรกคือการเก็บรวบรวมข้อมูลภาพ HDE view และทำการตรวจสอบว่าภาพที่ได้มานั้นมีที่ประเภทและตรวจสอบการทำงานของระบบเดิมจากชุดข้อมูลแรกที่ได้รับมานั้นจะมีภาพ HDE view ทั้งหมด 739 ภาพโดยจะแบ่งภาพ HDE view ออกเป็นสามประเภทได้แก่ ภาพดี (Good images) = 418 ภาพ, ภาพเสีย (Fail images) = 177 ภาพ และภาพดีแต่ระบบมองว่าเสีย (Overkill images) = 144 ภาพ ซึ่งทางผู้วิจัยได้ทำการแบ่งข้อมูลออกเป็นประเภทตามประเภทที่ใช้ในการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ ดังที่แสดงในตารางที่ 3.1 แบ่งออกเป็น 17 ประเภทและตรวจสอบภาพ HDE view เฉพาะ Fail images และ Overkill images จากตารางจะเห็นได้ว่าภาพ Overkill นั้นเกิดขึ้นจากตำแหน่งของ Spoiler Screw มากที่สุดทั้งหมด 59 ภาพลำดับต่อมาคือตำแหน่งของน๊อตบนแผง FPC (FPCSCR1) ลำดับต่อมาคือ Pivot และ OUTERCS ตามลำดับ

ตารางที่ 3.1 ตารางแสดงการตรวจสอบตำแหน่งผิดปกติของฮาร์ดดิสก์ไดรฟ์ Fail และ Overkill

| No. | Parameter    | EC | RS code | G-code | Note                     | Failed | Overkill |
|-----|--------------|----|---------|--------|--------------------------|--------|----------|
| 1   | BASE1/BASE2  | 3  | RS03    | 3279   | Extra parts in MBA       | 0      | 0        |
| 2   | EXTRA        | 3  | RS03    | 3279   | Extra parts in MBA       | 0      | 0        |
| 3   | ExPartDisk   | 20 | RS20    | 3292   | Extra parts on media     | 141    | 1        |
| 4   | C-Filter     | 6  | RS06    | 3281   | C-Filter missing         | 28     | 2        |
| 5   | CLIP         | 7  | RS07    | 3282   | Comb not remove          | 3      | 2        |
| 6   | FPC          | 10 | RS10    | 3284   | FCC or screw problem     | 0      | 0        |
| 7   | FPCSCR1      | 10 | RS10    | 3284   | FCC or screw problem     | 0      | 22       |
| 8   | FPCSCR2      | 10 | RS10    | 3284   | FCC or screw problem     | 2      | 0        |
| 9   | HSA          | 11 | RS11    | 3285   | HSA damage or no HSA     | 2      | 8        |
| 10  | OUTERCS      | 14 | RS14    | 3287   | ODCS damage or missing   | 0      | 15       |
| 11  | Pivot        | 15 | RS15    | 3288   | Pivot abnormal or damage | 0      | 16       |
| 12  | RAMP         | 16 | RS16    | 3437   | Ramp missing or damage   | 1      | 6        |
| 13  | SPOILER      | 12 | RS12    | 3431   | Spoiler or screw missing | 1      | 0        |
| 14  | SpoilerScrew | 12 | RS12    | 3431   | Spoiler or screw missing | 0      | 59       |
| 15  | TC           | 17 | RS17    | 3290   | Top clamp missing        | 0      | 0        |
| 16  | TC_Center    | 17 | RS17    | 3290   | Top clamp missing        | 0      | 14       |
| 17  | VCM          | 18 | RS18    | 3291   | TVCM missing             | 1      | 0        |

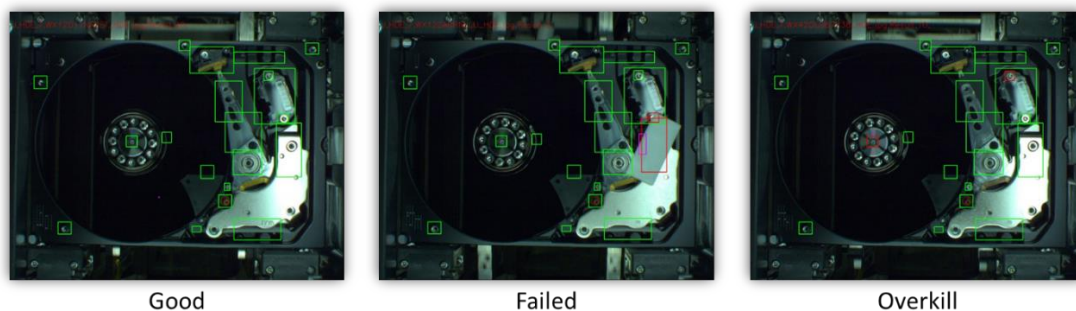
จากตารางข้างต้นจะสามารถสรุปความผิดปกติที่เกิดขึ้นกับภาพ HDE view ดังกราฟที่แสดง  
 ดังรูปที่ 3.2



รูปที่ 3.2 จำนวนภาพที่เกิดความผิดปกติจริงเทียบกับไม่จริง (Overkill) ของ HDE view

### 3.2.1 ประเภทของภาพ HDE view

จากภาพชุดแรกของ HDE view นั้น สามารถแบ่งประเภทของออกเป็น 3 ประเภท ได้แก่ Good image, Fail image, และ Overkill image ดังที่แสดงในรูปที่ 3.3 จากภาพ Overkill จะสังเกตเห็นว่าภาพ HDE view นั้น ไม่ได้ผิดปกติแต่อย่างใดแต่ระบบกลับมองว่าตำแหน่งขององค์ประกอบฮาร์ดดิสก์ไดรฟ์ไม่ถูกต้องจาก FOV สีแดงที่เกิดขึ้น



รูปที่ 3.3 ประเภทของภาพ HDE view

### 3.2.2 ข้อมูลของภาพ HDE view

ข้อมูลภาพของ HDE view ได้แก่

- ภาพมีขนาด 1280x1024 pixel
- ภาพเป็นไฟล์สกุล JPG
- เป็นภาพ Top view ของ HDD

### 3.3 วิธีการทดลอง

เริ่มจากการศึกษาและเก็บข้อมูลภาพ HDE view เพื่อตรวจสอบความผิดปกติที่เกิดขึ้น หลังจากนั้นก็ทำการออกแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึกโดยจะใช้เครื่องมือทางการเรียนรู้ของเครื่องจักรอยู่สองชนิดร่วมกันคือ โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution neural network) และการตรวจจับวัตถุภายในภาพด้วยอัลกอริทึม โยโลเวอร์ชันห้า (Object detection by YOLOv5 algorithm) เมื่อทำการออกแบบระบบเรียบร้อยแล้วก็เริ่มจากการเก็บข้อมูลภาพ HDE view เพื่อมาทำความสะอาดข้อมูล (cleaning data) และทำการตรวจสอบความถูกต้องของข้อมูลจัดประเภทให้พร้อมสำหรับการใช้งาน จากนั้น จะทำการส่งข้อมูลไปยังขั้นตอนถัดไป (Data selection) เพื่อให้เหมาะสมกับการใช้งานกับโมเดลของระบบไม่ว่าจะเป็น CNN และ YOLOv5 โดยการเตรียมข้อมูลสำหรับ CNN นั้นจะสร้างไว้สองประเภทคือ Pass images กับ Fail images สำหรับการฝึกสอนโมเดลให้โมเดล CNN แต่สำหรับโมเดล YOLOv5 นั้นจะเตรียม Pass images เท่านั้น แต่จะต้องทำการสร้างขอบเขตของวัตถุเป้าหมายให้กับแต่ละภาพด้วย เพื่อเป็นการบอกโมเดลว่าในกรอบที่สร้างขึ้นนั้นคือวัตถุอะไรเป็น class ไหนสำหรับฝึกสอนโมเดล YOLOv5 จากนั้นเมื่อทำการฝึกสอนโมเดลทั้งสองเสร็จเรียบร้อยแล้วก็จะทำการทดสอบ VI system ว่ามีประสิทธิภาพในการทำนายภาพ HDE view ได้ถูกต้องหรือไม่โดยผู้วิจัยคาดหวังไว้ว่าจะมีความแม่นยำในการทำนายมากกว่าหรือเท่ากับ 98% หากเปอร์เซ็นต์ความถูกต้องของระบบยังไม่ถึง

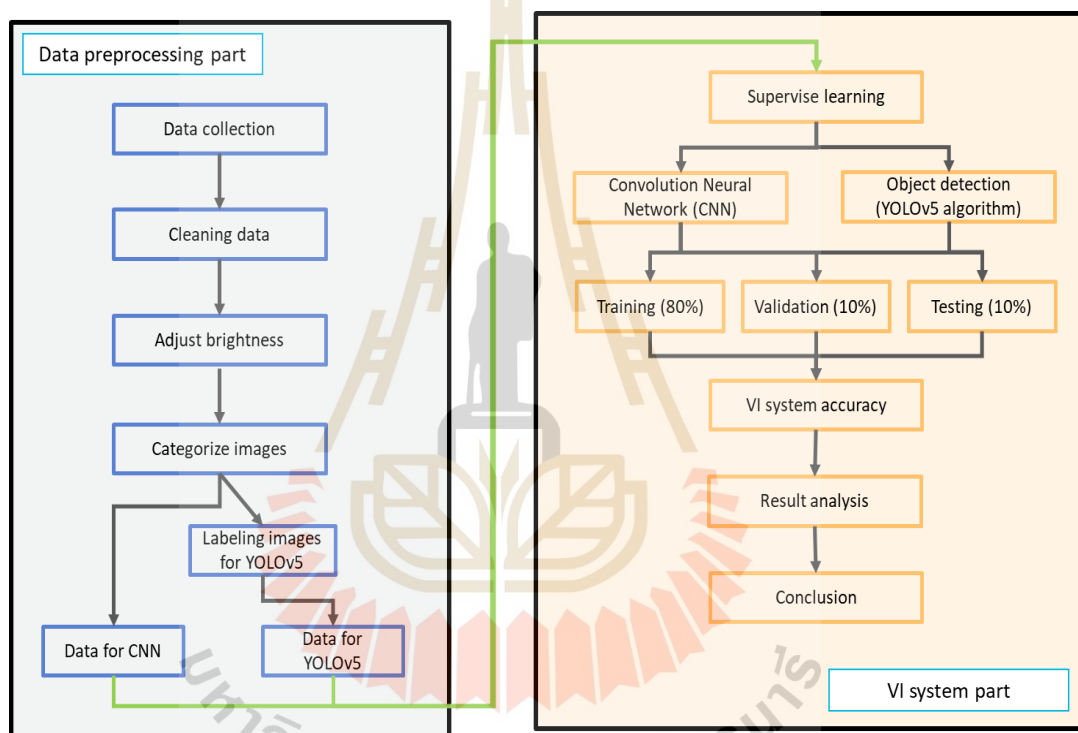
เป้าหมายที่คาดหวังไว้ก็จะต้องทำการฝึกสอนโมเดลใหม่จนกว่าจะได้ระบบที่มีความสามารถในการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ประมาณ 98% ขึ้นไปโดยแผนผังการดำเนินการทดลองแสดงดังรูปที่ 3.4 โดยที่คอมพิวเตอร์สำหรับการฝึกสอนและทดสอบโมเดลมีสเปคดังนี้

Computer spec

CPU : I9 gen11

Ram : 32GB DDR4

GPU : Nvidia RTX 3060Ti



รูปที่ 3.4 แผนผังการดำเนินการทดลอง VI system

### 3.3.1 การเก็บข้อมูล (Data collection)

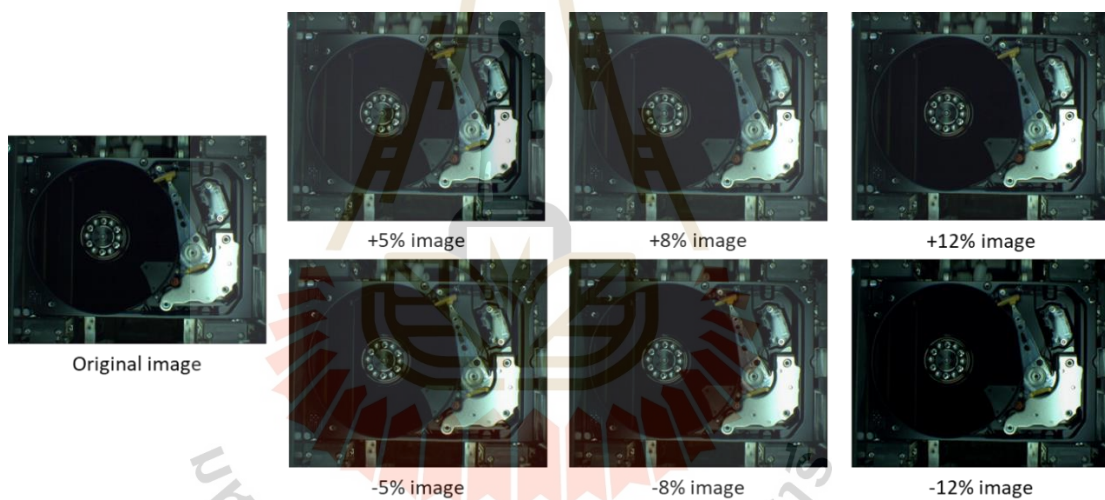
เก็บภาพ HDE view จากเครื่อง Visual inspection machine ของ NEO-Automation line (Venice product) ซึ่งสามารถเก็บรูป Pass มาทั้งหมด 2500 ภาพ และรูป Fail 637 ภาพ โดยจะแบ่งไว้สำหรับฝึกสอนโมเดล Pass = 1000 ภาพ และ Fail = 177 ภาพ ภาพที่เหลือจะใช้สำหรับสร้างชุดทดสอบ VI system โดยจะแบ่งออกเป็น 4 ชุดข้อมูล

### 3.3.2 การทำความสะอาดข้อมูล (Cleaning data)

เป็นขั้นตอนการตรวจสอบความถูกต้องของภาพ HDE view ว่ามีความถูกต้องหรือไม่ไฟล์ภาพ Pass ต้องไม่มีภาพที่ Fail ปนอยู่และสำหรับไฟล์ภาพ Fail ก็ต้องไม่มีภาพ Pass ปนอยู่เป็นการเตรียมข้อมูลให้มีความถูกต้องก่อนที่จะนำเข้าสู่กระบวนการถัดไป

### 3.3.3 เทคนิคการประมวลผลภาพเพิ่มลดความสว่างของภาพ (Image processing - adjust brightness)

รูปที่ 3.5 แสดงการปรับเพิ่มค่าความสว่างของภาพ HDE view โดยจะนำรูปที่ทำความสะอาดแล้วมาปรับความสว่างเพื่อเพิ่มจำนวนภาพให้ครอบคลุมช่วงค่าความสว่างที่กำหนดไว้ของเครื่อง Visual inspection machine โดยค่าความสว่างที่ยอมรับได้คือ  $\pm 10\%$  ดังนั้น ในขั้นตอนนี้จะทำการปรับค่าความสว่างของภาพ HDE view ดังนี้  $\pm 5\%$ ,  $\pm 8\%$  และ  $\pm 12\%$  สำหรับภาพที่จะนำไปฝึกสอนโมเดล



รูปที่ 3.5 การปรับความสว่างของภาพ HDE view

### 3.3.4 จัดประเภทของรูปภาพ (Categorize images)

หลังจากได้ทำการปรับค่าความสว่างของภาพแล้วจะทำการสร้าง folder สำหรับจัดเก็บภาพ HDE view เพื่อใช้งานด้านต่าง ๆ ดังนี้

#### 3.3.4.1 ข้อมูลสำหรับฝึกสอนโมเดล (Data for model training)

สำหรับการฝึกสอนโมเดลจะแบ่งออกเป็น 2 ส่วนคือข้อมูลสำหรับ CNN model และ ข้อมูลสำหรับ YOLOv5 model ดังนี้

- Data for CNN model

ภาพต้นฉบับของ Pass images = 1000 ภาพเมื่อปรับค่าความสว่างแล้วจะมีภาพทั้งหมด 7000 ภาพ

ภาพต้นฉบับของ Fail images = 177 ภาพเมื่อปรับค่าความสว่างแล้วจะมีภาพทั้งหมด 1239 ภาพ

โดยจะแบ่งภาพออกเป็น 3 ส่วนสำหรับการสร้างโมเดล CNN คือ

Pass images

Training (80%) = 5600 images

Validation (10%) = 700 images

Testing (10%) = 700 images

Fail images

Training (80%) = 991 images

Validation (10%) = 124 images

Testing (10%) = 124 images

- Data for YOLOv5 model

สำหรับภาพที่ใช้ในการสร้างโมเดลตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์นั้น จะใช้แค่ Pass images เท่านั้นแต่จะมีการเพิ่มขึ้นขั้นตอนการทำ Labeling เข้ามาเพื่อสร้างขอบเขตให้กับตำแหน่งต่าง ๆ ที่ต้องการตรวจสอบว่ามีความผิดปกติหรือไม่

ภาพต้นฉบับของ Pass images = 418 ภาพเมื่อปรับค่าความสว่างแล้วจะมีภาพทั้งหมด 2926 ภาพ

โดยจะแบ่งภาพออกเป็น 3 ส่วนสำหรับการสร้างโมเดล YOLOv5 คือ

Training (80%) = 2340 images

Validation (10%) = 293 images

Testing (10%) = 293 images

- Data for testing VI system

สำหรับภาพที่ใช้ในการทดสอบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึกนั้นได้ทำการแบ่งชุดทดสอบออกเป็น 4 ชุดโดยเป็นข้อมูลที่โมเดลไม่เคยเรียนรู้มาก่อนโดยจะแบ่งออกเป็น Pass images = 375 ภาพและ Fail images = 115 ภาพดังรูปที่ 3.6

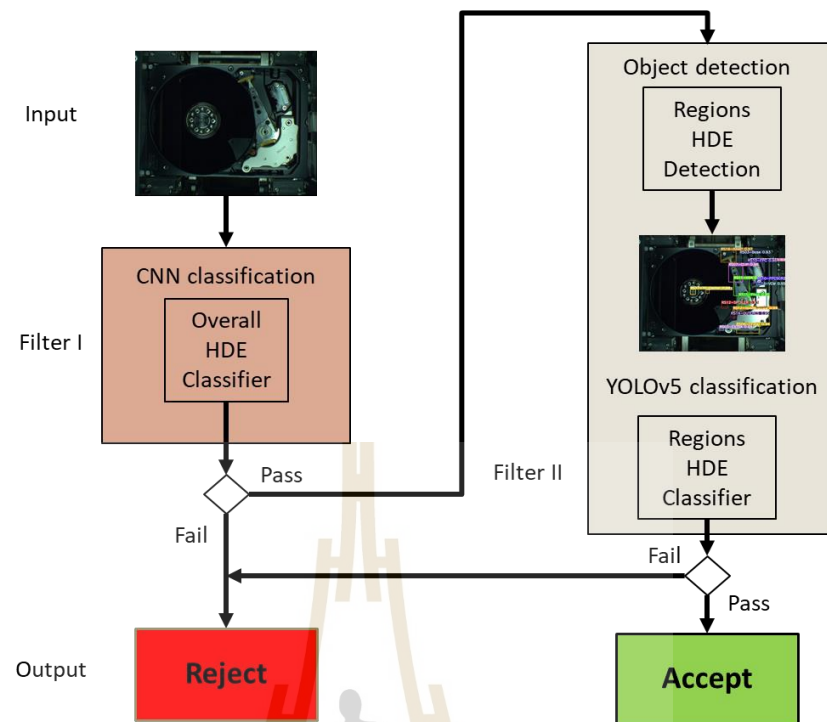


|        |                    |             |
|--------|--------------------|-------------|
| Set1   | 11/22/2021 6:37 AM | File folder |
| Set2   | 11/22/2021 6:37 AM | File folder |
| Set3   | 11/22/2021 6:37 AM | File folder |
| Set4   | 11/22/2021 6:37 AM | File folder |
| Fail   | 11/22/2021 6:24 AM | File folder |
| Pass   | 11/22/2021 6:26 AM | File folder |
| Result | 11/22/2021 6:37 AM | File folder |

รูปที่ 3.6 Folder ชุดทดสอบระบบ VI system

### 3.3.5 การออกแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไทรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึก

แนวคิดในการออกแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไทรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึกหรือ deep learning นั้น จะเริ่มต้นด้วยการนำภาพเข้าสู่ระบบด้วยไฟล์ต้นฉบับ (HDE view) จากนั้น ระบบจะใช้โมเดลโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution neural network) ในการตรวจสอบภาพรวมของภาพ HDE view ว่าถูกต้องหรือไม่ โดยจะทำการฝึกสอนให้แยกเป็น 2 class คือ Pass กับ Fail เปรียบเสมือนจุดตรวจสอบที่หนึ่ง หลังจากนั้นภาพที่ผ่านจุดตรวจสอบ CNN จะถูกส่งต่อไปยังจุดตรวจสอบที่สอง คือการตรวจเช็คองค์ประกอบต่าง ๆ ของฮาร์ดดิสก์ไทรฟ์ 16 ตำแหน่งเรียกจุดตรวจสอบที่สองว่า Object detection โดยการใช้ YOLOv5 algorithm ในการตรวจสอบหากระบบพบว่ามีตำแหน่งไหนที่ไม่ตรงตามภาพที่โมเดลได้ทำการเรียนรู้ไว้ก็จะกลายเป็น Fail สถาปัตยกรรมของระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไทรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึกแสดงดังรูปที่ 3.7



รูปที่ 3.7 สถาปัตยกรรมของระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึก

### 3.3.6 สร้างโมเดล CNN (Create CNN model)

ขั้นตอนต่อมาคือการสร้างโมเดล CNN โดยจะทำการทดลองปรับเปลี่ยนตั้งค่าในการสร้างโมเดล CNN แบบต่าง ๆ เพื่อหาโมเดล CNN ที่เหมาะสมที่สุดสำหรับระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (HDE view) สำหรับการแบ่งข้อมูลเพื่อสร้างโมเดล CNN ดังรูปที่ 3.8 และการตั้งค่าสำหรับการสร้างโมเดลดังนี้

CNN model training

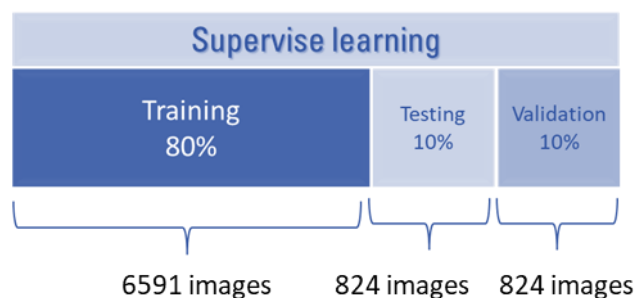
Image size = resize from 1280x1024 to 512x512 pixel

Training (80%) = 6591 images

Validation (10%) = 824 images

Testing (10%) = 824 images

Activation function = ReLU function



รูปที่ 3.8 การแบ่งข้อมูลสำหรับสร้างโมเดล CNN

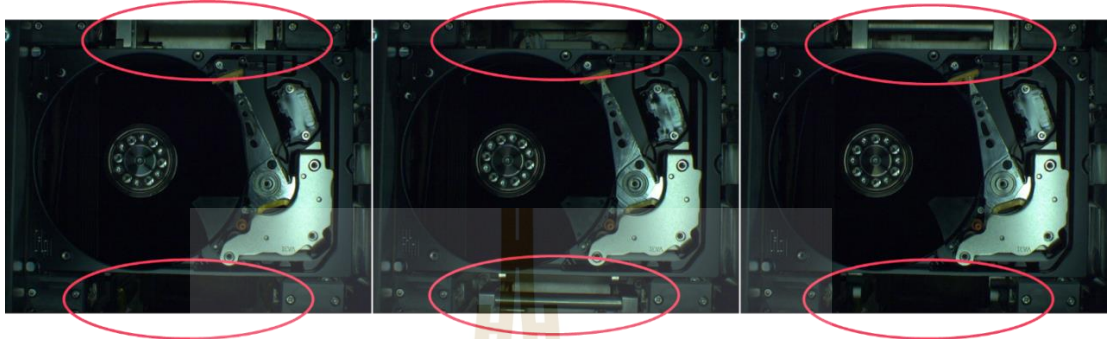
สำหรับการหาโมเดลที่มีประสิทธิภาพความแม่นยำสูงที่สุดนั้นได้ทำการปรับตัวแปรที่มีผลต่อเปอร์เซ็นต์ความแม่นยำให้ได้ผลลัพธ์ที่ดีที่สุดออกมาโดยที่จะเหลือการปรับเปลี่ยนตัวแปรเหลือแค่จำนวนรอบการฝึกสอน (epoch) เท่านั้น ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 ตารางเปรียบเทียบโมเดล CNN จากการฝึกสอนโมเดล

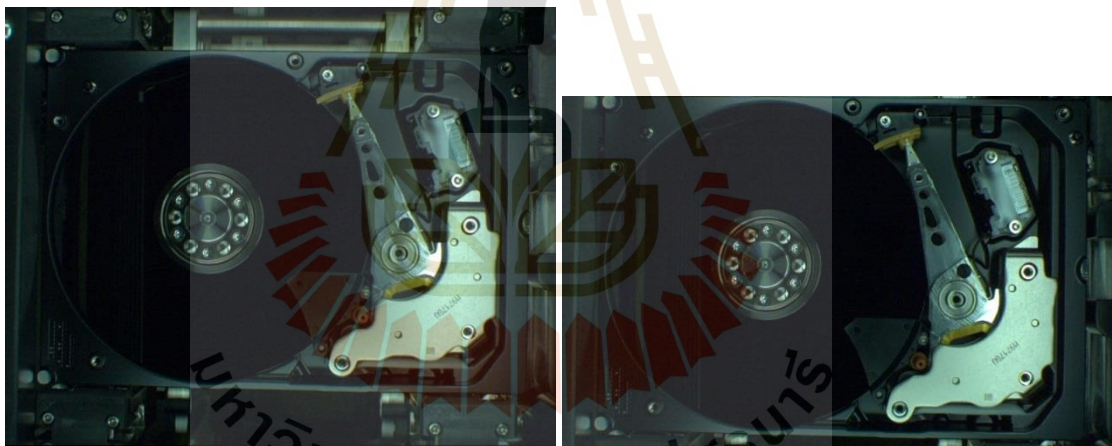
| CNN model training configuration |                    |       |            |              |                    |                 |           |
|----------------------------------|--------------------|-------|------------|--------------|--------------------|-----------------|-----------|
| No. model                        | Image size (pixel) | Epoch | Batch size | Conv. Layers | Max pooling layers | Activation Fcn. | %Accuracy |
| 1                                | 512x512            | 10    | 16         | 64           | 2x2                | ReLU            | 58.74%    |
| 2                                | 512x512            | 20    | 16         | 64           | 2x2                | ReLU            | 76.79%    |
| 3                                | 512x512            | 40    | 16         | 64           | 2x2                | ReLU            | 61.54%    |
| 4                                | 512x512            | 60    | 16         | 64           | 2x2                | ReLU            | 57.44%    |
| 5                                | 512x512            | 80    | 16         | 64           | 2x2                | ReLU            | 51.34%    |
| 6                                | 512x512            | 100   | 16         | 64           | 2x2                | ReLU            | 48.63%    |

จากตารางจะเห็นได้ว่าเปอร์เซ็นต์ความแม่นยำในการทำนายของ CNN นั้นยังไม่ถึงเป้าหมายที่คาดหวังไว้ถึงจะมีเปอร์เซ็นต์ความถูกต้องสูงที่สุดอยู่ที่ 76.79% สังเกตได้ว่าหากจำนวนรอบการฝึกสอนน้อยจนเกินไปเปอร์เซ็นต์ความถูกต้องก็ไม่ได้สูงมากเรียกว่า Model underfit แต่เมื่อเพิ่มจำนวนรอบการฝึกสอนให้มากขึ้นยิ่งมากขึ้นเท่าไร Model เกิดปัญหา Overfit ทำให้เปอร์เซ็นต์ความแม่นยำของโมเดลต่ำลงเหมือนกัน ดังนั้น ผู้วิจัยได้ทำการทดลองต่าง ๆ ปรับเปลี่ยนค่าต่าง ๆ ตรวจสอบข้อมูลว่าเกิดอะไรขึ้นเปอร์เซ็นต์ความถูกต้องของโมเดลถึงไม่ได้สูงอย่างที่คาดหวังไว้ผู้วิจัยได้สังเกตเห็นขอบบนและล่างของภาพว่ามีการปรับเปลี่ยนอยู่ตลอดเวลาลักษณะเด่นของขอบบนและล่างของภาพ HDE จึง

มีการเปลี่ยนไปมาดังตัวอย่างในรูปที่ 3.9 ดังนั้น จึงเกิดแนวคิดเพิ่มเติมเข้ามาในขั้นตอนการเตรียมข้อมูลและทดสอบระบบ VI system คือ การ cropping ภาพขอบบนและขอบล่างของภาพ HDE view ออกโดยจะทำการ cropping ภาพจาก 1280x1024 pixel เป็น 1280x814 pixel ดังตัวอย่างในรูปที่ 3.10

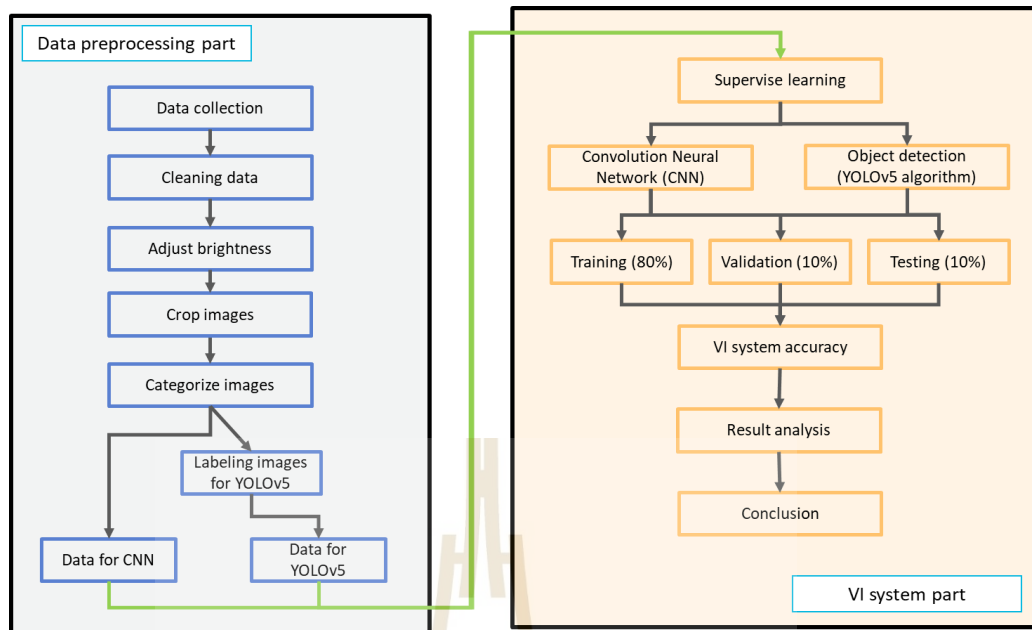


รูปที่ 3.9 ขอบบนและล่างของภาพ HDE view

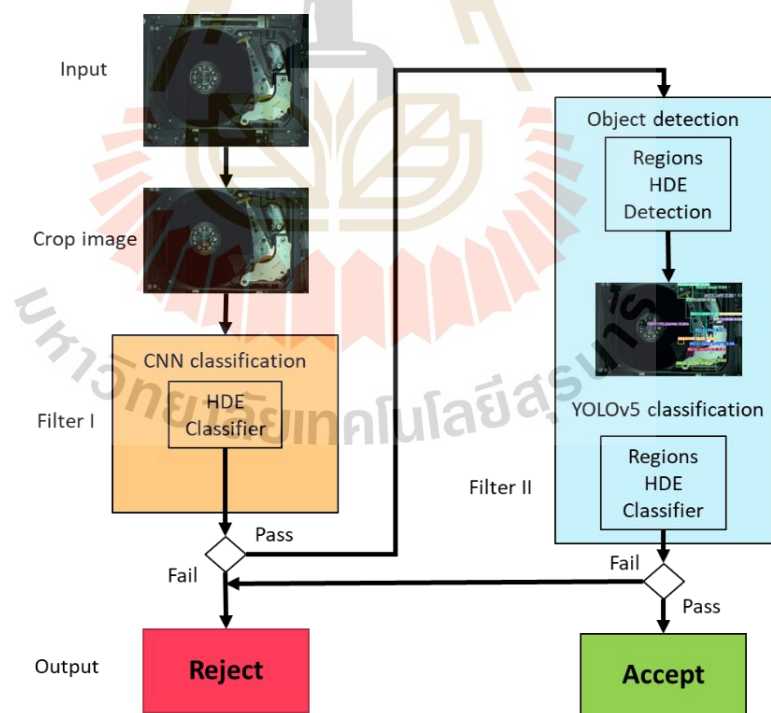


รูปที่ 3.10 การ cropping ภาพ HDE view

จากการปรับเปลี่ยนขั้นตอนการเตรียมข้อมูลสำหรับการฝึกสอนโมเดลใหม่จึงได้แผนผังการทำงานใหม่ดังรูปที่ 3.11 สำหรับสถาปัตยกรรมระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (Visual inspection of HDE components system architecture) นั้น ก็มีการปรับเปลี่ยนด้วยเพิ่มขั้นตอนการ cropping ภาพ HDE view ส่งข้อมูล input ไปยังตัวคัดกรองที่หนึ่ง CNN โดยสถาปัตยกรรมแบบใหม่แสดงดังรูปที่ 3.12



รูปที่ 3.11 แผนผังการดำเนินการทดลอง VI system ใหม่



รูปที่ 3.12 สถาปัตยกรรม VI system แบบใหม่

เมื่อได้ออกแบบสถาปัตยกรรมระบบ VI system แบบใหม่เรียบร้อยแล้วผู้วิจัยจึงได้ทำการสร้างโมเดล CNN ใหม่อีกครั้งด้วยข้อมูลชุดเดิมและได้ทำการปรับเปลี่ยนค่าตัวจำนวนรอบการฝึกสอน (epoch) และขนาดภาพโดยไม่ต้องทำการ resize image เพื่อฝึกสอนโมเดล CNN ดังนั้น ขนาดภาพที่ใช้ในการฝึกสอนคือ 1280x814 pixel ดังตารางที่ 3.3

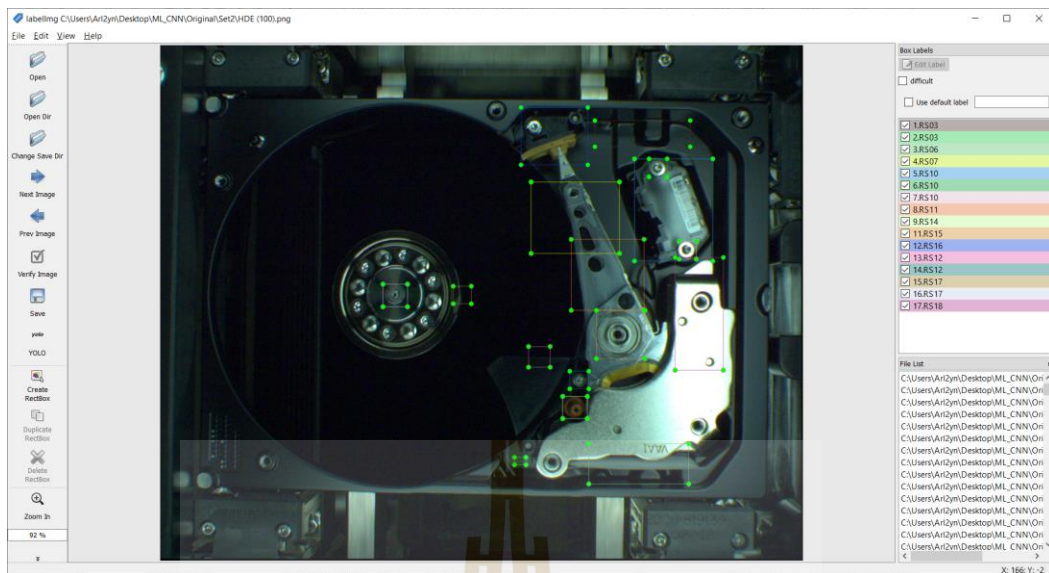
ตารางที่ 3.3 ตารางเปรียบเทียบการสร้างโมเดล CNN หลังจากการปรับรูปภาพเพื่อฝึกสอนใหม่

| CNN model training configuration |                    |       |            |              |                    |                 |           |
|----------------------------------|--------------------|-------|------------|--------------|--------------------|-----------------|-----------|
| No. model                        | Image size (pixel) | Epoch | Batch size | Conv. Layers | Max pooling layers | Activation Fcn. | %Accuracy |
| 7                                | 1280x814           | 10    | 16         | 64           | 2x2                | ReLU            | 99.2%     |
| 8                                | 1280x814           | 20    | 16         | 64           | 2x2                | ReLU            | 99.8%     |
| 9                                | 1280x814           | 40    | 16         | 64           | 2x2                | ReLU            | 99.8%     |
| 10                               | 1280x814           | 60    | 16         | 64           | 2x2                | ReLU            | 99.8%     |
| 11                               | 1280x814           | 80    | 16         | 64           | 2x2                | ReLU            | 99.8%     |
| 12                               | 1280x814           | 100   | 16         | 64           | 2x2                | ReLU            | 99.8%     |

จากการสร้างโมเดล CNN ใหม่ด้วยการครอบรูป HDE view จะเห็นได้ว่าเปอร์เซ็นต์ความถูกต้องของโมเดลนั้นสูงขึ้นถึงเป้าหมายที่ตั้งไว้โดยมีเปอร์เซ็นต์ความแม่นยำสูงถึง 99.8%

### 3.3.7 การสร้างโมเดล YOLOv5 (Create YOLOv5 object detection)

ขั้นตอนต่อมาคือการสร้างโมเดล YOLOv5 ใช้สำหรับเป็นตัวกรองที่สองของระบบ เมื่อรับข้อมูล input ที่ได้จาก CNN มาจะทำการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ในตำแหน่งต่าง ๆ ทั้งหมด 16 ตำแหน่งโดยภาพที่จะใช้ในการฝึกสอนโมเดล นั้น จะต้องเป็นภาพ Pass images ทั้งหมดและทำการสร้าง Label ของภาพไว้ด้วยเพื่อระบุตำแหน่งขององค์ประกอบฮาร์ดดิสก์ไดรฟ์แสดงดังรูปที่ 3.13



รูปที่ 3.13 การสร้าง Label image HDE view

และจะทำการแบ่งข้อมูลสำหรับสร้างโมเดล YOLOv5 ดังรูปที่ 3.144 และข้อมูลที่ใช้ในการฝึกสอนโมเดล YOLOv5 นั้น มีดังนี้

YOLOv5 model training

Image size = resize from 1280x1024 to 736x538 pixel

YOLOv5 model S

Training (80%) = 2340 images

Validation (10%) = 293 images

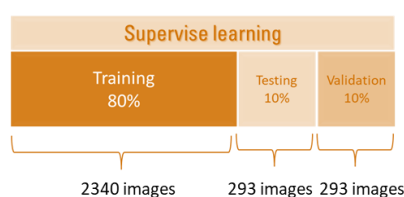
Testing (10%) = 293 images

Activation function = ReLU function

Confidence threshold = 80%

Regions = 16 regions

Group regions = 11 groups (RS code)



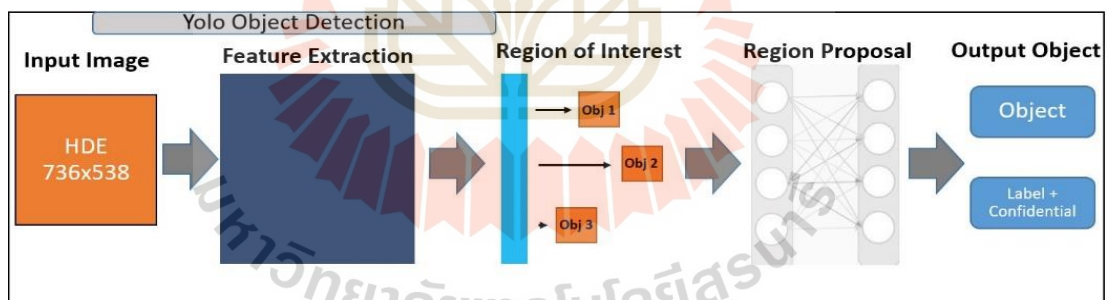
รูปที่ 3.14 การแบ่งข้อมูลสำหรับสร้างโมเดล YOLOv5

ตำแหน่งที่จะทำการตรวจสอบมีทั้งหมด 16 ตำแหน่งโดยจะแบ่งรหัสออกเป็น 11 กลุ่มตามรหัส RS code ดังแสดงในรูปที่ 3.15

| Camera | Parameter       | EC | RS   | G-code | Note                     |
|--------|-----------------|----|------|--------|--------------------------|
| HDE    | 1 BASE1/BASE2   | 3  | RS03 | 3279   | Extra parts in MBA       |
|        | 2 EXTRA         | 3  | RS03 | 3279   | Extra parts in MBA       |
|        | 3 C-Filter      | 6  | RS06 | 3281   | C-Filter missing         |
|        | 4 CLIP          | 7  | RS07 | 3282   | Comb not remove          |
|        | 5 FPC           | 10 | RS10 | 3284   | FCC or screw problem     |
|        | 6 FPCSCR1       | 10 | RS10 | 3284   | FCC or screw problem     |
|        | 7 FPCSCR2       | 10 | RS10 | 3284   | FCC or screw problem     |
|        | 8 HSA           | 11 | RS11 | 3285   | HSA damage or no HSA     |
|        | 9 OUTERCS       | 14 | RS14 | 3287   | ODCS damage or missing   |
|        | 10 Pivot        | 15 | RS15 | 3288   | Pivot abnormal or damage |
|        | 11 RAMP         | 16 | RS16 | 3437   | Ramp missing or damage   |
|        | 12 SPOILER      | 12 | RS12 | 3431   | Spoiler or screw missing |
|        | 13 SpoilerScrew | 12 | RS12 | 3431   | Spoiler or screw missing |
|        | 14 TC           | 17 | RS17 | 3290   | Top clamp missing        |
|        | 15 TC_Center    | 17 | RS17 | 3290   | Top clamp missing        |
|        | 16 VCM          | 18 | RS18 | 3291   | TVCM missing             |

รูปที่ 3.15 รหัส RS code และตำแหน่งตรวจสอบของภาพ HDE view

แผนผังการฝึกสอนโมเดล YOLOv5 แสดงดังรูปที่ 3.16



3.16 แผนผังการฝึกสอนโมเดล YOLOv5

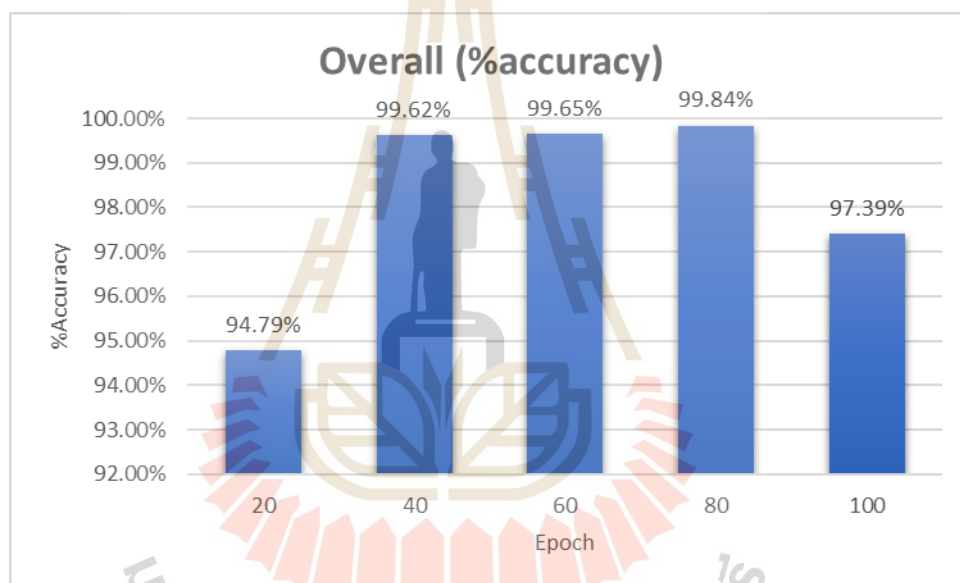
สำหรับโมเดล YOLOv5 ซึ่งมีหน้าที่ในการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ในช่วงตัวกรองที่สองของ VI system โดยที่มีการปรับเปลี่ยนการตั้งค่าตัวแปรต่าง ๆ เพื่อให้ได้โมเดลที่มีความแม่นยำในการตรวจจับองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ได้แม่นยำที่สุดโดยสุดท้ายแล้วจะเหลือตัวแปรที่ต้องปรับเพื่อเปรียบเทียบประสิทธิภาพของโมเดลคือจำนวนรอบการฝึกสอนโมเดล (epoch) เช่นเดียวกับการฝึกสอนโมเดล CNN โดยจะแสดงดังตารางที่ 3.4



ตารางที่ 3.4 ตารางแสดงผลการฝึกสอนโมเดล YOLOv5

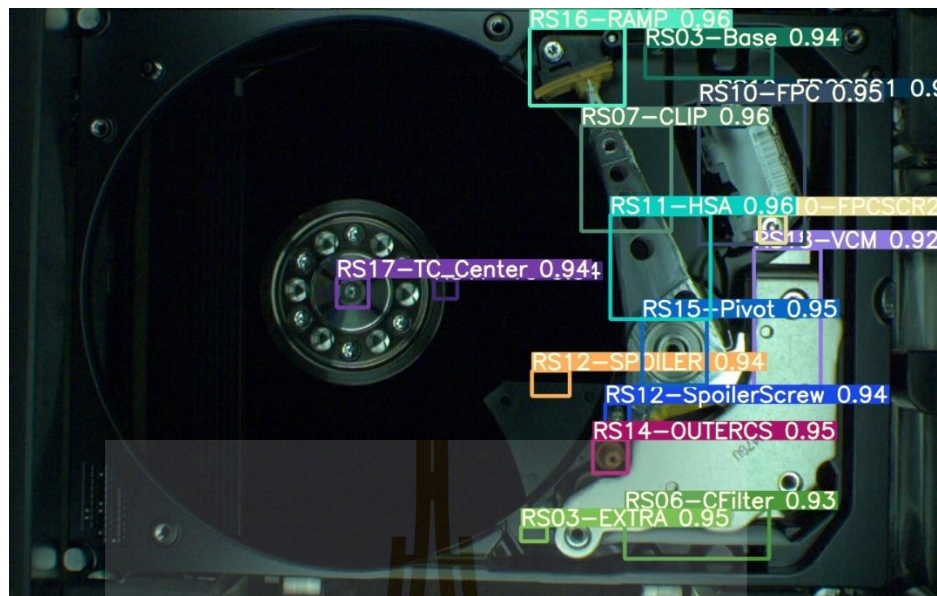
| Model          | Epoch | Batch size | Img. Size | Training time | Detection time | Good (%Accuracy) | Failed (%Accuracy) | Overkill (%Accuracy) | Overall (%accuracy) |
|----------------|-------|------------|-----------|---------------|----------------|------------------|--------------------|----------------------|---------------------|
| YOLOv5 model S | 20    | 4          | 738       | 12 mins       | 20-30 ms/img   | 95%              | 93.60%             | 95.77%               | 94.79%              |
|                | 40    | 4          | 738       | 24 mins       | 20-30 ms/img   | 99.85%           | 99.44%             | 99.56%               | 99.62%              |
|                | 60    | 4          | 738       | 36 mins       | 20-30 ms/img   | 99.88%           | 99.44%             | 99.62%               | 99.65%              |
|                | 80    | 4          | 738       | 48 mins       | 20-30 ms/img   | 99.90%           | 99.80%             | 99.81%               | 99.84%              |
|                | 100   | 4          | 738       | 1 Hr.         | 20-30 ms/img   | 96.76%           | 97.94%             | 97.48%               | 97.39%              |

จากตารางที่ 3.4 จะเห็นได้ว่าการฝึกสอนโมเดล YOLOv5 สำหรับตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์นั้นมีเปอร์เซ็นต์ความแม่นยำที่สูงเท่าๆ กันแต่เนื่องจากที่รอบการฝึกสอนที่ 80 มีเปอร์เซ็นต์ความแม่นยำโดยรวมสูงที่สุดที่ 99.84% ซึ่งสรุปออกมาเป็นกราฟดังรูปที่ 3.17 เพื่อให้เห็นภาพว่าที่จำนวนรอบการฝึกสอนที่ให้เปอร์เซ็นต์ความแม่นยำสูงที่สุด



รูปที่ 3.17 กราฟแสดงการเปรียบเทียบเปอร์เซ็นต์ความถูกต้องของโมเดล YOLOv5

โดยปกติแล้วการเปอร์เซ็นต์ความถูกต้องของการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์นั้นขึ้นอยู่กับค่าการตั้งค่าของค่าความเชื่อมั่น (Confidential threshold) ด้วย ซึ่งโมเดล YOLOv5 จะแสดงเปอร์เซ็นต์ของค่าความเชื่อมั่นออกมาเป็นตัวเลขหากมีค่าเข้าใกล้หนึ่งแสดงว่ามีความถูกต้องสูงมากดังแสดงในรูปที่ 3.18



รูปที่ 3.18 ภาพตัวอย่างของการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์

### 3.4 สรุป

วิธีการดำเนินการวิจัยจะต้องมีความรู้ความเข้าใจเกี่ยวกับทฤษฎีที่เกี่ยวข้องและการออกแบบวางแผนให้สามารถดำเนินการวิจัยไปได้ตามที่ตั้งวัตถุประสงค์ไว้ และยังต้องมีศึกษาเรียนรู้เทคนิคใหม่ ๆ อยู่ตลอดเวลาเพื่อที่จะสามารถนำมาปรับปรุงประยุกต์ใช้กับงานวิจัยแก้ไขปัญหาต่าง ๆ ที่อาจเกิดขึ้นระหว่างการทำวิจัยได้โดยหลักการทำงานพื้นฐานเกี่ยวกับซอฟต์แวร์ และนำมาประยุกต์ใช้จนบรรลุวัตถุประสงค์ได้นั้น แบ่งออกเป็น 3 ส่วนดังนี้

ส่วนที่ 1 คือการศึกษาปัญหาเกี่ยวกับการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ในไลน์การประกอบฮาร์ดดิสก์ไดรฟ์และเข้าใจถึงขั้นตอนการตรวจสอบความผิดปกติขององค์ประกอบเหล่านั้นตลอดจนถึงระบบเดิมที่ใช้ในการตรวจสอบเพื่อให้เข้าใจถึงปัญหาที่เกิดขึ้นและค้นหาวิธีการที่จะเพิ่มประสิทธิภาพในการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ให้มีประสิทธิภาพมากขึ้นรวดเร็วยิ่งขึ้น อาจจะได้ด้วยปรับปรุงระบบเดิมหรือ ด้วยการออกแบบระบบใหม่ขึ้นมาทดแทนระบบเดิมให้มีประสิทธิภาพมากขึ้นดังเช่นงานวิจัยนี้เสนอให้ใช้เทคนิคการเรียนรู้เชิงลึก (Deep learning technique) ในการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์แทนระบบเดิมที่เป็นแบบการมองเห็นของเครื่องจักร (Machine vision)

ส่วนที่ 2 คือการศึกษาทฤษฎีและความรู้ที่เกี่ยวข้องกับการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ไม่ว่าจะเป็นความรู้ด้านการประมวลผลภาพ (Image processing) ความรู้ด้านกาเรียนรู้ของเครื่องจักร (Machine learning) ความรู้ด้านการตรวจจับวัตถุภายในภาพ (Object

detection) โดยจะต้องเข้าใจถึงทฤษฎีและความรู้ต่างๆ เหล่านั้นเป็นอย่างดีและศึกษาปริทัศน์วรรณกรรมความรู้ที่นักวิจัยผู้อื่นได้ทำการทดลองศึกษามาก่อนหน้านี้ที่เป็นเนื้อหาใกล้เคียงกับงานวิจัยของเราตลอดจนความรู้ด้านการเขียนโปรแกรม python เพื่อใช้ในการสร้างและทดสอบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (Visual inspection of HDE components)

ส่วนที่ 3 คือการดำเนินการทดลองไปตามที่ได้วางแผนไว้โดยจะเริ่มจากการเก็บรวบรวมข้อมูลภาพ HDE view และทำการตรวจสอบคัดแยกให้ได้ข้อมูลที่ต้องการไม่ปะปนกันเพื่อที่จะนำข้อมูลเหล่านั้นไปทำการเตรียมข้อมูลให้พร้อมสำหรับการฝึกสอนโมเดลทั้ง CNN และ YOLOv5 ต่อไป โดยจะทำการแบ่งประเภทรูปภาพออกเป็นภาพดี (Pass images) และภาพเสีย (Fail images) โดยจะมีขั้นตอนการออกแบบและสร้างระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ดังนี้

ขั้นตอนที่ 1 คือการออกแบบแผนผังการทำงานเพื่อให้ผู้วิจัยดำเนินการได้ตามแผนที่วางไว้ โดยจะเริ่มจากการออกแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ซึ่งจะใช้เทคนิคการเรียนรู้เชิงลึกในการออกแบบโดยจะแบ่งออกเป็นสองส่วนด้วยกันคือ ส่วนที่หนึ่งคือการคัดกรองรูปภาพรวมด้วยเทคนิคโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network : CNN) และในส่วนที่สองคือการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ทั้งหมด 16 ตำแหน่งโดยใช้เทคนิคการตรวจจับวัตถุภายในภาพ (Object detection) โดยเลือกใช้ YOLOv5 algorithm ในการทำงานเนื่องจากมีการทำงานที่รวดเร็วและมีประสิทธิภาพที่ดี

ขั้นตอนที่ 2 คือการสร้างโมเดล CNN และ YOLOv5 โดยการนำภาพ HDE view ที่ได้ทำการเตรียมไว้แล้วมาฝึกสอนโมเดลทั้งสองโดยจะแบ่งเป็น Train 80%, Validation 10%, และ Test 10% จากนั้น จะทำการทดสอบโมเดลว่ามีประสิทธิภาพการทำงานที่แม่นยำหรือยัง และปรับเปลี่ยนค่าตัวแปรต่าง ๆ เพื่อค้นหาโมเดลที่ดีที่สุดโดยจะปรับค่าตัวแปรต่าง ๆ ดังนี้

- ขนาดของรูปภาพ HDE view
- จำนวนรอบการฝึกสอนโมเดล epoch
- Batch size การฝึกสอนโมเดล
- จำนวน Conv. Layers

เป็นต้นสำหรับ CNN นั้น ในขั้นตอนสุดท้ายที่ทางผู้วิจัยไม่สามารถทำให้ประสิทธิภาพความแม่นยำมากขึ้นได้เนื่องจากปัญหาขอบบน และขอบล่างของภาพ HDE view มีการปรับเปลี่ยนอยู่ตลอดเวลาทางผู้วิจัยจึงได้ใช้เทคนิคการครอปรูปภาพเข้าใจใช้จึงทำให้ประสิทธิภาพความแม่นยำในการทำงานของโมเดล CNN สูงขึ้นเป็นไปตามวัตถุประสงค์ที่ตั้งไว้

ขั้นตอนที่ 3 ประเมินประสิทธิภาพการทำงานของระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยชุดภาพ HDE view ที่โมเดลไม่เคยเจอมาก่อนทั้งหมด 4 ชุดโดยจะแบ่งภาพออกเป็น Pass images = 375 ภาพ และ Fail images = 115 ภาพ ทั้งหมด 4 ชุดเท่า ๆ จากนั้นทำ

การทดสอบระบบ VI system และทำการประเมินประสิทธิภาพของโมเดลด้วย Confusion matrix และคะแนน F1-Score ของระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์

ขั้นตอนที่ 4 สรุปผลการทดลอง



## บทที่ 4

### ผลการทดลอง และการวิเคราะห์ผลการทดลอง

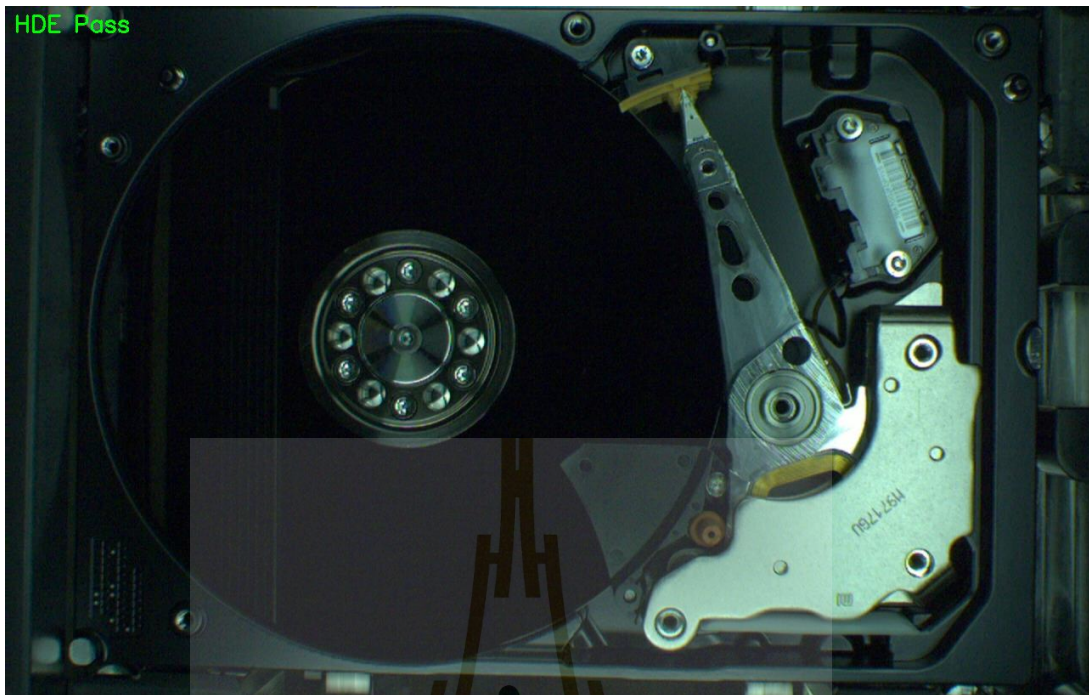
#### 4.1 บทนำ

ในบทนี้จะเป็นการทดสอบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึกสำหรับภาพฮาร์ดดิสก์ไดรฟ์ HDE view โดยจะทำการเปรียบเทียบประสิทธิภาพของ VI system ที่ใช้การออกแบบเดิมโดยการฝึกสอนโมเดล CNN และ YOLOv5 ด้วยภาพต้นฉบับไม่มีการครอบภาพเหมือนกับการออกแบบล่าสุดเพื่อให้เห็นถึงประสิทธิภาพในแยกภาพ HDE view รวมถึงการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วย YOLOv5 โดยจะมีชุดข้อมูลทดสอบทั้งหมด 4 ชุดประกอบไปด้วย Pass images = 375 ภาพและ Fail images = 115 ภาพในแต่ละชุดแบ่งเท่าๆ กัน โดยจะทำการทดสอบและประเมินประสิทธิภาพของระบบ VI system ด้วย confusion matrix และ คะแนน F1-score ของผลทดสอบนั้นๆ

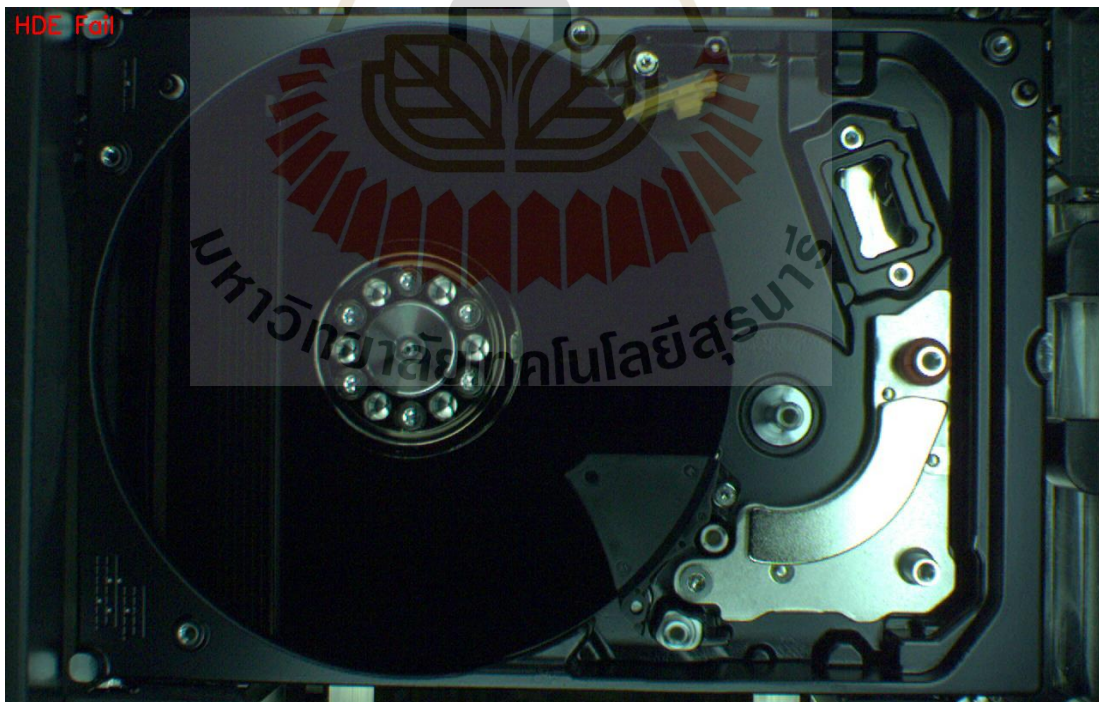
#### 4.2 ผลการตรวจสอบภาพรวมด้วย CNN

ผลการตรวจสอบที่ตัวคัดกรองแรกของ VI system คือการตรวจสอบด้วยโมเดล CNN ซึ่ง จะทำการรับข้อมูล Input เข้ามาแล้วเป็นภาพ HDE view ต้นฉบับและทำการครอบรูปภาพจาก 1280x1024 pixel เป็น 1280x814 pixel ก่อนหลังจากนั้นจะนำข้อมูลที่ถูกครอบแล้วเข้าสู่กระบวนการประมวลผลด้วยโครงข่ายประสาทเทียมแบบคอนโวลูชัน และทำการคัดแยกภาพ HDE view ว่ามีความถูกต้องหรือไม่ มีตำแหน่งองค์ประกอบใดภายในภาพผิดปกติหรือไม่โดยจะแยกภาพออกเป็น HDE Pass สำหรับภาพที่ปกติดังแสดงในรูปที่ 4.1 และ HDE Fail สำหรับภาพที่ผิดปกติดังแสดงในรูปที่ 4.2

ในส่วนของ CNN นั้นจะทำการคัดกรองเบื้องต้นเท่านั้นสำหรับภาพ HDE view ที่มีตำแหน่งองค์ประกอบผิดปกติชัดเจนภายในภาพและระบบสามารถคัดกรองออกมาได้เป็นอย่างดี



รูปที่ 4.1 ภาพ HDE view Pass image จากโมเดล CNN



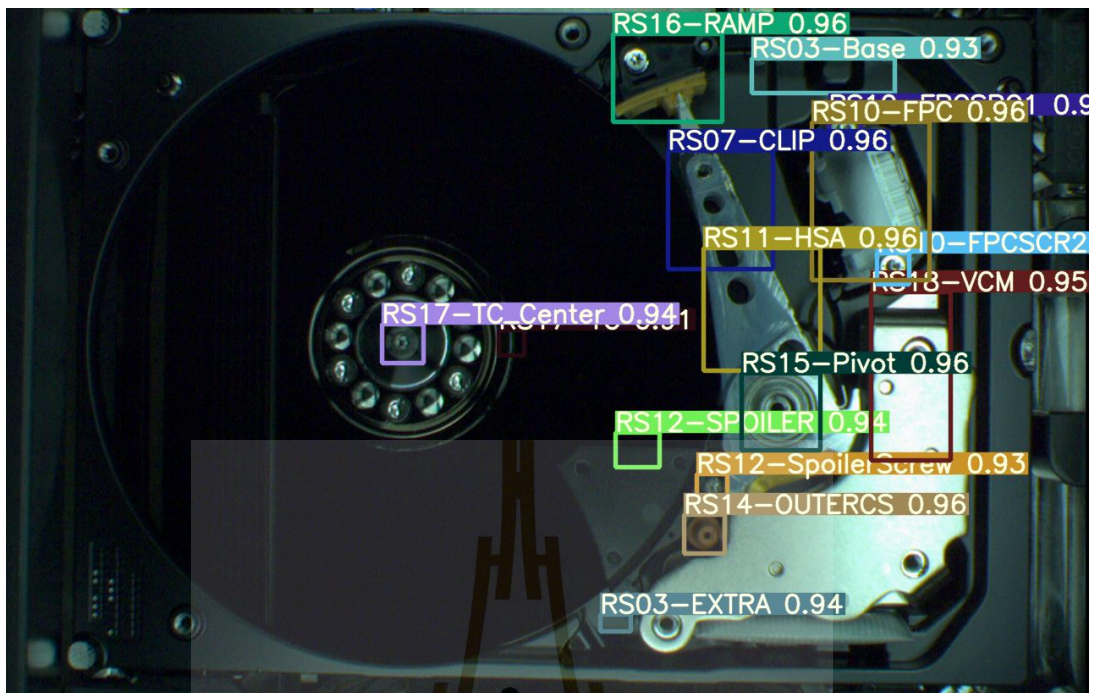
รูปที่ 4.2 ภาพ HDE view Fail image จากโมเดล CNN

### 4.3 ผลการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไทรฟ์ด้วย YOLOv5

หลังจากข้อมูล input ที่ได้รับเข้ามาผ่านตัวกรองที่หนึ่งอย่าง CNN แล้วระบบจะส่งภาพที่เป็น Pass image ส่งเข้ามายังระบบ Object detection เพื่อทำการตรวจสอบองค์ประกอบต่างๆ ที่ได้ฝึกสอนไว้ทั้งหมด 16 ตำแหน่งโดยระบบจะทำการ resize ภาพจาก input ที่ได้รับเข้ามาให้เป็นภาพขนาด 736x538 pixel ก่อนจะทำการตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไทรฟ์ภายในภาพต่อไปโดยระบบได้กำหนดค่าความเชื่อมั่นหรือ Confidential threshold ไว้ที่ 80% YOLOv5 จะทำการตรวจสอบไปยังตำแหน่งต่างๆ และเทียบค่าความเชื่อมั่นว่าสูงถึงระดับที่ตั้งค่าไว้หรือไม่ หากระบบตรวจสอบครบทั้ง 16 ตำแหน่งแล้วค่าความเชื่อมั่นสูงเท่ากับค่าที่ตั้งไว้ทุกตำแหน่งภาพ HDE view นั้นจะเป็น Pass image หากภาพ HDE view นั้นมีตำแหน่งที่ผิดปกติไปทำให้ค่าความเชื่อมั่นไม่ถึงเกณฑ์ที่ตั้งไว้ภาพ HDE view นั้นก็จะเป็น Fail image โดยที่ จะแสดงภาพ Pass image และ Fail image ของโมเดล YOLOv5 ดังรูปที่ 4.3 และ 4.4 ตามลำดับ



รูปที่ 4.3 ภาพ HDE view Pass image จาก YOLOv5



รูปที่ 4.4 ภาพ HDE view Fail image จาก YOLOv5

จากรูปที่ 4.4 จะเห็นได้ว่ามีตำแหน่งที่ไม่ได้ถูกตีกรอบไว้คือตำแหน่ง Filter (RS-06) ซึ่งจากภาพจะเห็นว่าความผิดปกตินั้นคือ filter คู่มือผิดปกติโดยในความเป็นจริงแล้วอาจจะเกิดขึ้นเนื่องจากปัจจัยภายนอกก็เป็นได้

#### 4.4 ผลการทดสอบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ (HDE view)

จากการทดลองสร้างโมเดล CNN จากไฟล์ข้อมูลภาพ HDE view ต้นฉบับได้โมเดลที่มีประสิทธิภาพความแม่นยำในการทำนายที่สูงที่สุดคือ 76.79% ที่โมเดล 20 epoch เนื่องจากภาพต้นฉบับของ HDE view มีสีของภาพขอบบนและขอบล่างที่ไม่สม่ำเสมอจึงทำให้การทำนายนั้นมีความผิดพลาดเกิดขึ้นได้แต่เมื่อทำการปรับปรุงและออกแบบการสร้างโมเดลใหม่รวมถึงการทดสอบระบบ VI system ใหม่โดยการครอปรูปภาพ HDE view และทำการทดสอบระบบใหม่อีกครั้งสิ่งที่เกิดขึ้นคือโมเดล CNN มีความผิดพลาดน้อยลงเป็นอย่างมากโดยผู้วิจัยได้ทำการสร้างโมเดล CNN ใหม่ขึ้นด้วยการปรับเปลี่ยนจำนวนรอบการฝึกสอนเป็น 10 epoch, 20 epoch, 40 epoch, 60 epoch, 80 epoch, และ 100 epoch ตามลำดับผลที่ได้คือโมเดล CNN ตั้งแต่รอบการฝึกสอน 20 epoch มีประสิทธิภาพในการตัดแยกภาพ Pass image และ Fail image อยู่ที่ 99.8%



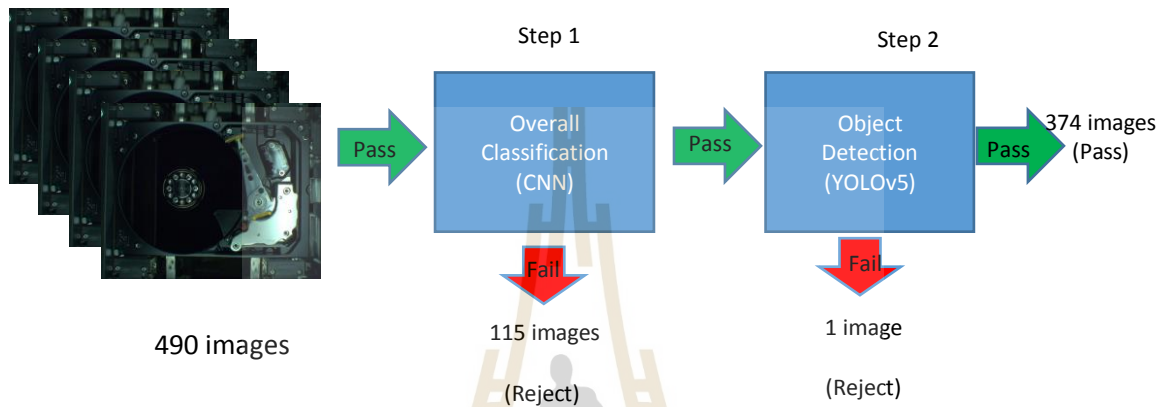
#### 4.4.1 ผลการทดสอบระบบ VI system กับข้อมูลชุดที่ 1

##### HDE view test set 1

Pass images = 375 images

Fail images = 115 images

Dataset 1 test result (Set1)



รูปที่ 4.5 Flow chart แสดงการทดสอบ VI system กับ Dataset1

Pass = 374 images

Fail = 116 images

Confusion matrix result

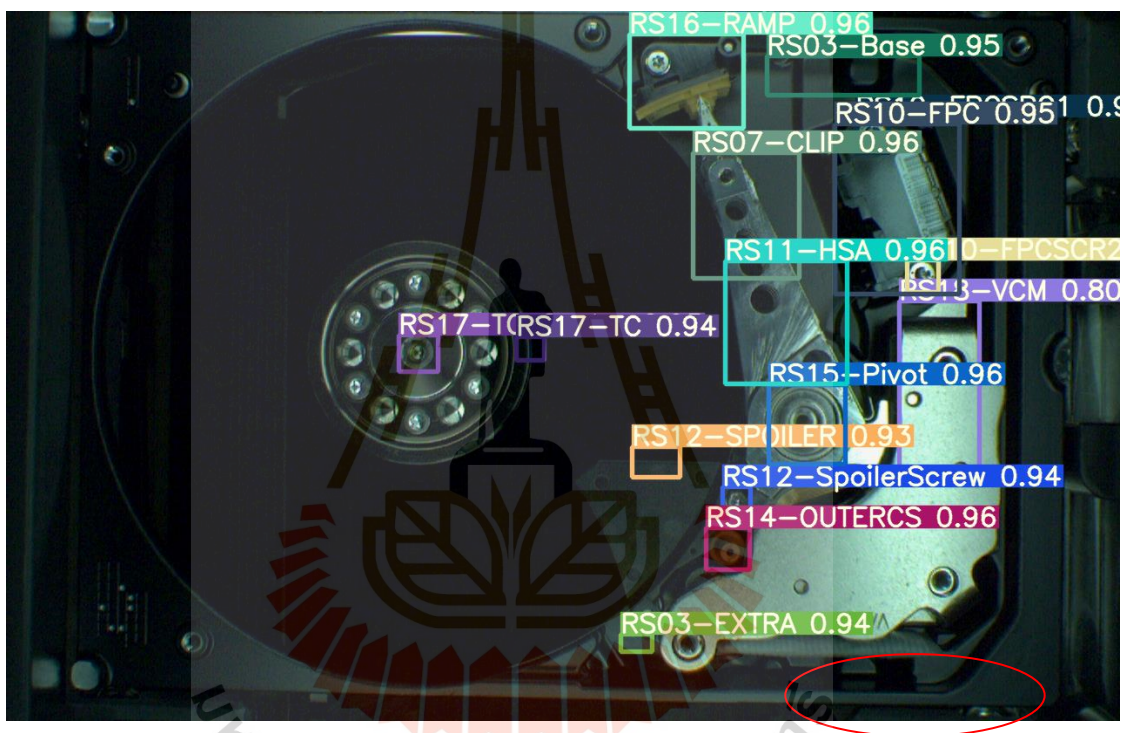
| HDE test (490 images) |                      | Predicted |       |                   |
|-----------------------|----------------------|-----------|-------|-------------------|
|                       |                      | TRUE      | FALSE | accuracy (Recall) |
| Actual                | TRUE                 | 374       | 0     | 100%              |
|                       | FALSE                | 1         | 115   | 99.138%           |
|                       | accuracy (Precision) | 99.733%   | 100%  | 99.796%           |

รูปที่ 4.6 Confusion matrix สำหรับ VI system กับ Dataset1

**%Accuracy of model = 99.796%**

**F1-Score = 0.9987**

จาก Confusion matrix ของการทดสอบ VI system กับข้อมูลชุดที่หนึ่งพบว่ามีภาพ HDE view ในความเป็นจริงคือภาพที่ดีแต่ระบบกลับมองว่าเป็น Fail image แสดงในรูปที่ 4.7 ซึ่งเป็น Overkill image อาจเกิดขึ้นได้เนื่องจากตรงตำแหน่ง Filter (RS-06) นั้นมีความผิดปกติเกิดขึ้นอาจจะได้รับการรบกวนจากปัจจัยภายนอกจึงทำให้ภาพดูผิดเพี้ยนไปดังแสดงในรูปที่ 4.7



รูปที่ 4.7 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset1

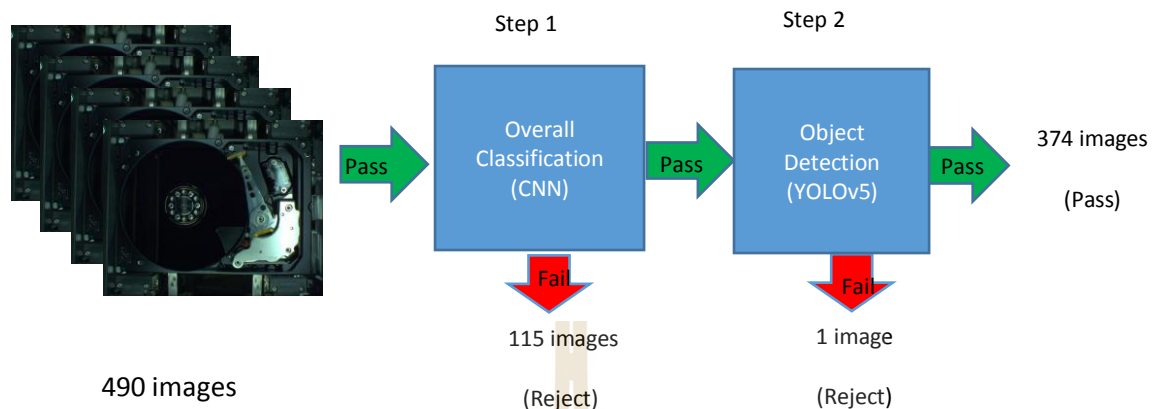
#### 4.4.2 ผลการทดสอบระบบ VI system กับข้อมูลชุดที่ 2

**HDE view test set 2**

Pass images = 375 images

Fail images = 115 images

## Dataset 2 test result (Set1)



รูปที่ 4.8 Flow chart แสดงการทดสอบ VI system กับ Dataset2

Pass = 374 images

Fail = 116 images

Confusion matrix result

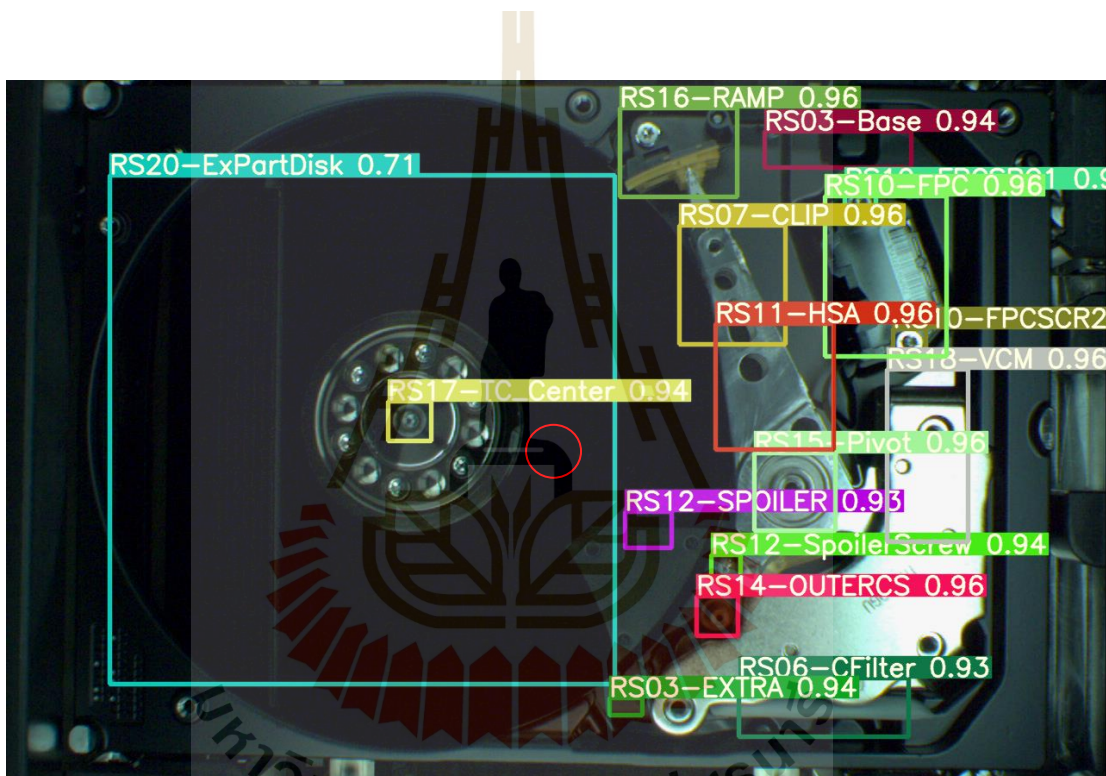
| HDE test (490 images) |                      | Predicted |       |                   |
|-----------------------|----------------------|-----------|-------|-------------------|
|                       |                      | TRUE      | FALSE | accuracy (Recall) |
| Actual                | TRUE                 | 374       | 0     | 100%              |
|                       | FALSE                | 1         | 115   | 99.138%           |
|                       | accuracy (Precision) | 99.733%   | 100%  | 99.796%           |

รูปที่ 4.9 Confusion matrix สำหรับ VI system กับ Dataset2

**%Accuracy of model = 99.796%**

**F1-Score = 0.9987**

จาก Confusion matrix ของการทดสอบ VI system กับข้อมูลชุดที่สองพบว่า มีภาพ HDE view ในความเป็นจริงคือภาพที่ดีแต่ระบบกลับมองว่าเป็น Fail image แสดงในรูปแบบที่ ซึ่งเป็น Overkill image เกิดขึ้นเนื่องจากตำแหน่งของ Top Clamp (RS-17) วงรอบนอกมีผิวที่ไม่เรียบซึ่งในความเป็นจริงแล้วตำแหน่งนั้นคือ Pass image ไม่ได้เกิดความผิดปกติแต่อย่างใด แต่โมเดล YOLOv5 เข้าใจว่าผิดปกติ จึงทำให้ไม่สามารถ detect ได้ดังที่แสดงในรูปแบบที่ 4.10



รูปที่ 4.10 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset2

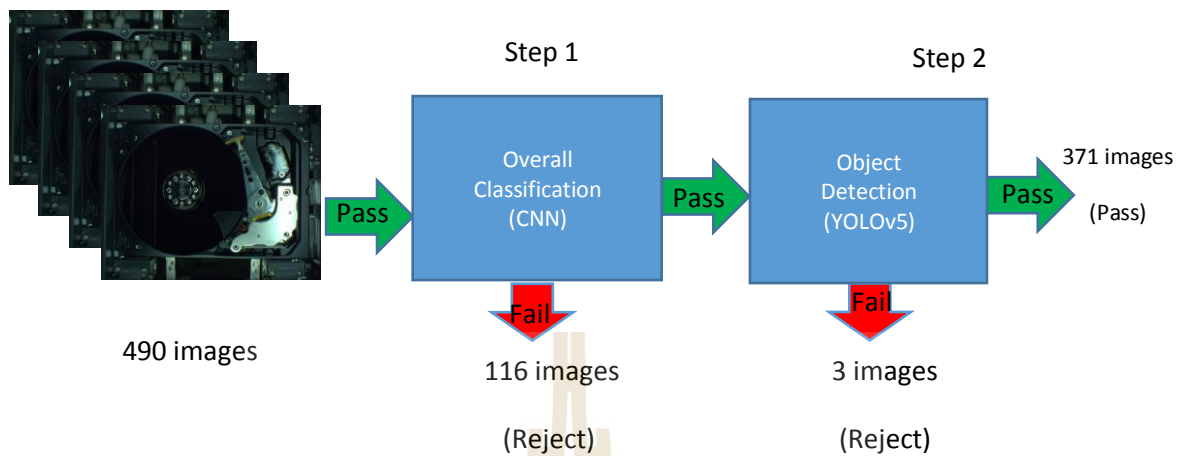
#### 4.4.3 ผลการทดสอบระบบ VI system กับข้อมูลชุดที่ 3

**HDE view test set 3**

Pass images = 375 images

Fail images = 115 images

## Dataset 3 test result (Set1)



รูปที่ 4.11 Flow chart แสดงการทดสอบ VI system กับ Dataset3

Pass = 371 images

Fail = 119 images

Confusion matrix result

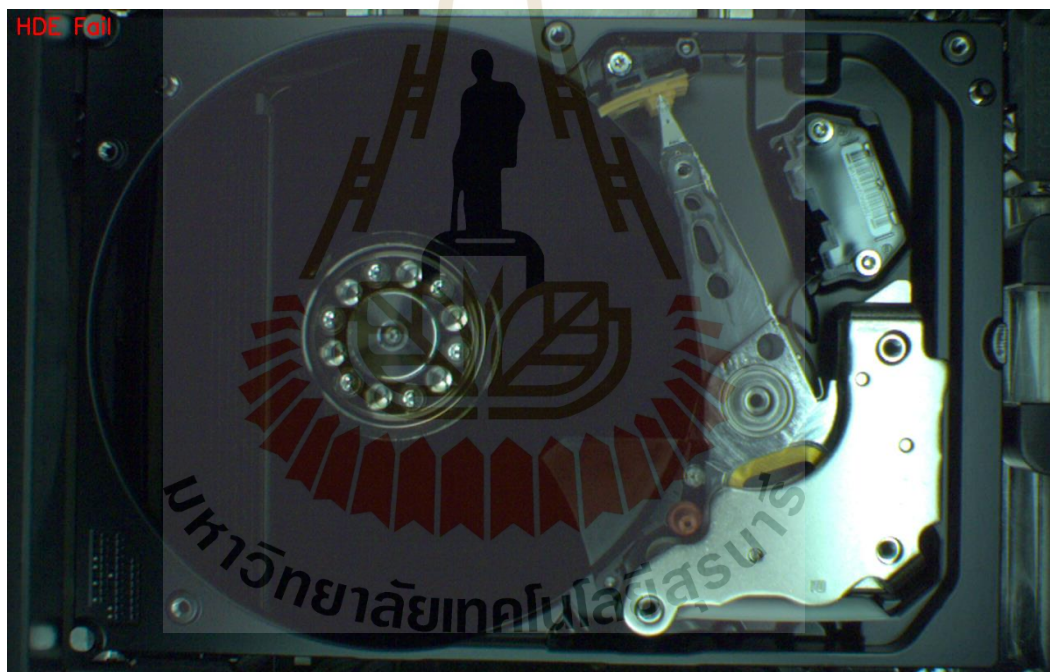
| HDE test (490 images) |                      | Predicted |       |                   |
|-----------------------|----------------------|-----------|-------|-------------------|
|                       |                      | TRUE      | FALSE | accuracy (Recall) |
| Actual                | TRUE                 | 371       | 0     | 100%              |
|                       | FALSE                | 4         | 115   | 96.639%           |
|                       | accuracy (Precision) | 99.933%   | 100%  | 99.184%           |

รูปที่ 4.12 Confusion matrix สำหรับ VI system กับ Dataset3

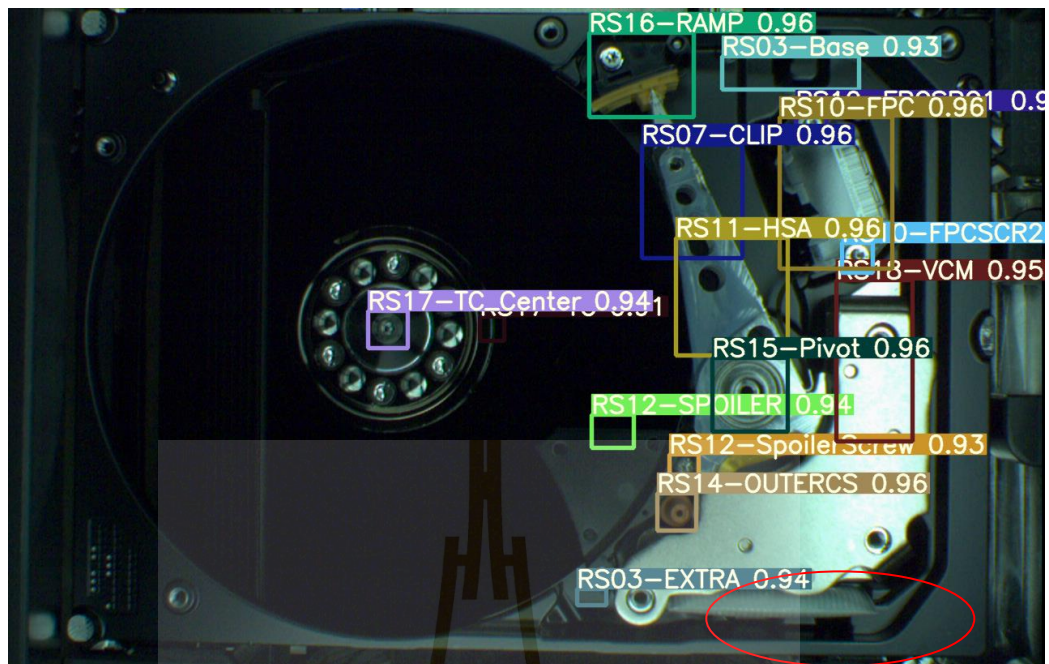
**%Accuracy of model = 99.184%**

**F1-Score = 0.9946**

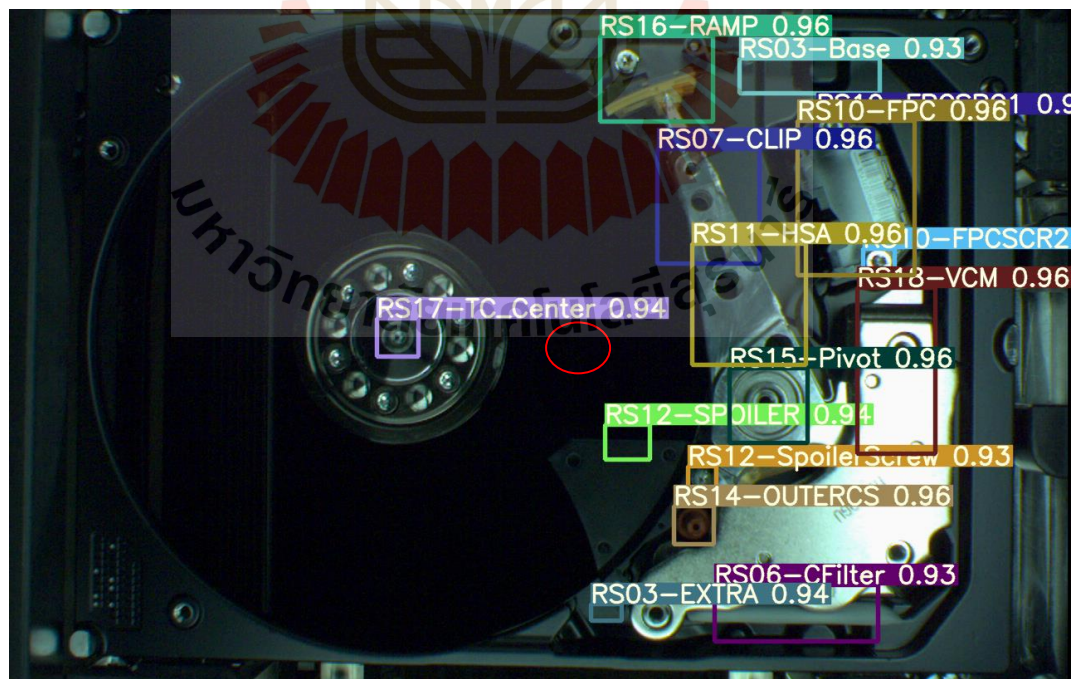
ผลการทดสอบระบบ VI system กับชุดทดสอบที่ 3 นั้นมีภาพ HDE view ที่เกิด Overkill images ทั้งหมด 4 ภาพโดยภาพที่หนึ่งนั้นแสดงให้เห็นในรูปที่ 4.13 เป็นการทำนายผิดพลาดของโมเดล CNN โดยระบบมองว่าภาพเป็น Fail image ทั้งๆ ที่เป็น Pass image อาจเกิดขึ้นจากค่าเฉลี่ยโดยรวมของภาพไม่เข้าข่ายข้อมูลที่ทำการฝึกสอนโมเดล CNN ไว้จึงทำให้ CNN มองว่าเป็น Fail image รูปที่ 4.14 และ 4.16 เกิดปัญหาเดียวกันกับรูปที่ 4.7 ของชุดทดสอบที่หนึ่ง คือโมเดล YOLOv5 ตรวจสอบ Filter ว่า เป็น Fail image สำหรับรูปที่ 4.15 เกิดปัญหาเดียวกันกับรูปที่ 4.10 คือตำแหน่งตรวจสอบวงแหวนของ Top clamp นั้นภาพไม่ได้อยู่ในขอบเขตการฝึกสอนโมเดล YOLOv5



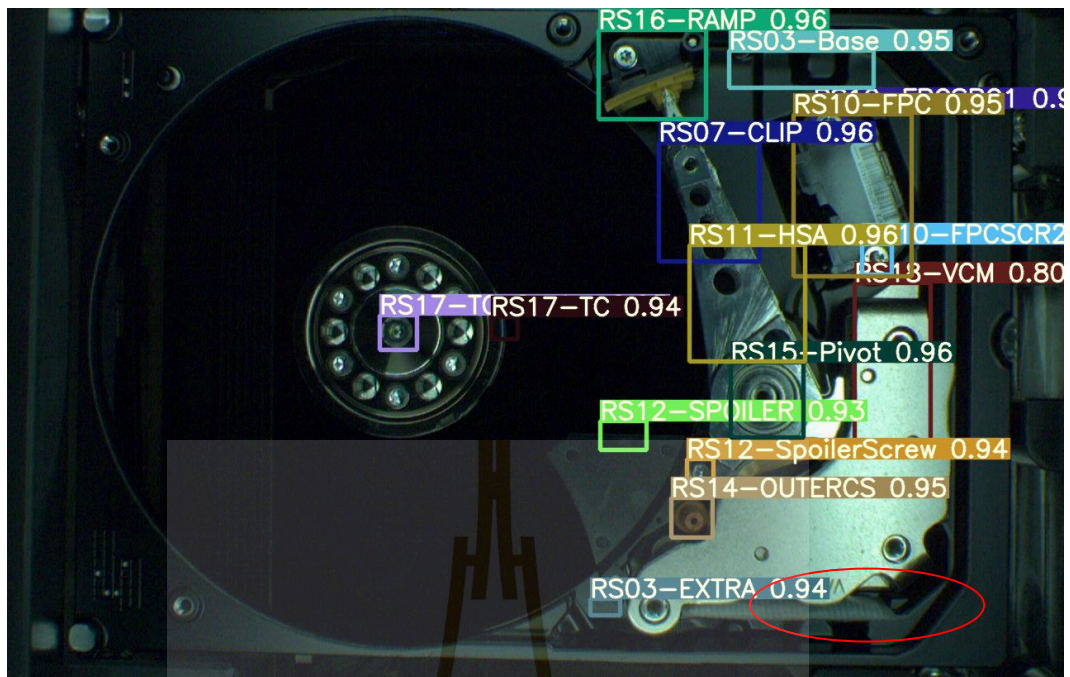
รูปที่ 4.13 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset3



รูปที่ 4.14 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset3



รูปที่ 4.15 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset3



รูปที่ 4.16 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset3

#### 4.4.4 ผลการทดสอบระบบ VI system กับข้อมูลชุดที่ 4

HDE view test set 4

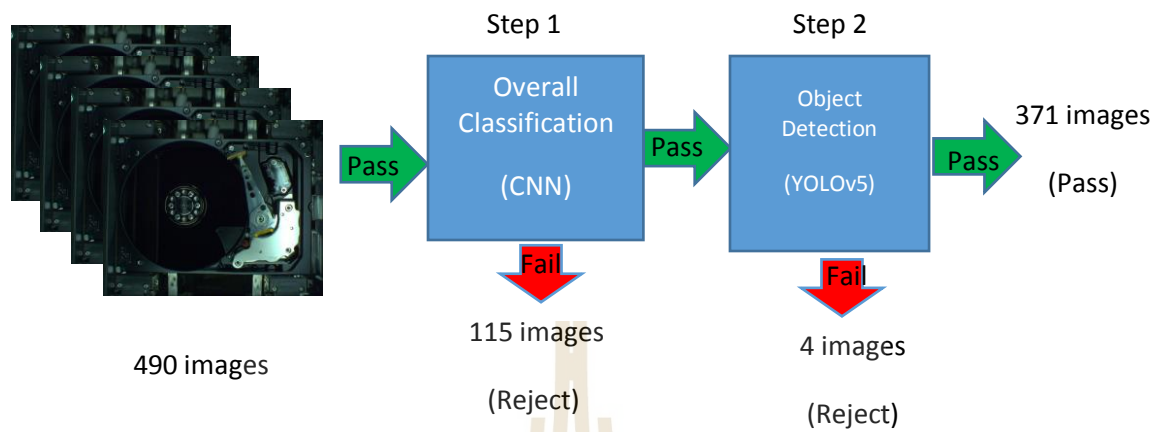
Pass images = 375 images

Fail images = 115 images

มหาวิทยาลัยเทคโนโลยีสุรนารี



## Dataset 4 test result (Set1)



รูปที่ 4.17 Flow chart แสดงการทดสอบ VI system กับ Dataset4

Pass = 371 images

Fail = 119 images

Confusion matrix result

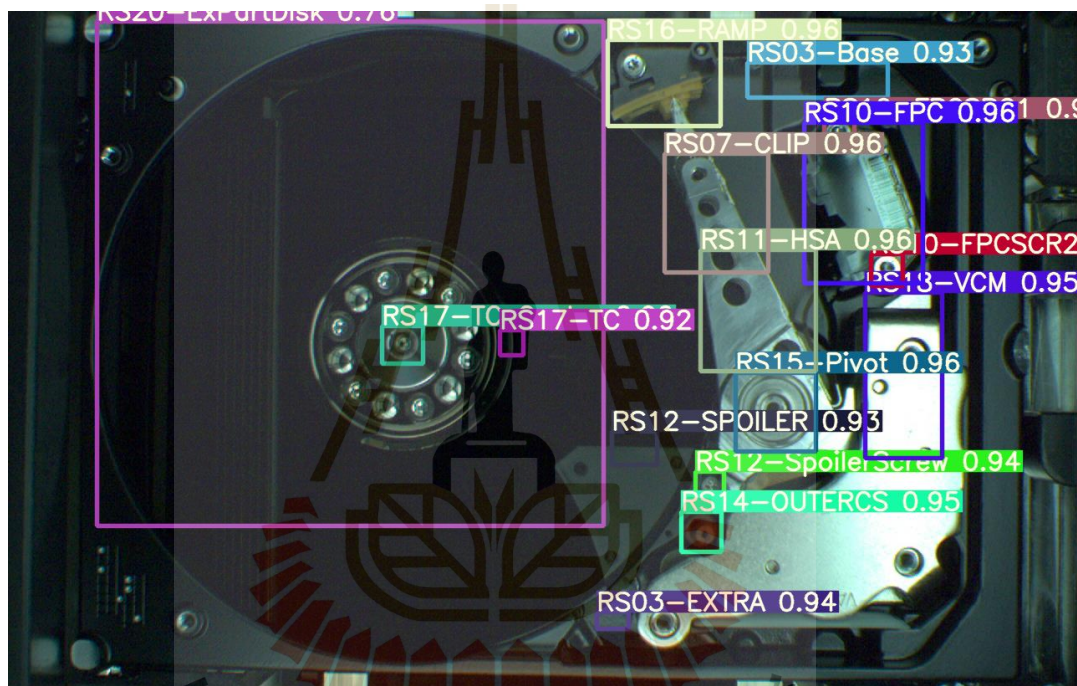
| HDE test (490 images) |                      | Predicted |       |                   |
|-----------------------|----------------------|-----------|-------|-------------------|
|                       |                      | TRUE      | FALSE | accuracy (Recall) |
| Actual                | TRUE                 | 371       | 0     | 100%              |
|                       | FALSE                | 4         | 115   | 96.639%           |
|                       | accuracy (Precision) | 99.933%   | 100%  | 99.184%           |

รูปที่ 4.18 Confusion matrix สำหรับ VI system กับ Dataset4

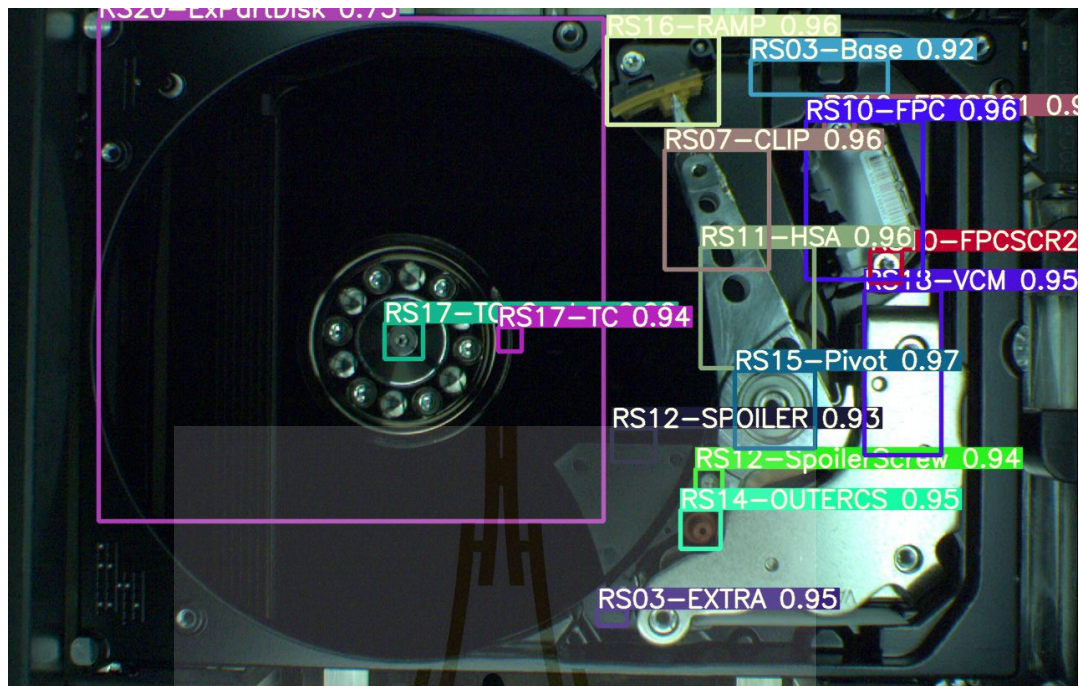
**%Accuracy of model = 99.184%**

**F1-Score = 0.9946**

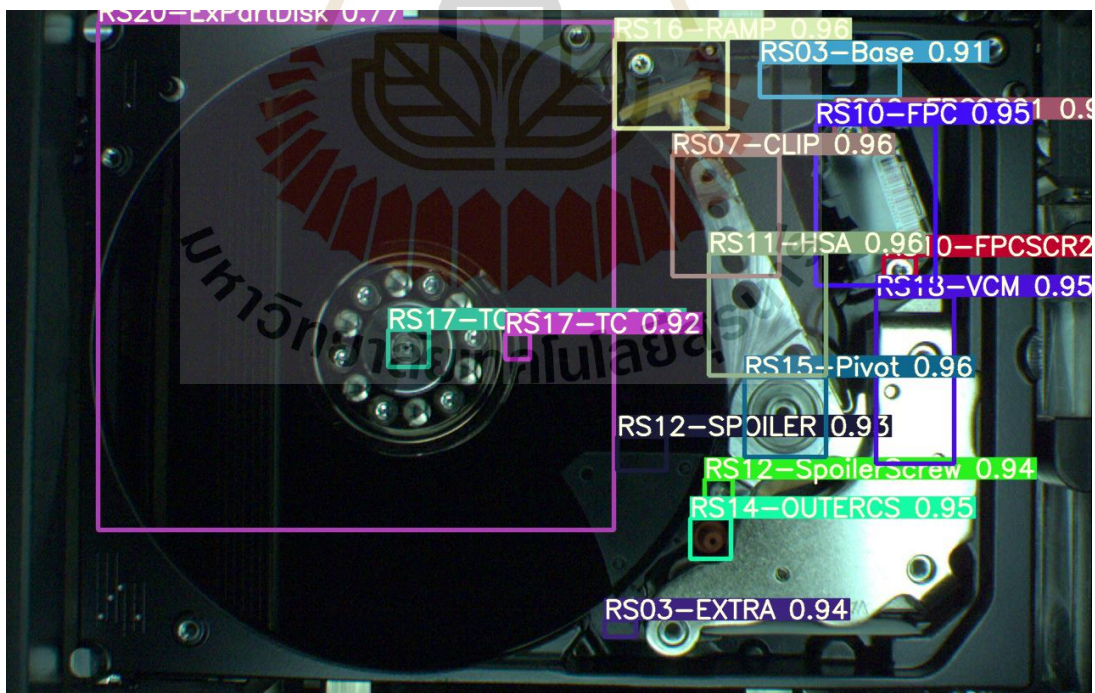
สำหรับการทดสอบระบบ VI system กับชุดข้อมูล Dataset4 นั้น ทุกรูปที่เกิดปัญหา Overkill ขึ้นที่ตำแหน่งของ Filter ทั้งหมดจากรูปอาจเป็นไปได้ว่ารูปที่จะใช้ฝึกสอนโมเดล YOLOv5 นั้นภาพ filter ไม่ได้เล็กและตรงเท่ากับภาพชุดนี้จึงทำให้ระบบมองว่าเป็น Fail image ได้โดย Dataset4 นี้เกิด Overkill images ขึ้นทั้งหมด 4 ภาพ



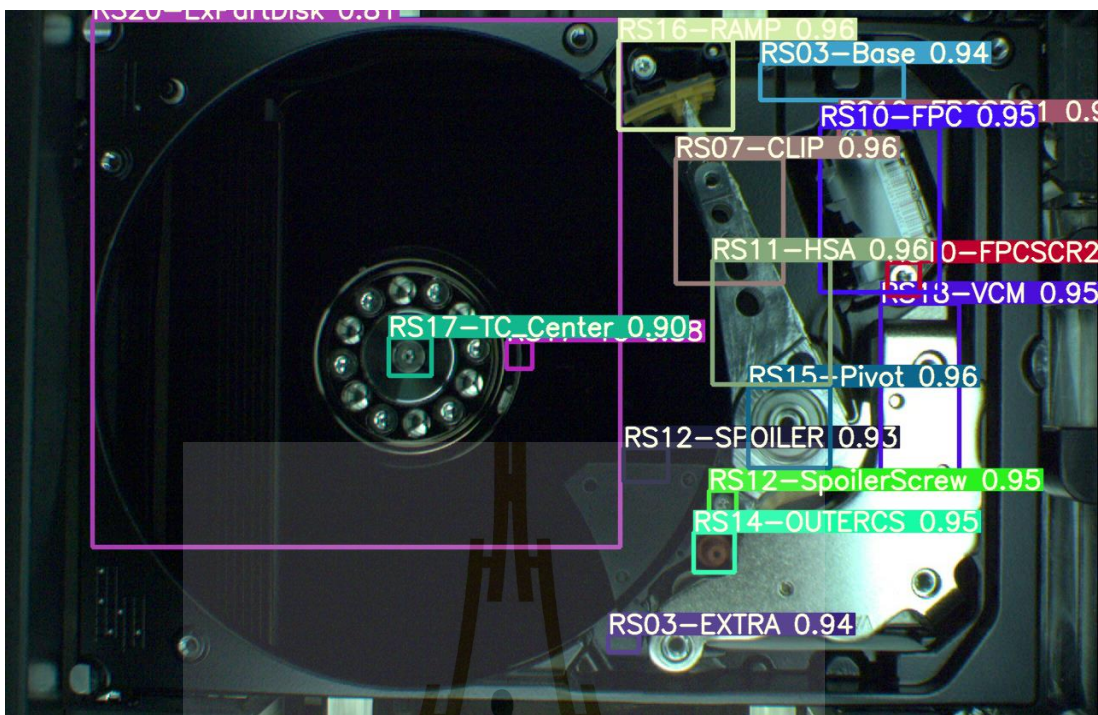
รูปที่ 4.19 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset4



รูปที่ 4.20 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset4



รูปที่ 4.21 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset4



รูปที่ 4.22 ภาพ Overkill ที่เกิดขึ้นจากการทดสอบ VI system กับ Dataset4

#### 4.5 สรุปผลการทดลอง

จากการทดลองระบบตรวจสอบองค์ประกอบฮาร์ดดิสก์ไทรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึก หรือ VI system นั้น ประสิทธิภาพในการตรวจสอบภาพ HDE view นั้นมีความแม่นยำที่สูงมากโดยการทดสอบระบบ VI system ด้วยชุดข้อมูลทั้งหมด 4 ชุดโดยที่ชุดข้อมูลทีหนึ่งระบบ VI system มีความแม่นยำในการทำนายอยู่ที่ 99.796% และ F1-score = 0.9987 โดยใช้เวลาในการตรวจสอบเฉลี่ยต่อภาพอยู่ที่ 0.039 วินาที สำหรับชุดข้อมูลที่สองระบบ VI system มีความแม่นยำในการทำนายอยู่ที่ 99.796% และ F1-score = 0.9987 โดยใช้เวลาในการตรวจสอบเฉลี่ยต่อภาพอยู่ที่ 0.039 วินาที สำหรับชุดข้อมูลที่สามระบบ VI system มีความแม่นยำในการทำนายอยู่ที่ 99.184% และ F1-score = 0.9946 โดยใช้เวลาในการตรวจสอบเฉลี่ยต่อภาพอยู่ที่ 0.039 วินาที สำหรับชุดข้อมูลที่สี่ระบบ VI system มีความแม่นยำในการทำนายอยู่ที่ 99.184% และ F1-score = 0.9946 โดยใช้เวลาในการตรวจสอบเฉลี่ยต่อภาพอยู่ที่ 0.039 วินาที ดังนั้นจากผลการทดสอบระบบ VI system ทั้งหมดจะมีความแม่นยำในการทำนายเฉลี่ยอยู่ที่ 99.49% และ F1-score เฉลี่ยอยู่ที่ 0.9967 โดยใช้เวลาเฉลี่ยในการตรวจสอบภาพ HDE view หนึ่งภาพเท่ากับ 0.0393 วินาทีดังที่แสดงในตารางที่ 4.1

ตารางที่ 4.1 ตารางสรุปผลการทดสอบระบบตรวจสอบองค์ประกอบฮาร์ดดิสก์ไทรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึก

| Visual Inspection Test Result |               |               |                           |                            |                            |           |          |
|-------------------------------|---------------|---------------|---------------------------|----------------------------|----------------------------|-----------|----------|
| No. dataset                   | True Positive | True Negative | False Positive (Overkill) | False Negative (Underkill) | Avg. time process/img. (s) | %Accuracy | F1-Score |
| 1                             | 374           | 115           | 1                         | 0                          | 0.039                      | 99.796%   | 0.9987   |
| 2                             | 374           | 115           | 1                         | 0                          | 0.039                      | 99.796%   | 0.9987   |
| 3                             | 371           | 115           | 4                         | 0                          | 0.039                      | 99.184%   | 0.9946   |
| 4                             | 371           | 115           | 4                         | 0                          | 0.040                      | 99.184%   | 0.9946   |
| Overall Score                 | 1490          | 460           | 10                        | 0                          | 0.0393                     | 99.49%    | 0.9967   |



## บทที่ 5

### สรุปและข้อเสนอแนะ

#### 5.1 สรุป

วิทยานิพนธ์เล่มนี้จัดทำขึ้นเพื่อศึกษาออกแบบและพัฒนาระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึก (Deep learning technique) สำหรับผลิตภัณฑ์ Venice โดยใช้มุมมองด้านบน (Top view) หรือเรียกอีกชื่อว่า HDE view ของผลิตภัณฑ์ Venice ของ NEO-Automation line โดยได้ประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกแบบโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network) และสองการตรวจจับวัตถุภายในภาพด้วยโวลเวอร์ชันห้า (Object detection by YOLOv5 algorithm) เพื่อตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ โดยที่ระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์นี้จะสามารถคัดแยกภาพที่ผ่าน (Pass image) และไม่ผ่าน (Fail image) ได้ซึ่งขั้นตอนการวิจัยและผลของงานวิจัยสรุปได้ดังนี้

1. เก็บข้อมูลภาพ HDE view จากเครื่อง Visual inspection ของไลน์การประกอบฮาร์ดดิสก์ไดรฟ์ (Neo-Automation line) ด้วยภาพ Venice product จะได้รูปภาพ Pass images ทั้งหมด 2500 ภาพและ Fail images ทั้งหมด 637 ภาพและทำการจัดเตรียมข้อมูลรูปภาพให้พร้อมสำหรับการนำไปใช้งานต่างๆ สำหรับงานวิจัยนี้โดยการนำภาพที่ได้ทำการปรับค่าความสว่างของภาพโดยรวมซึ่งเป็นไปตามค่ามาตรฐานของเครื่อง Visual Inspection ยอมรับได้โดยการปรับค่าแสงเพิ่มขึ้นและลดลง  $\pm 5\%$ ,  $\pm 8\%$  และ  $\pm 12\%$  ตามลำดับจะได้ภาพเพิ่มขึ้นจากต้นฉบับรวมแล้ว 7 ชุดจึงทำให้มีภาพทั้งหมด Pass images = 7000 ภาพ และ Fail images = 1239 ภาพ เนื่องจากการคัดภาพต้นฉบับออกไปเป็นจำนวน 4 ชุดโดยมี Pass images = 375 ภาพและ Fail images = 115 ภาพ ในแต่ละชุดเท่า ๆ กันเพื่อจะใช้ในการทดสอบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ต่อไป

2. ออกแบบระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึกโดยโมเดลโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network) สำหรับตรวจสอบภาพรวมของภาพ HDE view โดยจะทำการครอบภาพจากความละเอียด  $1280 \times 1024$  pixel ครอบภาพส่วนขอบบนและล่างให้เหลือขนาด  $1280 \times 814$  pixel เนื่องจากเกิดปัญหาความไม่สม่ำเสมอของสีภาพในตำแหน่งดังกล่าวจึงเกิดปัญหาค่าความแม่นยำของโมเดล CNN ไม่ได้เป็นไปตามวัตถุประสงค์ที่ต้องการจากนั้นทำการฝึกสอนและสร้างโมเดลด้วยภาพ Pass images = 7000 ภาพ และ Fail images = 1239 ภาพ โดยจะทำการแบ่งข้อมูลออกเป็น Training = 80%, Validation =

10% และ Testing = 10% โดยมีจำนวนรอบการฝึกสอนอยู่ที่ 20 epoch เนื่องจากหากมากเกินไปจะทำให้ model overfitting ขึ้น

3. ระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ในส่วนที่สองคือการตรวจสอบองค์ประกอบต่าง ๆ ของฮาร์ดดิสก์ไดรฟ์ (Object detection) โดยใช้โมเดล YOLOv5 algorithm ในการตรวจสอบองค์ประกอบต่าง ๆ ทั้งหมด 16 ตำแหน่งข้อดีของ YOLOv5 คือมีความรวดเร็วในการตรวจจับวัตถุภายในภาพและยังสามารถตรวจจับแบบ Real-time ได้อีกด้วยดังนั้นต้นแบบของ VI system จึงเลือกใช้โมเดล YOLOv5 ในการออกแบบระบบตรวจสอบองค์ประกอบตำแหน่งต่างๆ ของฮาร์ดดิสก์ไดรฟ์ใหม่ สำหรับการฝึกสอนโมเดล YOLOv5 นั้นจะเริ่มจากการนำภาพ Pass images ของ HDE view มาทำการตีกรอบตำแหน่งที่สนใจ (Labeling image) เพื่อระบุตำแหน่งองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ที่สนใจจากโดยมีภาพ Pass images = 2926 ภาพจากนั้นจะนำภาพและไฟล์ระบุ Label ของภาพไปไว้ใน Folder สำหรับการฝึกสอนโมเดลโดยจะแบ่งข้อมูลในการฝึกดังนี้ Training = 80%, Validation = 10% และ Testing = 10% โดยมีจำนวนรอบการฝึกสอนอยู่ที่ 80 epoch หากจำนวนรอบการฝึกสอนโมเดลน้อยเกินไปจะเกิดปัญหา model underfitting ได้และถ้ามากเกินไปก็เกิดปัญหา model overfitting และทำการ resize image จาก 1280x1024 เป็น 736x538 สำหรับการออกแบบระบบใหม่ทำให้ผลการฝึกสอนโมเดลมีผลที่ดีขึ้นตามวัตถุประสงค์ของงานวิจัยนี้

4. จากการทดสอบระบบ VI system ด้วยชุดข้อมูลภาพ HDE view ทั้งหมด 4 ชุดนั้นระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ด้วยเทคนิคการเรียนรู้เชิงลึกมีประสิทธิภาพในการตรวจสอบได้เป็นอย่างดีทั้งในส่วนของโมเดลโครงข่ายประสาทเทียมแบบคอนโวลูชัน และโมเดล YOLOv5 ที่ใช้สำหรับตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ทั้ง 16 ตำแหน่งโดยค่าความแม่นยำเฉลี่ยของ VI system อยู่ที่ 99.49% และมีคะแนน F1-score อยู่ที่ 0.9967 คะแนนและเวลาโดยเฉลี่ยในการตรวจสอบต่อหนึ่งภาพอยู่ที่ 0.0393 วินาที ซึ่งทั้งหมดเป็นผลลัพธ์ที่บรรลุวัตถุประสงค์ของการทำวิจัยนี้โดยผลสรุปการทดสอบระบบ VI system

## 5.2 ข้อเสนอแนะ

1. อย่างไรก็ตามงานวิจัยนี้เป็นเพียงต้นแบบของระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ เท่านั้น โดยทำการศึกษาและออกแบบเฉพาะมุมมอง HDE view เท่านั้น ยังเหลืออีก 2 มุมมองที่ต้องทำการออกแบบและศึกษาคือ Side view และ Ramp view โดยที่ระบบ VI system ที่ออกแบบมานี้ยังต้องมีการพัฒนาเพิ่มเติมและฝึกสอนโมเดลทั้งสองให้มีประสิทธิภาพมากขึ้น รongรับผลิตภัณฑ์ใหม่ ๆ เพิ่มขึ้นและยังต้องมีการตรวจสอบภาพ HDE view ที่เกิดปัญหาขึ้นเพิ่มเติมไม่ว่าจะเกิดขึ้นจากตัวระบบ VI system เองหรือเกิดจากปัจจัยภายนอก หากมีข้อมูลภาพ HDE view ชุดใด

เกิดปัญหาไม่สามารถตรวจสอบองค์ประกอบได้ก็ควรที่จะเร่งปรับปรุงระบบ VI system ทำการ Re-Train ใหม่เพื่อให้มีความสามารถครอบคลุมภาพ HDE view ในค่าสีต่าง ๆ ได้ดีที่สุดหาก

2. หากต้องการให้การทำงานของระบบ VI system ทำงานได้อย่างรวดเร็วปัจจัยหนึ่งที่ขาดไม่ได้เลยคือ คอมพิวเตอร์ที่ใช้ในการประมวลผลข้อมูล และยังสามารถลดระยะเวลาในการสร้างโมเดลได้อีกด้วย

3. ควรเก็บข้อมูลและเพิ่มชุดทดสอบให้ได้ภาพจำนวนมากกว่านี้โดยเฉพาะภาพ Fail images ซึ่งมีภาพค่อนข้างน้อย





## รายการอ้างอิง

- [1] Manisha V and Brijesh K, "A Survey Paper on Object Detection Methods in Image Processing", 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), 13-14 March 2020, Gunupur, India
- [2] Joseph R, Santosh D, Ross G and Ali F, "You Only Look Once: Unified, Real-Time Object Detection", 2016, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27-30 June 2016, Las Vegas, USA.
- [3] Joseph R and Ali F, "YOLOv3: An Incremental Improvement", 8 Apr 2018, arXiv:1804.02767
- [4] Akhil K, Arvind K, Kinshuk V, Akashdeep S and Manisha K, "Scaling up face masks detection with YOLO on a novel dataset", International Journal for Light and Electron Optics, 2021, Vol.239
- [5] Yang L, Peng S, Nickolas W and Yi S, "A survey and performance evaluation of deep learning methods for small object detection", Expert Systems With Applications, 2021, Vol.172
- [6] Alexey B, Chien-Yao W, Hong-Yuan Mark L, "YOLOv4: Optimal Speed and Accuracy of Object Detection", 23 Apr 2020, arXiv:2004.10934
- [7] Niall O' Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, Joseph Walsh, "Deep Learning vs. Traditional Computer Vision", Computer Vision Conference (CVC) 2019, 25-26 April 2019, Las Vegas, USA
- [8] Rishabh C 2020 Tele Stroke System for Stroke Detection (Cork institute of technology: Faculty of Engineering and Science) pp 18-29
- [9] Nguyen Quoc Khanh Le and Van-Nui Nguyen, "SNARE-CNN: a 2D convolutional neural network architecture to identify SNARE proteins from high-throughput sequencing data", PeerJ Computer Science, 2019

- [10] Wang, Rui, Wei Li, and Liang Zhang. "Blur image identification with ensemble convolution neural networks." *Signal Processing* 155 (2019): 73-82.
- [11] Dung, Cao Vu. "Autonomous concrete crack detection using deep fully convolutional neural network." *Automation in Construction* 99 (2019): 52-58.
- [12] Bu, Xingyuan, et al. "Deep convolutional network with locality and sparsity constraints for texture classification." *Pattern Recognition* 91 (2019): 34-46.
- [13] Ahn, Byunghyun, Jeongmin Kim, and Byeongkeun Choi. "Artificial intelligence-based machine learning considering flow and temperature of the pipeline for leak early detection using acoustic emission." *Engineering Fracture Mechanics* 210 (2019): 381-392.





ภาคผนวก ก

โค้ดโปรแกรมสำหรับการสร้างระบบตรวจสอบองค์ประกอบ

ฮาร์ดดิสก์ไดรฟ์ HDE view

มหาวิทยาลัยเทคโนโลยีสุรนารี

### ก.1 โปรแกรมประมวลผลภาพสำหรับการปรับค่าแสงโดยรวมของภาพ

```

#from PIL import ImageEnhance

def adjust_gamma(image, gamma=1):

    invGamma = 1 / gamma
    table = np.array([(i / 255.0) ** invGamma) * 255
                      for i in np.arange(0, 256)]).astype("uint8")

    return cv2.LUT(image, table)

for u in range(1,200): #จำนวนรูป
    for x in range(1,200):
        image = cv2.imread(r"C:\Users\Arl2yn\Desktop\My
        Thesis\Final_Thesis\Topview_HDD\Reject\Train_Pic_Reject\HDE_Reject
        ("+str(u)+").png")
        # image = cv2.imread(r'C:\Users\Arl2yn\Desktop\My
        Thesis\Final_Thesis\Topview_HDD\Train_Pic\HDE_(1).jpg')

        new_image = np.zeros(image.shape,image.dtype)
        #enhancer = ImageEnhance.Sharpness (image)
        #image = enhancer.enhance(1.0)
        brightness_value = +2
        contrast_value = 1

        for y in range(image.shape[0]):
            for x in range(image.shape[1]):
                for c in range(image.shape[2]):
                    new_image[y,x,c] = np.clip(contrast_value*image[y,x,c] +
                    brightness_value, 0, 255)

        #cv2.imshow('image', image)
        #cv2.imshow('edited image', new_image)5

        print('Process No.{}'.format(u))
        path = 'C:\\Users\\Arl2yn\\Desktop\\My
        Thesis\\Final_Thesis\\Topview_HDD\\Reject\\New_Pic_Reject\\+2'

        cv2.imwrite(os.path.join(path , 'HDE_Reject+2_'+str(u)+'.png'), new_image)
        #cv2.imwrite(os.path.join(path,'HDE_(1 ).jpg'), new_image)
        #cv2.waitKey()
        #cv2.destroyAllWindows()
        print('Finised')

```

### ก.2 โปรแกรมประมวลผลภาพสำหรับครอบภาพ

```

import cv2
import os
import time
import numpy as np
from PIL import Image

for u in range(1,1100):
    for x in range(1,1100):
        img = cv2.imread(r"C:\HDECNN\HDE_Crop\Pass\HDE (" +str(u)+").png")

        imgCrop = img[100:914,0:1280]

        print('Process No.{}'.format(u))
        path = 'C:\\HDECNN\\HDE_Crop\\New\\New_Pass'

        cv2.imwrite(os.path.join(path , 'HDE_C_(' +str(u)+').png'), imgCrop)

print('Finised')

```

### ก.3 โปรแกรมสำหรับฝึกสอนโมเดล CNN

```

import os
import numpy as np
from keras.models import Sequential
from keras.layers import Activation, Dropout, Flatten, Dense
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Convolution2D, MaxPooling2D, ZeroPadding2D
from keras import optimizers
import cv2

# dimensions of our images.
img_width, img_height = 1280, 814 # ขนาดของภาพแล้วแต่จะปรับเลขครับ

```

```

train_data_dir = 'train' # โฟลเดอร์สำหรับเก็บไฟล์ที่ใช้ train
validation_data_dir = 'validation' # โฟลเดอร์สำหรับเก็บไฟล์ที่ใช้ validation

# used to rescale the pixel values from [0, 255] to [0, 1] interval
datagen = ImageDataGenerator(rescale=1./255) # เราจะปรับสเกลให้แคบลงเพื่อที่จะได้เรียนรู้เร็ว
ขึ้น

# automagically retrieve images and their classes for train and validation sets
train_generator = datagen.flow_from_directory( # ตรงนี้จะสร้างอปเจ็กไว้เทรน
    train_data_dir,
    target_size=(img_width, img_height),
    batch_size=16,
    class_mode='binary')

validation_generator = datagen.flow_from_directory( # ตรงนี้จะสร้างอปเจ็กไว้ validation
    validation_data_dir,
    target_size=(img_width, img_height),
    batch_size=16,
    class_mode='binary')

# ข้างล่างลงไปนี่เป็นการสร้างโมเดลครับ ลองปรับเล่นๆดูได้
model = Sequential()
model.add(Convolution2D(32, 3, 3, input_shape=(img_width, img_height,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Convolution2D(32, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

```

```

model.add(Convolution2D(64, 3, 3))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])

nb_epoch = 20 #จำนวน epoch คือรอบที่จะทำให้ Model วิวัฒนาการ(ผมไม่รู้ว่าศัพท์ไทยเรียกว่าอะไร
              ครับ)ขึ้นไปเรื่อย ๆ
nb_train_samples = 90 #จำนวน sample ที่นำมาเทรน
nb_validation_samples = 25 #จำนวน sample ที่นำมา validate

#ตรงนี้ก็คือนั่นตอนการสอน โมเดลจริงๆครับ
model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples,
    epochs=nb_epoch,
    validation_data=validation_generator,
    validation_steps=nb_validation_samples)

#สอนเสร็จก็เซฟ

```

```
open("CNNHDEModelC_20_V2.h5", "wb")  
model.save_weights('CNNHDEModelC_20_V2.h5')  
model.summary()
```







ภาคผนวก ข

ได้ระบบตรวจสอบองค์ประกอบฮาร์ดแวร์ที่ตรงกับเทคนิคการเรียนรู้เชิงลึก

มหาวิทยาลัยเทคโนโลยีสุรนารี

### ข.1 โค้ดระบบตรวจสอบองค์ประกอบฮาร์ดแวร์ HDE view

```

import argparse
import os
import platform
import shutil
import time
from pathlib import Path
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

import cv2
import torch
import torch.backends.cudnn as cudnn
from numpy import random

import numpy as np
from keras.models import Sequential
from keras.layers import Activation, Dropout, Flatten, Dense
from keras.preprocessing.image import ImageDataGenerator, load_img,
img_to_array
from keras.layers import Convolution2D, MaxPooling2D, ZeroPadding2D
from keras import optimizers

from numba import cuda

from models.experimental import attempt_load
from utils.datasets import LoadStreams, LoadImages
from utils.general import (
    check_img_size, non_max_suppression, apply_classifier, scale_coords,
    xyxy2xywh, plot_one_box, strip_optimizer, set_logging)
from utils.torch_utils import select_device, load_classifier, time_synchronized

```

```
import easygui

config = ConfigProto()
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)

AV1=0.00000
Pa1 = 0
Fa1 = 0
Ttest1 = 0

AV2=0.00000
Pa2 = 0
Fa2 = 0
Ttest2 = 0

def CNNDetect():
    img_width, img_height = 1280, 814

    model = Sequential()
    model.add(Convolution2D(32, 3, 3, input_shape=(img_width, img_height,3)))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Convolution2D(32, 3, 3))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    model.add(Convolution2D(64, 3, 3))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])

model.load_weights('CNNHDEModelC_20_V6.h5')

dataset = LoadImages('InputImage2/', img_size=1280)

P=0
F=0
TC=0
VT=0
AVT=0

Continue ='Continue'
if os.path.exists(Continue):
    shutil.rmtree(Continue)
os.makedirs(Continue)

t0 = time.time()

for path, img, im0s, vid_cap in dataset:
```

```

imgCopy=cv2.imread(path)
imgg = load_img(path, target_size=(img_width,img_height))

out='OutputCNN2'
Continue ='Continue'
CNNFail ='CNNFail'
save_path = str(Path(out) / Path(path).name)

save_ContinuePass = str(Path(Continue) / Path(path).name)
save_CNNFail = str(Path(CNNFail) / Path(path).name)

t1 = time.time()
resized = cv2.resize(imgCopy, (img_width, img_height), interpolation =
cv2.INTER_AREA)
predictg = img_to_array(imgg)
predictiong = model.predict_classes(predictg.reshape((1,img_width,
img_height,3)),batch_size=16, verbose=0)

t2 = time.time()-t1
if predictiong[0][0] == 0:
    cv2.putText(resized, 'HDE Pass', (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.9,
(0, 255, 0), 2,cv2.LINE_4)
    cv2.imwrite(save_path, resized)
    cv2.imwrite(save_ContinuePass, imgCopy)
    print('Result = Pass (%.3fs)' % (t2))
    P+=1;

else :
    cv2.putText(resized, 'HDE Fail', (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.9,
(0, 0, 255), 2,cv2.LINE_4)

```

```

cv2.imwrite(save_path, resized)
cv2.imwrite(save_CNNFail, imgCopy)
print('Result = Fail (%.3fs)' % (t2))
F+=1

TC+=t2
VT+=1

AVT=TC/VT

print('Pass: {} and Fail: {}'.format(P,F))

return AVT, P, F ,t0

def ObjectDetect(save_img=False):
    out, source, weights, view_img, save_txt, imgsiz = \
        opt.output, opt.source, opt.weights, opt.view_img, opt.save_txt, opt.img_size
    webcam = source.isnumeric() or source.startswith('rtsp') or
    source.startswith('http') or source.endswith('.txt')

    set_logging()
    device = select_device(opt.device)
    if os.path.exists(out):
        shutil.rmtree(out)
    os.makedirs(out)
    half = device.type != 'cpu'

    model = attempt_load(weights, map_location=device)

```

```
imgsz = check_img_size(imgsz, s=model.stride.max())
if half:
    model.half()

classify = False
if classify:
    modelc = load_classifier(name='resnet101', n=2)
    modelc.load_state_dict(torch.load('weights/resnet101.pt',
map_location=device)['model'])
    modelc.to(device).eval()

vid_path, vid_writer = None, None
if webcam:
    view_img = True
    cudnn.benchmark = True
    dataset = LoadStreams(source, img_size=imgsz)
else:
    save_img = True
    dataset = LoadImages(source, img_size=imgsz)

names = model.module.names if hasattr(model, 'module') else model.names
colors = [[random.randint(0, 255) for _ in range(3)] for _ in range(len(names))]

C1=0
C3=0
C4=0
C5=0
C6=0
```

```
C7=0
C8=0
C9=0
C10=0
C11=0
C12=0
C13=0
C14=0
C15=0
C16=0
C17=0
Pass=0
Fail=0
TC=0
VT=0
AVT=0
```

```
t0 = time.time()
img = torch.zeros((1, 3, imgsz, imgsz), device=device)
_ = model(img.half() if half else img) if device.type != 'cpu' else None
for path, img, im0s, vid_cap in dataset:
    img = torch.from_numpy(img).to(device)
    img = img.half() if half else img.float()
    img /= 255.0
    if img.ndimension() == 3:
        img = img.unsqueeze(0)

    t1 = time_synchronized()
    pred = model(img, augment=opt.augment)[0]
```



```

    pred = non_max_suppression(pred, opt.conf_thres, opt.iou_thres,
classes=opt.classes, agnostic=opt.agnostic_nms)
    t2 = time_synchronized()

    if classify:
        pred = apply_classifier(pred, modelc, img, im0s)

    for i, det in enumerate(pred):
        if webcam:
            p, s, im0 = path[i], '%g: ' % i, im0s[i].copy()
        else:
            p, s, im0 = path, "", im0s

        save_path = str(Path(out) / Path(p).name)
        txt_path = str(Path(out) / Path(p).stem) + ('_%g' % dataset.frame if
dataset.mode == 'video' else "")
        s += '%gx%g ' % img.shape[2:]
        gn = torch.tensor(im0.shape)[[1, 0, 1, 0]]
        if det is not None and len(det):

            det[:, :4] = scale_coords(img.shape[2:], det[:, :4], im0.shape).round()

        for c in det[:, -1].unique():
            n = (det[:, -1] == c).sum()
            s += '%g %ss, ' % (n, names[int(c)])

```

```

for *xyxy, conf, cls in reversed(det):
    if save_txt:
        xywh = (xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-
1).tolist()

        with open(txt_path + '.txt', 'a') as f:
            f.write('%g ' * 5 + '\n') % (cls, *xywh)

    if save_img or view_img:
        label = '%s %.2f' % (names[int(cls)], conf)
        plot_one_box(xyxy, im0, label=label, color=colors[int(cls)],
line_thickness=3)

        if int(cls)==0:
            C1=1
        elif int(cls)==1:
            C3=1
        elif int(cls)==3:
            C4=1
        elif int(cls)==4:
            C5=1
        elif int(cls)==5:
            C6=1
        elif int(cls)==6:
            C7=1
        elif int(cls)==7:
            C8=1
        elif int(cls)==8:
            C9=1
        elif int(cls)==9:
            C10=1
        elif int(cls)==10:

```

```

        C11=1
    elif int(cls)==11:
        C12=1
    elif int(cls)==12:
        C13=1
    elif int(cls)==13:
        C14=1
    elif int(cls)==14:
        C15=1
    elif int(cls)==15:
        C16=1
    elif int(cls)==16:
        C17=1

if
C1==1&C3==1&C4==1&C5==1&C6==1&C7==1&C8==1&C9==1&C10==1&C11==1&C1
2==1&C13==1&C14==1&C15==1&C16==1&C17==1:
    print('%sDone. Result = Pass (%.3fs)' % (s, t2 - t1))
    Pass+=1;

else:
    print('%sDone. Result = Fail (%.3fs)' % (s, t2 - t1))
    Fail+=1;

TC+=t2 - t1
VT+=1

C1=0
C3=0
C4=0
C5=0

```

```
C6=0  
C7=0  
C8=0  
C9=0  
C10=0  
C11=0  
C12=0  
C13=0  
C14=0  
C15=0  
C16=0  
C17=0
```

```
if view_img:
```

```
    cv2.imshow(p, im0)
```

```
    if cv2.waitKey(1) == ord('q'):
```

```
        raise StopIteration
```

```
if save_img:
```

```
    if dataset.mode == 'images':
```

```
        cv2.imwrite(save_path, im0)
```

```
    else:
```

```
        if vid_path != save_path:
```

```
            vid_path = save_path
```

```
            if isinstance(vid_writer, cv2.VideoWriter):
```

```
                vid_writer.release()
```

```
            fourcc = 'mp4v'
```

```
            fps = vid_cap.get(cv2.CAP_PROP_FPS)
```

```
w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
vid_writer = cv2.VideoWriter(save_path,
cv2.VideoWriter_fourcc(*fourcc), fps, (w, h))
vid_writer.write(im0)

if save_txt or save_img:
    print('Results saved to %s' % Path(out))
    if platform.system() == 'Darwin' and not opt.update:
        os.system('open ' + save_path)

AVT=TCVT

return AVT, Pass, Fail ,t0

if __name__ == '__main__':

    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', nargs='+', type=str, default='best.pt',
help='model.pt path(s)')
    parser.add_argument('--source', type=str, default='Continue', help='source')
    parser.add_argument('--output', type=str, default='OutputObject', help='output
folder')
    parser.add_argument('--img-size', type=int, default=1280, help='inference size
(pixels)')
    parser.add_argument('--conf-thres', type=float, default=0.7, help='object
confidence threshold')
```

```

parser.add_argument('--iou-thres', type=float, default=0.5, help='IOU threshold
for NMS')
parser.add_argument('--device', default="", help='cuda device, i.e. 0 or 0,1,2,3 or
cpu')
parser.add_argument('--view-img', action='store_true', help='display results')
parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')
parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --class
0, or --class 0 2 3')
parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic
NMS')
parser.add_argument('--augment', action='store_true', help='augmented
inference')
parser.add_argument('--update', action='store_true', help='update all models')
opt = parser.parse_args()
print(opt)

with torch.no_grad():
    if opt.update:
        for opt.weights in ['yolov5s.pt', 'yolov5m.pt', 'yolov5l.pt', 'yolov5x.pt']:
            detect()
            strip_optimizer(opt.weights)
    else:
        AV1, Pa1, Fa1 ,Ttest= CNNDetect()

        device = cuda.get_current_device()
        device.reset()

        AV2, Pa2, Fa2 ,Ttest= ObjectDetect()

print("")
print("")

```

```
print('Summary of Step1 (Convolution Neural Network Process)')
print('Average Time Per Image {:.3f} s'.format(AV1))
print('Pass = {}'.format(str(Pa1)))
print('Fail = {}'.format(str(Fa1)))
print("")
print("")
print('Summary of Step2 (Object Detection Process)')
print('Average Time Per Image {:.3f} s'.format(AV2))
print('Pass = {}'.format(str(Pa2)))
print('Fail = {}'.format(str(Fa2)))
print("")
print("")
print('Summary of HDE Process')
print('All Picture = {}'.format(str(Pa2+Fa1+Fa2)))
print('Pass = {}'.format(str(Pa2)))
print('Fail = {}'.format(str(Fa1+Fa2)))
print('Average Time Process Per Image {:.3f} s'.format(AV1+AV2))
```



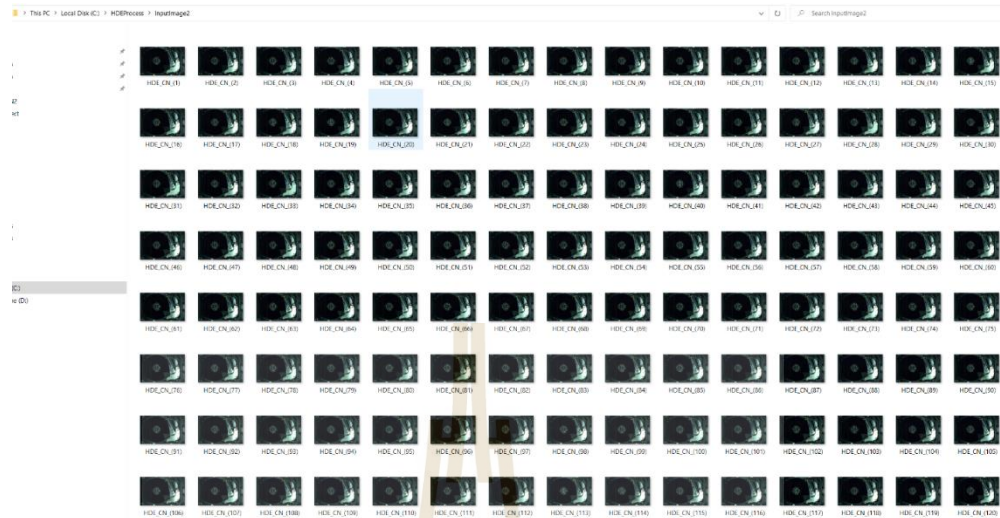
ภาคผนวก ค

การใช้ระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไดรฟ์ HDE view

มหาวิทยาลัยเทคโนโลยีสุรนารี

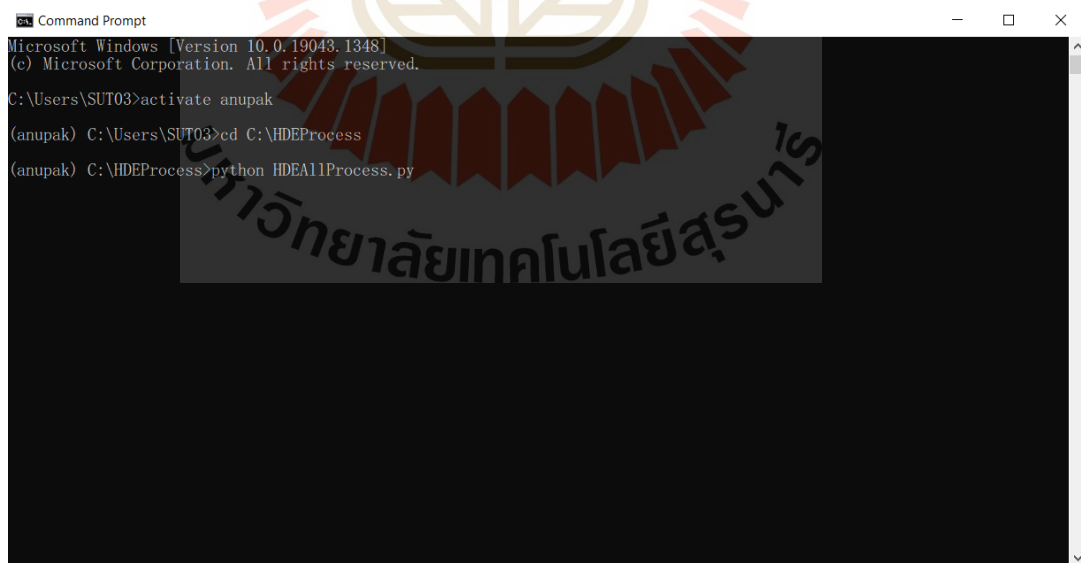


### ค.1 การใช้งานโปรแกรมตรวจสอบองค์ประกอบฮาร์ดดิสก์ไทรฟ์ HDE view



รูปที่ ค.1 ข้อมูล HDE view สำหรับทดสอบระบบ

- เตรียมข้อมูลภาพให้พร้อมเพื่อเป็น input ให้กับโปรแกรมตรวจสอบองค์ประกอบฮาร์ดดิสก์ไทรฟ์



รูปที่ ค.2 Command Prompt สำหรับ run โปรแกรม

- ใช้ Command Prompt เพื่อเตรียม environment สำหรับระบบตรวจสอบองค์ประกอบของฮาร์ดดิสก์ไทรฟ์ HDE view

```

Command Prompt - python HDEAllProcess.py
library cudnn64_8.dll
2021-11-26 14:00:12.211331: I tensorflow/core/platform/windows/subprocess.cc:308] SubProcess ended with return code: 0
2021-11-26 14:00:12.267373: I tensorflow/core/platform/windows/subprocess.cc:308] SubProcess ended with return code: 0
2021-11-26 14:00:12.316764: I tensorflow/stream_executor/cuda/cuda_blas.cc:1838] TensorFlow-32 will be used for the matrix multiplication. This will only be logged once.
Result = Pass (2.954s)
image 2/375 C:\HDEProcess\InputImage2\HDE_CN (10).png: Result = Pass (0.026s)
image 3/375 C:\HDEProcess\InputImage2\HDE_CN (100).png: Result = Pass (0.025s)
image 4/375 C:\HDEProcess\InputImage2\HDE_CN (101).png: Result = Pass (0.025s)
image 5/375 C:\HDEProcess\InputImage2\HDE_CN (102).png: Result = Pass (0.032s)
image 6/375 C:\HDEProcess\InputImage2\HDE_CN (103).png: Result = Pass (0.025s)
image 7/375 C:\HDEProcess\InputImage2\HDE_CN (104).png: Result = Pass (0.026s)
image 8/375 C:\HDEProcess\InputImage2\HDE_CN (105).png: Result = Pass (0.025s)
image 9/375 C:\HDEProcess\InputImage2\HDE_CN (106).png: Result = Pass (0.028s)
image 10/375 C:\HDEProcess\InputImage2\HDE_CN (107).png: Result = Pass (0.026s)
image 11/375 C:\HDEProcess\InputImage2\HDE_CN (108).png: Result = Pass (0.030s)
image 12/375 C:\HDEProcess\InputImage2\HDE_CN (109).png: Result = Pass (0.025s)
image 13/375 C:\HDEProcess\InputImage2\HDE_CN (11).png: Result = Pass (0.026s)
image 14/375 C:\HDEProcess\InputImage2\HDE_CN (110).png: Result = Pass (0.027s)
image 15/375 C:\HDEProcess\InputImage2\HDE_CN (111).png: Result = Pass (0.025s)
image 16/375 C:\HDEProcess\InputImage2\HDE_CN (112).png: Result = Pass (0.024s)
image 17/375 C:\HDEProcess\InputImage2\HDE_CN (113).png: Result = Pass (0.026s)
image 18/375 C:\HDEProcess\InputImage2\HDE_CN (114).png: Result = Pass (0.025s)
image 19/375 C:\HDEProcess\InputImage2\HDE_CN (115).png: Result = Pass (0.026s)
image 20/375 C:\HDEProcess\InputImage2\HDE_CN (116).png: Result = Pass (0.029s)
image 21/375 C:\HDEProcess\InputImage2\HDE_CN (117).png: Result = Pass (0.026s)
image 22/375 C:\HDEProcess\InputImage2\HDE_CN (118).png: Result = Pass (0.026s)
image 23/375 C:\HDEProcess\InputImage2\HDE_CN (119).png:

```

รูปที่ ค.3 Run โปรแกรม

- จากนั้นก็กด run โปรแกรมเพื่อทำการตรวจสอบองค์ประกอบฮาร์ดดิสก์ไทรฟ์
- 

```

Summary of Step1 (Convolution Neural Network Process)
Average Time Per Image 0.029 s
Pass = 375
Fail = 115

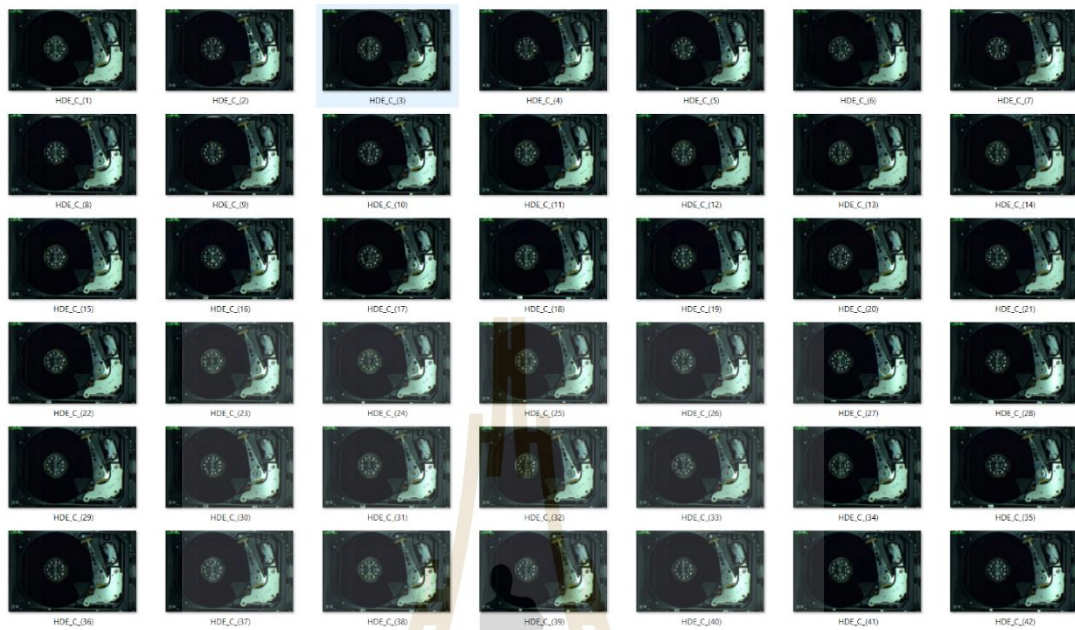
Summary of Step2 (Object Detection Process)
Average Time Per Image 0.011 s
Pass = 374
Fail = 1

Summary of HDE Process
All Picture = 490
Pass = 374
Fail = 116
Average Time Process Per Image 0.039 s

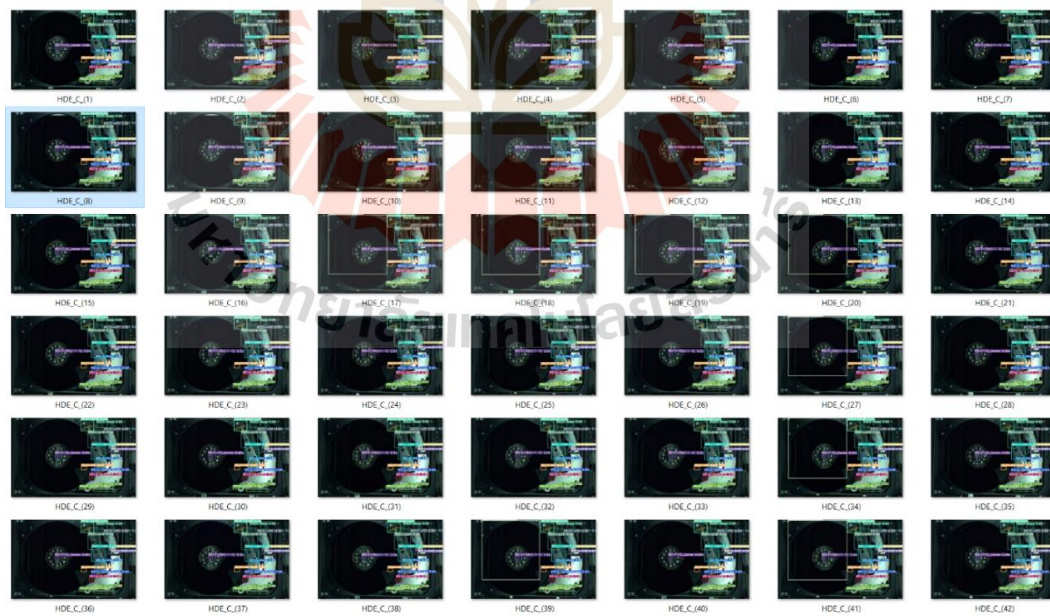
```

รูปที่ ค.4 ผลสรุปผลการทำงานระบบตรวจสอบองค์ประกอบฮาร์ดดิสก์ไทรฟ์

- ทำการตรวจสอบภาพผลลัพธ์จากระบบ



รูปที่ ค.5 รูปจากโมเดล CNN



รูปที่ ค.6 รูปจากโมเดล YOLOV5



ภาคผนวก ง

บทความที่ได้รับการตีพิมพ์เผยแพร่

## รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

International Virtual Conference on Science and Technology (SUT-IVCST 2021) hosted by Suranaree University of Technology, Thailand hold on 6 th August, 2021.



SUT International Virtual Conference on Science and Technology  
Nakhon-Ratchasima, Thailand  
6<sup>th</sup> August 2021



**AECXXX**

(this number will be assigned after full manuscript is accepted)

## Visual inspection of the Hard Disk Drive components in fully automated assembly line using by deep learning technique

A Punyathansombat<sup>1</sup>, S Pawako<sup>1</sup>, K Chamniprasart and J Srisertpol<sup>1,\*</sup>

<sup>1</sup>School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000 THAILAND

\* Corresponding Author: jiraphon@sut.ac.th

**Abstract.** This paper aims to investigate the visual inspection (VI) system in hard disk drive (HDD) components of the visual inspection using by deep learning technique. The VI program includes fault classification using convolutional neural networks (CNNs) and object detection using region based convolution neural networks (R-CNNs) with resize image. Number of training images and testing images for learning of CNNs is important to be accurate and precise. Thus, we increased number of original images as training images for learning of CNNs via created from original image by using adjusting brightness. The percent increase and decrease brightness from original images are 5%, 8% and 12%, respectively related to a range of the brightness control machine. Training images resized from 1280x1080 to 736x538. The test results of VI program using deep learning (CNNs, R-CNNs) with image processing technique can classified pass and failed by CNNs and percentage accuracy of the R-CNNs best model is 99.81%, detected region pictures with high accuracy. Classification efficiency (Pass or Fail) of the VI system is 98.22%.

**Keywords:** Image processing, Deep learning, Object detection, Hard disk drive components.

### 1. Introduction

“Data is the new oil” is a powerful word for the present and the future. Since the amount of the newly was created data in the year 2010 about 2 zettabytes due to data in 2020 about 47 zettabytes, however the predict data in year 2035 will be creasing up to 2,142 zettabytes.

Therefore, a hard disk drive is very important data storage devices in the era of big data. This paper focuses on develop the visual inspection system of the hard disk drive components which is an important step in the hard disk drive assembly (HDA) process. The interested problem with this procedure is overkill of inspection. There are 17 positions of hard disk drive components such as platters, Head Stack Assembly (HSA), RAMP, Filter, etc. that are checked. The over rejection cause is the hard disk drive image out of the image boundary perhaps mistakenly checked from the machine vision system which is the current object detection system without the flexibility of label image. This causes a delay in the HDA process. Therefore, this problem has significant implications for the amount of HDA quality.

Currently, the central Processing Units (CPUs) and graphics processing units (GPUs) are extremely efficient. Thus, object detection technology was applied by deep learning technique are the tooling of machine vision that are using widely in industry, automobiles, drones, medical, etc. The method of object detection set interest point within the picture or videos which can be defined in more than one position.

The paper on the method was used algorithms applied to machine learning models to train objective as the human brain by deep learning technique was done by Vashisht et al.<sup>[1]</sup>. The application of the one of the State-Of-The-Art objects detect (YOLO algorithm) was developed by Joseph Redmon et al.<sup>[2]</sup>. Then the algorithm models were used to quickly for detect in real-time for unified architecture.

These methods now can be use algorithms applied to machine learning models to train them to be as efficient as the human brain by deep learning technique. Joseph Redmon and Ali Farhadi<sup>[3]</sup> were development YOLOv3 a real-time object detection algorithm using logistic regression. The result revealed that an object detect score enhance extremely fast and accurate than using other deep learning method in term of specification objective images as live feed, images and videos.

The part of this article is structured as follows step. In Section 2, proposed relate works and theory for object detection. Section 3 introduces the methodology of research and design the visual inspection using by CNNs and R-CNNs. Section 4 result experiment of this paper. Finally, the conclusions and suggestions of this paper are described.

## 2. Relate works

A few years ago, object detection technology has been popular in various circles such as the industrial circle, the medical industry, the electric car industry, unmanned aerial vehicle industry, etc. The reviews on the object detection system, a large amount and YOLO algorithm will be provided as follow. Typically, the mask detection is difficult for object detection because mask sizes are different and the detection area within the image is small. Then, the YOLO algorithm is needed to detect masks by comparing between tiny original YOLOv4 and tiny YOLOv4. The results indicated that the addition of mAP value on the original YOLO v4 and tiny YOLO v4 will be 71.69 and 57.71 %, respectively, was proposed by A. Kumar et al.<sup>[4]</sup>. Yang Liu et al.<sup>[5]</sup> studied the object detection system for small point they including YOLOv3, Faster R-CNN, and SSD models for test datasets of small objects. The results were accuracy to be obtain the good model was low than 0.4, Faster R-CNNs is the best, while YOLOv3. The base of YOLOv3<sup>[3]</sup> architecture improved from YOLOv2 and now latest YOLO version is YOLOv5 improved from YOLOv4<sup>[6]</sup>.

### 2.1 Computer vision vs deep learning

For computer vision (CV) and deep learning (DL) techniques were applied on digital image correlation processes. The deep learning techniques are widely used but do not make computer vision obsolete. However, the appropriate choice is still the best answer for right now whether DL or CV Niall O' Mahony et al<sup>[7]</sup>.

What is Computer vision? Computer vision is a programing design for helping computers see the problem through the use of advanced methods and the use of general algorithms. It is used in many fields of education and industry. Computer vision objective is to distinctions analysis the content of the digital images. It is used a similar typically the development of applications that related to verify the image.

What is deep learning? Deep learning (DL) is a part of machine learning (ML) and also a little part of artificial intelligent (AI) with particularity in use which is similar to the human brain. DL there will be a connection between hidden layers to find the output layer. The workflow difference between computer vision and deep learning as show in Figure.1

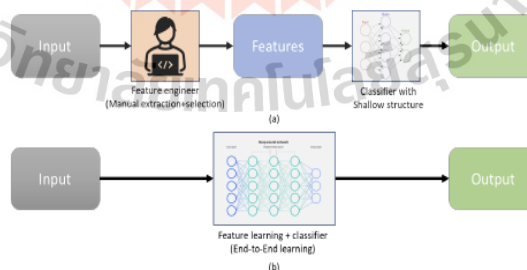


Figure 1. (a) Computer Vision workflow vs. (b) Deep Learning workflow

## 2.2 Classification Accuracy

The number to measure % accuracy of classification made by comparison between actual and predicted from a machine learning model.

### 2.2.1 Confusion Matrix

Efficiency of classification was an important methodology in term of evaluated based on the measure accuracy, precision, recall and F1-Score, this proposes to find out reasons for error which the true values are known. The proposed correction confusion matrix method is based on assessment of the visualization of classifier errors and define number of correctly classifier actual classes. The parameter is represented the number of True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP) of the classical approach of predicted of them as shown in Figure 2.

|           |          | Actual                          |                                |
|-----------|----------|---------------------------------|--------------------------------|
|           |          | Positive                        | Negative                       |
| Predicted | Positive | True positive                   | False positive<br>Type I error |
|           | Negative | False negative<br>Type II error | True negative                  |

Figure 2. Confusion matrix

$$\text{Accuracy(ACC)} = \frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}} \quad (1)$$

### 2.2.2 F1-score

F1-Score is the weighted average of two other metrics: precision and recall. Precision value (also called the positive predictive value) is represented the performance of correct positive classifications, which is defined as equations. The range for F1-Score is (0, 1)<sup>[8]</sup>.

F1-score:

$$F1 - score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2)$$

Recall is the number the classification precision is calculated relative of the amount samples size classified as adopt the actually were that class. The recall value (the sensitivity) is the proportion of samples size in order to requisite class and can more accurately represent the correctly prediction, on feature extracted by value.

Recall:

$$\text{Recall} = \frac{\sum \text{True positive}}{\sum \text{Actual positive}} \quad (3)$$

Precision (positive predictive model value) is the fraction number of relevant instances correct positive results divided by the number of positive predictions by the classifier parameters.

Precision:

$$\text{Precision} = \frac{\sum \text{True positive}}{\sum \text{Predicted positive}} \quad (4)$$



### 3. Methodology

This section describes design methodologies of visual inspection by comparing the current process of a visual inspection system using computer vision with deep learning techniques as shown in Table 1. The new visual inspection consists of convolutional neural networks (CNNs) model and Region Based Convolutional Neural Networks (R-CNNs) models. A quick glance to check for observable abnormalities is CNNs at the picture intending to notice the irregularities in the various elements is R-CNNs. The current process of visual inspection hard disk drive components doesn't have the flexibility and cannot evaluate the accuracy of the system but deep learning can evaluate model accuracy and improve efficiency of the model using more data training.

**Table 1.** A visual comparison between deep learning and computer vision

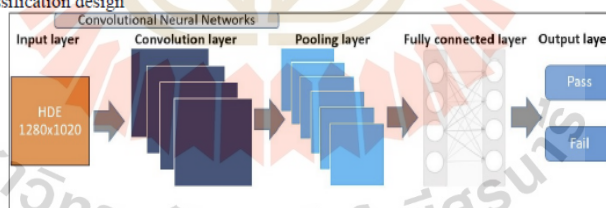
| Differences                                  | Deep learning | computer vision |
|--|---------------|-----------------|
| Manual feature extraction required?          | ✗             | ✓               |
| Training computationally resource-heavy?     | ✓             | ✗               |
| Requires labeled or classification datasets? | ✓             | ✗               |
| Black-box models?                            | ✓             | ✗               |
| Easy to deploy even on microprocessors?      | ✗             | ✓               |
| Yield high accuracy rates                    | ✓             | ✗               |

The total image for training is 595 images (Good = 418 images), (Failed = 177 images). The specification of percent increase and decrease brightness from original images are 5%, 8% and 12%, respectively related to a range of the brightness control machine. Thus, causing the creation of a new image from the original image used for training model (Data set) as shown in Table 2.

**Table 2.** The number of generation image from the brightness specification

| Picture Original | Brightness Genator |     |      |     |     |      |     | total |
|------------------|--------------------|-----|------|-----|-----|------|-----|-------|
|                  | +5%                | +8% | +12% | -5% | -8% | -12% |     |       |
| Pass             | 418                | 418 | 418  | 418 | 418 | 418  | 418 | 2,926 |
| Fail             | 177                | 177 | 177  | 177 | 177 | 177  | 177 | 1,239 |
|                  | Total              |     |      |     |     |      |     | 4,165 |

#### 3.1 CNN Classification design



**Figure 3.** Architecture of CNN Model for HDE

According to the CNN modeling techniques as shown in Fig 3.<sup>[9]</sup> was employed the training model which include total images combine good and fail images. uses total image both good and failed for train model. The specification size of image is 1280x1024 For Dividing the number of images into 80% for training, 20% for testing. The number of epoch = 50 epoch at batch size 4 past to convolution layer 4 layers and past pooling layer 1 layer using activation function is ReLU function.

### 3.2 YOLO object detection design

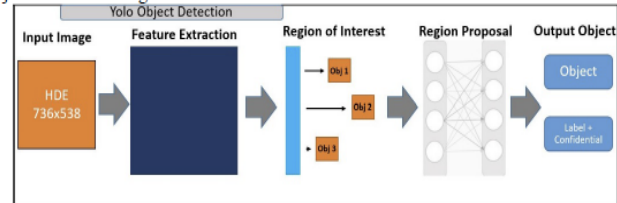


Figure 4. Architecture of R-CNN Model for HDE

YOLO modeling using only good image for train (2,926 images). The specification of image resize from 1280x1024 to 736x538 & increase/decrease of brightness 5%, 8% and 12% due to hardware requirements for dividing the number of images into 80% for training, 20% for testing. The number of epoch = 80 at batch size 4.

This part design the visual inspection system first one is CNN model initially classification result (Good or Failed) next is result from R-CNN model (YOLOv5) for object detection hard disk drive components. The operation flowchart as show in Figure 5.

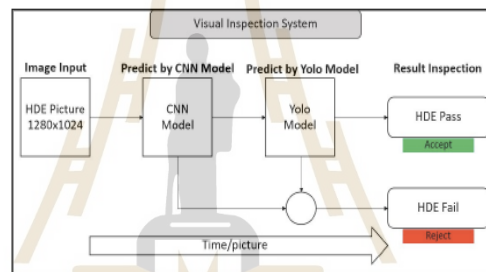


Figure 5. Flowchart of visual inspection system

### 3.3 Deep learning better than traditional computer vision?

In this scope of hard disk drive components with the visual inspection system. The mismatch is detected so that defective objects can be excluded before they reach the final packaging stage. The guidelines summarize the common attributes of each technology from the preceding discussions as show in Table 2. shows the strengths and weaknesses of both. However, deep learning can be classified components whether it be pass.

Table 2. The below table summarizes the comparison between deep learning and computer vision

| Selection criteria     | Deep learning | computer vision |
|------------------------|---------------|-----------------|
| Training dataset       | Large         | Small           |
| Computing power        | High          | Low             |
| Training time          | Long          | Short           |
| Annotation time        | Long          | Short           |
| Feature engineering    | Unnecessary   | Required        |
| Deployment flexibility | Low           | High            |
| Expenditure (BOM)      | High          | Low             |

## 4. Experimental results

The result in this section as follows: First-step of visual inspection program. By CNN model classified HDE images as pass or fail shown in Figure 6. Second-step, all pass of HDE images verifying HDD components 16 positions (11 groups) using by YOLOv5 models for the example classification as shown in Figure 7.

#### 4.1 Visual inspection results

For CNN model can separate the HDE view quite good for better images, but fail images have some errors maybe because fail images not enough for training model. However, CNN has satisfactory performance.

Object detection of VI using by YOLOv5 has the best performance at epoch of 80. The experimental results shows at the Table 3. The confidence threshold = 0.8 and dataset for test VI (good images = 296, fail images = 177, overkill images = 144). This paper focus to classify the hard disk drive components for check passed or not. The confusion matrix of VI using by YOLOv5 as shown in Figure 8. which consists of good HDE combine with Overkill HDE is 440 images and fail image is 177 images.

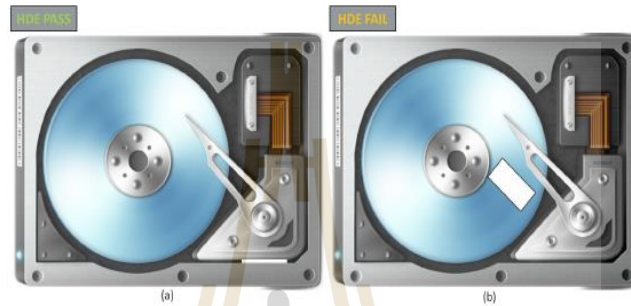


Figure 6. Sample HDE view (a) HDE view pass (b) HDE view failed classification by CNN (source : <https://www.pngegg.com/>)

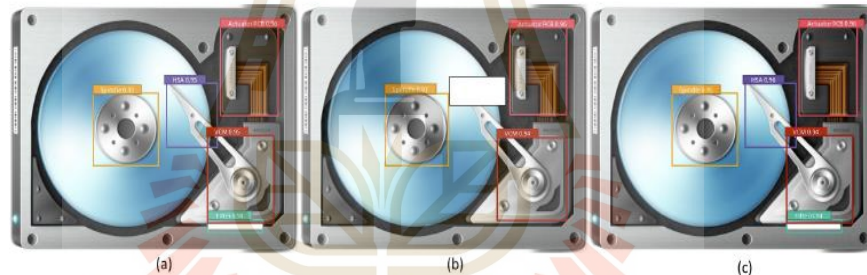


Figure 7. Sample HDE view (a) HDE view good image (b) HDE view failed image (c) HDE view overkill image classification by YOLOv5 (source : <https://www.pngegg.com/>)

Table 3. The result of YOLOv5 model training and testing

| Model          | Epoch | Batch size | Img. size | Training time | Detection time | Good (%Acc.) | Failed (%Acc.) | Overkill (%Acc.) | Overall (%Acc.) |
|----------------|-------|------------|-----------|---------------|----------------|--------------|----------------|------------------|-----------------|
| YOLOv5 model s | 20    | 4          | 736x538   | 12 mins       | 20-30 ms/img   | 95%          | 93.60%         | 95.77%           | 94.79%          |
|                | 40    | 4          | 736x538   | 24 mins       | 20-30 ms/img   | 99.65%       | 99.44%         | 99.56%           | 99.62%          |
|                | 60    | 4          | 736x538   | 36 mins       | 20-30 ms/img   | 99.88%       | 99.44%         | 99.62%           | 99.63%          |
|                | 80    | 4          | 736x538   | 48 mins       | 20-30 ms/img   | 99.90%       | 99.80%         | 99.81%           | 99.84%          |
|                | 100   | 4          | 736x538   | 1 hr.         | 20-30 ms/img   | 96.76%       | 97.94%         | 97.48%           | 97.39%          |

#### 4.2 Hardware

Hardware specification of personal desktop computer consists of CPU intel i9 gen10, GPU Nvidia RTX 3060 and RAM 32 GB DDR4 for training model. Hardware specification of personal laptop consists of CPU intel i7 gen9, GPU Nvidia RTX 2060 (laptop ver.) and RAM 16 GB DDR4 for testing model.

| Confidential Threshold 80%   | Actual HDE class    |        | User's accuracy (Precision) |
|------------------------------|---------------------|--------|-----------------------------|
|                              | Predicted HDE class | 433    |                             |
|                              | 9                   | 173    | 95.0.5%                     |
| Producer's accuracy (Recall) | 98%                 | 98.86% | 98.22%                      |

Figure 8. Confusion matrix of YOLOv5 classification (Pass or Fail)

### 5. Conclusion & Suggestion

Visual inspection systems are very important in the hard disk drive industry for control assembly quality and verifying components of a hard disk drive. The summary of visual inspection system (VI system) using CNN and R-CNN together for classification hard disk drive components initially CNN model can be used for verifying good or fail. The accuracy of VI from deep learning depends first on the amount of data set in training (Good data quality) second the algorithms must be selected suitable for the model. Finally, set up the parameter of model training such as epoch, batch, image size, for high efficiency of the model must not be overfit or underfit. The conclusion of the new visual inspection has a satisfactory efficiency of 98.22% and F1-score = 0.9875.

### Acknowledgements

This work was supported by Suranaree University of Technology and Western Digital (Thailand) Co.Ltd.

### Reference

- [1] Manisha V and Brijesh K, "A Survey Paper on Object Detection Methods in Image Processing", 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), 13-14 March 2020, Gunupur, India
- [2] Joseph R, Santosh D, Ross G and Ali F, "You Only Look Once: Unified, Real-Time Object Detection", 2016, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27-30 June 2016, Las Vegas, USA
- [3] Joseph R and Ali F, "YOLOv3: An Incremental Improvement", 8 Apr 2018, arXiv:1804.02767
- [4] Akhil K, Arvind K, Kinshuk V, Akashdeep S and Manisha K, "Scaling up face masks detection with YOLO on a novel dataset", International Journal for Light and Electron Optics, 2021, Vol.239
- [5] Yang L, Peng S, Nickolas W and Yi S, "A survey and performance evaluation of deep learning methods for small object detection", Expert Systems With Applications, 2021, Vol.172
- [6] Alexey B, Chien-Yao W, Hong-Yuan Mark L, "YOLOv4: Optimal Speed and Accuracy of Object Detection", 23 Apr 2020, arXiv:2004.10934
- [7] Niall O' Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, Joseph Walsh, "Deep Learning vs. Traditional Computer Vision", Computer Vision Conference (CVC) 2019, 25-26 April 2019, Las Vegas, USA
- [8] Rishabh C 2020 Tele Stroke System for Stroke Detection (Cork institute of technology: Faculty of Engineering and Science) pp 18-29
- [9] Nguyen Quoc Khanh Le and Van-Nui Nguyen, "SNARE-CNN: a 2D convolutional neural network architecture to identify SNARE proteins from high-throughput sequencing data", PeerJ Computer Science, 2019

## ประวัติผู้เขียน

นายอนุภาค ปัญญานสมบัติ เกิดเมื่อวันที่ 10 เมษายน 2538 ที่ตำบลกุดชุม อำเภอกุดชุม จังหวัดยโสธร เริ่มเข้าศึกษาในระดับประถมศึกษาชั้นประถมศึกษาที่ 1-6 ที่โรงเรียนบ้านซ่งแย้พิทยา และเข้าศึกษาในระดับชั้นมัธยมศึกษาชั้นปีที่ 1-6 ที่โรงเรียนกาญจนานิเชกวิทยาลัย จังหวัดกาฬสินธุ์ สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมเมคคาทรอนิกส์) สำนักศึกษาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อ พ.ศ. 2562 ได้ศึกษาต่อระดับวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเมคคาทรอนิกส์ ขณะศึกษาได้ทำงานอยู่ที่บริษัทเวสเทิร์น ดิจิตอล ประเทศไทย จำกัด ขณะศึกษาระดับปริญญาโทได้นำเสนอผลงานทางวิชาการจำนวน 1 เรื่องในการประชุมวิชาการที่ International Virtual Conference on Science and Technology (SUT-IVCST 2021) hosted by Suranaree University of Technology, Thailand hold on 6 th August, 2021.รายละเอียดมีปรากฏในภาคผนวก ง.

