

การออกแบบเครื่องตรวจความผิดปกติของการหลอมตะกั่วบนแผ่นวงจรพิมพ์ด้วย โมเดลการเรียนรู้เชิงลึก

The design of dewetting defect detection machine on printed circuit board using deep learning model

ศุวิกรม อุดลรัตนานุวัตร*

ธนภัทร มั่งกะจิตร**

บทคัดย่อ

ในอุตสาหกรรมอิเล็กทรอนิกส์ ปัญหาการตรวจชิ้นงานที่ไม่ได้คุณภาพมาจากทั้งจากคนและเครื่องจักร ถึงแม้ว่าจะมีเครื่องจักรช่วยในการตรวจชิ้นงานโดยใช้สถิติเบื้องต้นของค่าปริมาณสี แต่พนักงานต้องทำการตรวจงานซ้ำทั้งชิ้นงานเนื่องจากเครื่องจักรใช้อัลกอริทึมที่ไม่มีประสิทธิภาพ ในโรงงานกรณีศึกษาพบว่าพนักงานต้องตรวจชิ้นงานซ้ำถึงในอัตรา 10% จากยอดผลิต 2000 ชิ้นต่อวัน ทั้งนี้ปัญหานี้ซึ่งเป็นหนึ่งปัญหาทางด้านคอมพิวเตอร์วิชั่นสามารถบรรเทาเบาบางได้โดยใช้ความก้าวหน้าของเทคนิคการเรียนรู้เชิงลึก

งานสารนิพนธ์ฉบับนี้จึงได้ออกแบบเครื่องจักรที่นำมาใช้ตรวจต่อจากเครื่องจักรเดิมเพื่อแยกลักษณะงานเสียจากการหลอมเหลวที่ผิดปกติ เริ่มจากค้นหาตำแหน่งที่เสียบ่อยในชิ้นงานโดยเพิ่มความยืดหยุ่นในการตรวจในกรณีที่ชิ้นงานเอียง หลังจากนั้นจะประยุกต์ใช้อัลกอริทึมการเรียนรู้เชิงลึก (ResNet 101) ในการทำนายว่าเป็นงานเสียหรือไม่ โดยหลังจากทดลองปรับค่า Threshold สำหรับการทำนายชิ้นงานเสียเป็นมากกว่า 80% พบว่าสามารถทำนายชิ้นงานที่มาทดสอบแม่นยำมากโดยทำนายชิ้นงานเสียถูกต้องเพิ่มขึ้นเป็น 98% และทำนายชิ้นงานดีแม่นยำถึง 100% ซึ่งช่วยลดภาระของพนักงานอย่างมีนัยสำคัญในการช่วยระบุตำแหน่งที่เสียในชิ้นงาน

คำสำคัญ: คอมพิวเตอร์วิชั่น; เครื่องตรวจความผิดปกติ; การเรียนรู้เชิงลึก

*นักศึกษาหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่

**ที่ปรึกษาสารนิพนธ์

ABSTRACT

In the electronics industry, the problems in inspecting defective workpieces come from both machine and human errors. Although, the inspection machine can mitigate this problem by using basic statistical data based on colors on the workpieces. The inspectors need to examine the whole workpieces again due to the less efficient algorithms used by the machines. Regarding our reference factory, the inspection rate is about 10% of the 2,000 workpieces per day. Fortunately, this kind of computer vision problem can be alleviated using advanced deep learning techniques.

In this work, we design an additional machine to isolate the defective workpieces from abnormal melting workpieces. Firstly, our machine seeks the most frequent defective positions in the workpieces. With some degrees of rotation, these positions can still be detected by using the cumulative gradient orientation technique. Lastly, these positions would be classified as "defective" or "normal" using a ResNet101 deep learning model. By adjusting the probability threshold of predicting defective workpieces to 80%, the model's accuracy increases to 98%. Meanwhile, the model obtains 100% accuracy while classifying the normal workpieces. As a result, our machine can further significantly reduce the workload of the inspectors by precisely predicting the defective position in the workpieces.

Keywords: computer vision; defect detection machine; deep learning

บทนำ

ในอุตสาหกรรมอิเล็กทรอนิกส์พบปัญหาด้านคุณภาพการผลิตกับการตรวจชิ้นงานมาจากทั้งจากคนและเครื่องจักร ถึงแม้ว่าจะมีเครื่องจักรช่วยในการตรวจชิ้นงานแต่พนักงานต้องทำการตรวจงานซ้ำทั้งชิ้นงานเนื่องจากเครื่องจักรใช้อัลกอริทึมที่ไม่มีประสิทธิภาพ โดยใช้เพียงสถิติเบื้องต้นของค่าปริมาณสีและการกำหนดค่าขีดจำกัดในการตัดสินใจเท่านั้น ในโรงงานกรณีศึกษาพบว่า พนักงานต้องตรวจชิ้นงานซ้ำถึงในอัตรา 10% จากยอดผลิต 2000 ชิ้นต่อวัน

ด้วยเครื่องจักรที่ใช้งานยังคงใช้อัลกอริทึมการวัดเชิงปริมาณของค่า พิกเซล จึงมีแนวความคิดที่จะนำอัลกอริทึมของปัญญาประดิษฐ์มาประยุกต์ใช้งานและเปรียบเทียบผลที่เกิดขึ้นโดยปัญหาที่พบในระบบเก่าคือความซับซ้อนในการหาปัจจัยซึ่งในปัญญาประดิษฐ์ จะทำการหาปัจจัยให้อัตโนมัติและเปรียบเทียบผลหลังจากการเทรนโมเดลแล้ว โดยการศึกษาจะประยุกต์ใช้ Deep Learning โมเดล RestNet-101 (Stjepan Ložnjak, Tin Kramberger, Ivan Cesar, Renata Kramberger, 2020 : 16) นำมาใช้ในการแยกประเภทงานเสีย

ในการศึกษานี้เป็นการทำระบบเพื่อปรับปรุงเปรียบเทียบการตัดสินใจของ ชนิดงานเสียเฉพาะที่สนใจ โดยลด False call จากเดิม 10% โดยเครื่องจักรและระบบนี้จะนำไปติดตั้งต่อกับเครื่องจักรเดิมและทำการตรวจเฉพาะชนิดงานเสียให้มีประสิทธิภาพมากขึ้น และทำเครื่องหมายระบุงานเสียในตำแหน่งนั้นทั้งทางกายภาพและในระบบเพื่อทำการทวนสอบต่อไปในระบบฐานข้อมูล

ในขั้นตอนของการศึกษาความเป็นไปได้และปัญหาต่าง ๆ ในการทำระบบ เพื่อแก้ไขปัญหาดังกล่าวพบว่าการเก็บภาพที่มีความแม่นยำส่งผลต่อประสิทธิภาพการแทนและการตัดสินใจ เนื่องด้วยตำแหน่งที่ถ่ายภาพมีขนาดเล็กเมื่อเทียบกับขนาดภาพที่ต้องถ่าย และชิ้นงานมีความบาง โดยตำแหน่งสามารถขยับได้จากตำแหน่งที่ทำการตีกรอบไว้ในตอนแรกเพื่อตรวจตำแหน่งดังกล่าว ทำให้ภาพที่ได้ไม่ใช่ตำแหน่งที่ต้องการตรวจจริง ๆ จึงต้องนำวิธีการ Reference method สำหรับตรวจจับวัตถุโดยใช้วิธี Cumulative gradient orientation โดยการอ้างอิงหาตำแหน่งภาพเพื่อสามารถให้ตำแหน่งงานสามารถมีความคลาดเคลื่อนที่ยอมรับได้ และถ่ายตำแหน่งงานเสียมีความแม่นยำ

ท้ายสุดนี้การประยุกต์ใช้งาน Deep learning กับ Machine Vision สามารถลดภาระของผู้พัฒนาระบบ ซึ่งตามปกติต้องเขียนโดยใช้ อัลกอริทึมตามกฎ (Rules-Based Algorithm) ทั้งยังสามารถเพิ่มประสิทธิภาพของการตัดสินใจได้แม่นยำขึ้น และเพิ่มความหลากหลายในการประยุกต์ใช้งานระบบดังกล่าว เช่นภาคอุตสาหกรรมการผลิต ภาคการแพทย์และบริการด้านสุขภาพ ภาคประกันภัย ภาคความมั่นคงและความปลอดภัย ในการศึกษานี้เป็นส่วนหนึ่งของการนำไปประยุกต์ในภาคอุตสาหกรรมการผลิต ซึ่งผู้ศึกษาเองมีความตั้งใจที่จะนำระบบที่พัฒนาขึ้นไปประยุกต์ใช้งานที่หลากหลายขึ้นจึงออกแบบระบบให้สามารถนำกลับมาใช้ใหม่ได้โดยง่ายและลดเวลาการพัฒนา

วัตถุประสงค์

ลดการตรวจชิ้นงานเสียที่เกิดขึ้นโดยการเพิ่มความแม่นยำในการทำนายชิ้นงานเสียและระบุตำแหน่งที่เกิดขึ้น

ขอบเขตของการวิจัย

1. การศึกษานี้ มุ่งศึกษาเฉพาะสายการผลิตตัวอย่างที่พบจากโรงงานกรณีศึกษา
2. ผู้วิจัยทำการศึกษาลักษณะงานเสียที่พบได้บ่อยและต้องการปรับปรุง
3. กลุ่มตัวอย่างข้อมูลที่ใช้ในการศึกษา มาจาก 1 โมเดลของสายการผลิต โดยมีภาพงานดีและงานเสีย

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำปัญญาประดิษฐ์มาประยุกต์ใช้แก้ปัญหาที่พบเจอในเชิงอุตสาหกรรม
2. เพิ่มประสิทธิภาพของกระบวนการทำงาน โดยการลดการซ้ำซ้อนในการทำงานและใช้เครื่องจักรที่ออกแบบมาปรับปรุงในการทำงาน

แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบเครื่องจักรสำหรับการตรวจชิ้นงาน ผู้วิจัยได้ศึกษางานวิจัยที่ได้รับการตีพิมพ์และสรุปได้ดังนี้

Automatic PCB inspection เป็นบทความที่แสดงวิธีการในการตรวจชิ้นงาน โดยแบ่งเป็น 1) การตรวจโดยอ้างอิงจากภาพต้นแบบ เป็นการเปรียบเทียบระหว่างภาพต้นแบบกับภาพที่ต้องทำการตรวจ เพื่อวัดผลต่างจากภาพทั้งสอง โดยการลบค่าผลต่าง พิกเซลในตำแหน่งเดียวกันหลังจากทำ การ template matching แล้วการลบค่าผลต่างด้วย XOR operator ผลที่ได้จากวิธีดังกล่าวจะเป็นสัดส่วนของงานเสีย สิ่งสำคัญคือทั้งรูปต้นแบบและรูปอ้างอิงที่นำมาตรวจต้องมีชิ้นงานขนาดเดียวกันในรูปภาพ และตำแหน่งต้องตรงกันด้วย การค้นหาตำแหน่งของรูปอ้างอิงทำได้ด้วยการ template matching ซึ่งเป็นการวัด similarity ที่มีข้อมูลปริมาณมาก โดยวิธีการวัด Similarity 2) การตรวจโดยไม่อ้างอิงจากภาพต้นแบบ เป็นการตรวจโดยการกำหนดกฎในการตรวจสอบ design rule verification ส่วนมากใช้ในการกำหนดขนาด ค่าต่ำสุดและมากที่สุดของ ความกว้างลายวงจร ขนาดเส้นผ่านศูนย์กลางของ pad ขนาดของรูเจาะ 3) การตรวจแบบผสมผสาน เป็นการนำวิธีการตรวจโดยอ้างอิงจากภาพต้นแบบและไม่อ้างอิง เพื่อนำข้อดีของแต่ละแบบ เพื่อเพิ่มประสิทธิภาพ ของแต่ละวิธี

Object Detection and Classification of Metal Polishing Shaft Surface Defects Based on Convolutional Neural Network Deep Learning เป็นงานวิจัยที่ นำแต่ละ สถาปัตยกรรมที่มีข้อดีที่แตกต่างกันมาใช้ร่วมกัน โดยมีการใช้โมเดล Resnet101 Deep Learning สำหรับการหาปัจจัย หลังจากนั้นนำ ปัจจัยดังกล่าวมาประยุกต์ใช้กับ Faster-RCNN เพื่อทำการ Object detection โดยพบว่า Accuracy ในการ Detection 100% เมื่อเทียบกับ R-FCN และ YOLOv3 ที่ 88% และ 83% ตามลำดับ โดยประสิทธิภาพของโมเดล RestNet 101 ทำให้ผู้วิจัยสนใจในการนำโมเดลดังกล่าวมาเปรียบเทียบกับ โมเดลของผู้วิจัยเพื่อตัดสินใจในการนำโมเดลไปใช้งานต่อไป โดย RestNet คือ Deep Residual Network โดยมีการนำเสนอการแก้ปัญหาเรื่อง Vanishing Gradient ซึ่งเกิดขึ้นกรณีที่มี Layer จำนวนมาก โดยการใส่ทางลัด (Shortcut) ลงใน

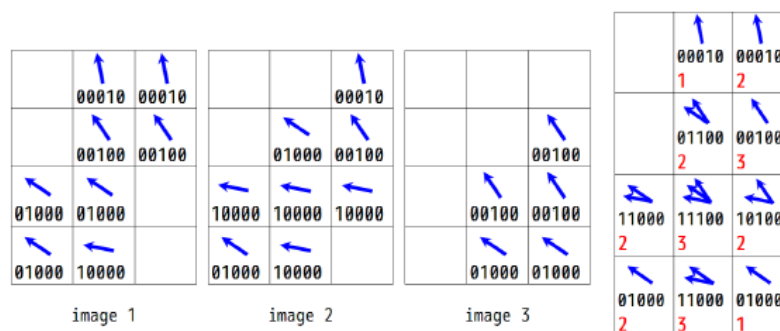
สถาปัตยกรรม เพื่อให้ผลของ input ของ layer ก่อนหน้าไม่ได้ถูกลดทอนข้อมูล จนหายไป จึงมี shortcut เพื่อนำข้อมูล Input ก่อนหน้า มารวมกับ output ที่ออกมาใน layer ถัดไป

ในสารนิพนธ์นี้ พิจารณาใช้สถาปัตยกรรม ResNet101 เพื่อนำมาแยกประเภทงานเสียเนื่องจาก ผลการ แยกประเภทเมื่อเทียบกับ โมเดล AlexNet , VGG-16, VGG-19, GoogleNet, ResNet-50, ResNet-101 พบว่ามี error ในการ Classification น้อยที่สุด ส่วนที่แตกต่างกันของ ResNet-101 เมื่อเทียบกับสถาปัตยกรรมอื่น ๆ พบว่า มีจำนวน Layer จำนวนมากกว่าโดยมี Layer อยู่ที่ 101 Layer และมี Residual Block โดย Layer ที่นำมาต้องมีขนาดเท่ากัน กรณีที่มีขนาดไม่เท่ากันต้องมีการปรับขนาดของปัจจัย Map การปรับขนาดไปมาแบบนี้ เรียกว่า Bottleneck Building Block

Textureless object detection using cumulative orientation feature เป็นงานวิจัยการจับภาพชิ้นงานด้วยกล้อง ภาพที่ได้จะมีตำแหน่งที่สนใจ ROI (Region of Interest) อยู่หลายตำแหน่ง และมีขนาดเล็กเมื่อเทียบกับขนาดภาพที่ต้องตรวจชิ้นงาน เพื่อแก้ไขปัญหาดังกล่าวจึงต้องตรวจจับชิ้นงานในตำแหน่งที่สามารถตรวจจับได้อย่างแม่นยำและรู้ตำแหน่งที่แน่นอน แล้วจึงตรวจอ้างอิงตำแหน่งนั้น ๆ มาที่ตำแหน่งที่สนใจ วิธีตรวจจับวัตถุ จะทำโดยการหา Gradient ของ Image ก่อน หลังจากหา Image gradient และ ทิศทางของ Gradient ได้แล้ว ในหนึ่งทิศทาง หลังจากนั้นทำการหมุน ทิศทาง Gradient ในที่นี้จะเรียก Template Image ตามความละเอียดที่ต้องการโดยกำหนดเป็นค่าองศา เช่น 1 องศา ก็จะต้องทำการหมุน 360 ตัวอย่าง หรือ 45 องศา จะต้องทำ 8 ตัวอย่าง เพื่อตรวจจับชิ้นงานที่หมุนที่มีการหมุนอยู่ โดยพื้นฐาน สามารถนำแต่ละ Template Image ไป สแกนหาตำแหน่งของวัตถุได้เลย โดยมีการกำหนดค่า Threshold ว่าภาพที่สแกนต้องเหมือนกับต้นแบบ (Template) เท่าไร จึงจะแสดงว่าภาพดังกล่าวคือวัตถุที่ต้องการตรวจจับ 1) ในการใช้ Template Image ในแต่ละรูปในการสแกนทั้งรูปที่ต้องการค้นหาพบจะต้องใช้เวลานานในการประมวลผล เช่น จำนวน Template Image 360 ตัวอย่าง ต้องใช้เวลาสแกน 360 ในแต่ละรูป เพื่อลดเวลาในการสแกนดังกล่าว จึงมีการสร้าง Template Image ที่เป็น Template Image ที่ผ่านการ bit OR operation (Cumulative Orientation) โดยตำแหน่งที่ สแกนไปแล้วจะไม่มีสแกนซ้ำ 2) พัฒนา Detection Rate ให้สูงขึ้นจากอัลกอริทึมแบบเดิมโดยเพิ่มและลดขนาดของ Template Image 5% จาก Master Template Image และองศา 5 ในแต่ละ Step ในการทำ Template Image +/-180 องศา พบว่า Detection Rate อยู่ที่ 85.8% เมื่อเทียบกับอัลกอริทึม Halcon , LINE2D, BOLD อยู่ที่ 45.8% , 51.0%, 85.1%



ภาพที่ 1 การสร้างรูป template ภาพต้นแบบ (ซ้าย) Gradient Image (กลาง) การสร้าง Gradient Image ในหลายทิศทาง



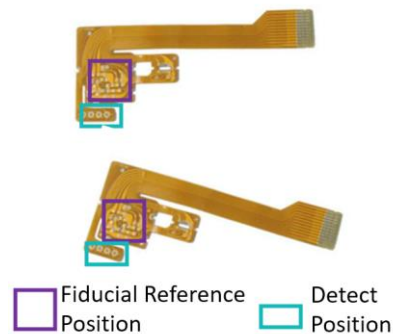
ภาพที่ 2 องค์ประกอบของ Gradient ของทั้ง 3 วัตถุ (ซ้าย) แสดงการทำ องค์ประกอบในแต่ละทิศทาง ของชิ้นงาน โดยใช้ bit OR operation โดย ตัวอักษรสีแดงคือความถี่ของแต่ละทิศทาง(ขวา)

ระเบียบวิธีวิจัย

1. การค้นหาตำแหน่งงานเสียที่พบบ่อย

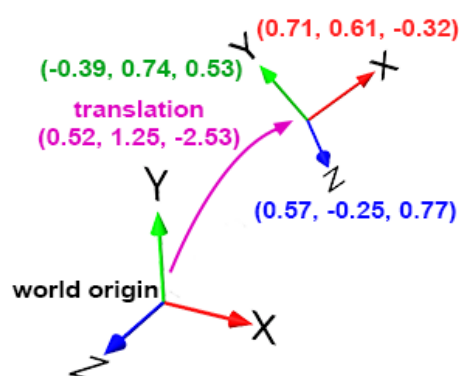
ตำแหน่งงานเสียที่พบจะอยู่ตำแหน่งเฉพาะของบนชิ้นงานในแต่ละแผ่นงาน ที่มีจำนวนงาน 20-100 ชิ้นในแต่ละแผ่นงาน และชิ้นงานดังกล่าวเป็นแผงวงจรยึดหุ่นที่มีความบางและสามารถให้ตัวได้ในแต่ละตำแหน่ง การบันทึกภาพเพื่อนำมาวิเคราะห์ จึงต้องทำการชดเชยตำแหน่งที่เกิดจากการคลาดเคลื่อนดังกล่าว การค้นหาตำแหน่งสามารถทำได้ด้วยวิธีค้นหาของ Image Gradient ของภาพต้นแบบหลังจากนั้นจึงทำการสแกนค้นหาทั่วทั้งภาพที่ต้องการตรวจ ด้วยการเปรียบเทียบค่า Similarity ของภาพต้นแบบและพื้นที่ในการค้นหา หลังจากนั้น นำค่า Similarity มาเปรียบเทียบกับค่า Threshold ที่กำหนดขอบเขตไว้เพื่อกำหนดว่าถ้าค่า Similarity มากกว่าค่าดังกล่าวแล้ว ผลลัพธ์ที่ได้จะเป็นตำแหน่งของค่าอ้างอิง ซึ่งจะได้พิกัด

มาเป็นตำแหน่ง X และ Y ของวัตถุที่ค้นพบ พบว่าตำแหน่งงานที่ห่างออกไปจากตำแหน่งอ้างอิง มีผลกระทบจากองศา ของชิ้นงานด้วย จึงจำเป็นต้องทำการหาค่าองศาของตำแหน่งอ้างอิงดังกล่าว โดยการสแกนจะกำหนดค่าความละเอียดในการสแกนหามุมดังกล่าว เช่น 0.1 , 1 , 10 องศาแล้วแต่ความต้องการ โดยค่าที่มีค่า Similarity มากที่สุดจะเป็นมุมของชิ้นงานนั้น ๆ จะได้ตำแหน่ง X Y Theta ของแต่ละชิ้นงาน



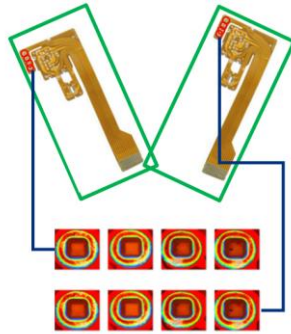
ภาพที่ 3 แสดงตำแหน่งอ้างอิงที่เกิดจากการชดเชยเพียงค่า X และ Y และความคลาดเคลื่อนจากตำแหน่งการตรวจจับงานเสีย

จะพบว่าตำแหน่งงานเสียกับตำแหน่งอ้างอิงอยู่ใน Local Coordinate จึงต้องทำการ Transform ตำแหน่งงานเสีย ให้อยู่ในพิกัด Global Coordinate โดยการนำค่า X Y Theta ที่เกิดขึ้นมาหาตำแหน่งงานเสียดังกล่าว หลังจากที่ได้ตำแหน่งงานเสียแล้วระบบจะบันทึกภาพตำแหน่งนั้น ๆ และ ผูกกับตำแหน่งตัวงานดังกล่าวเพื่อระบุว่าตัวงานชิ้นไหนที่เสียต่อไป



ภาพที่ 4 แสดง Global and Local coordinate Transformation

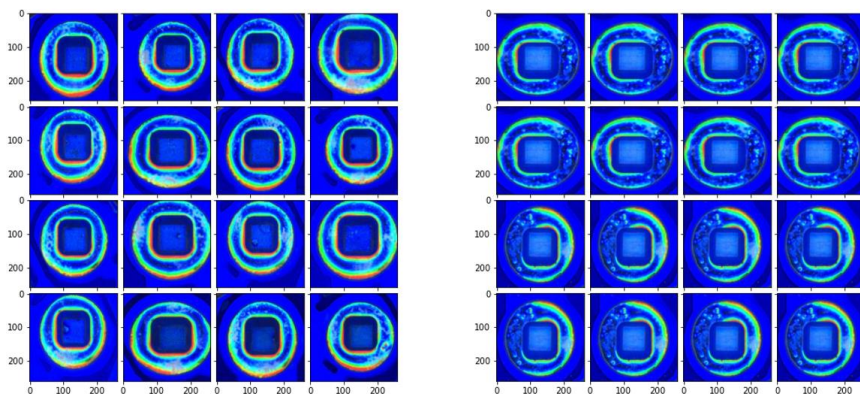
ที่มา: <https://www.scratchapixel.com/lessons/3d-basic-rendering/computing-pixel-coordinates-of-3d-point/mathematics-computing-2d-coordinates-of-3d-points>)



ภาพที่ 5 แสดงตำแหน่งงานเสียที่ตรวจพบบนแต่ละชิ้นงาน

2. ทำการแยกประเภทงานดีงานเสีย

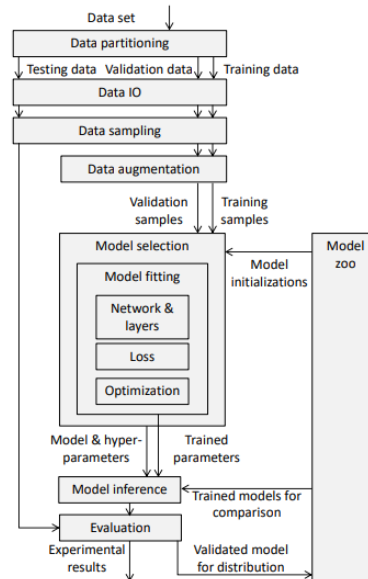
หลังจากทำการบันทึกภาพและตำแหน่งงานเสียไว้ในหน่วยความจำแล้ว จึงนำภาพตำแหน่งดังกล่าว มาแยกประเภทงานดี งานเสีย (Classification) โดยนำตัวอย่างข้อมูลมาประมวลผล ข้อมูลประกอบไปด้วย 2 กลุ่ม โดยแบ่งเป็น 1) งานดี 150 ตัวอย่าง แบ่งข้อมูลเป็น 2 ส่วน Train 100 ตัวอย่าง Test 50 ตัวอย่าง 2) งานเสีย 150 ตัวอย่าง แบ่งข้อมูลเป็น 2 ส่วน Train 100 ตัวอย่าง Test 50 ตัวอย่าง



ภาพที่ 6 ตัวอย่างงานดีด้านซ้ายมือและตัวอย่างงานเสียด้านขวามือ

หลังจากนั้นนำภาพที่ได้ไปผ่านกระบวนการในการประยุกต์ใช้ Deep Learning ซึ่งประกอบไปด้วยสามส่วน 1) การเลือกโมเดลเป็นการนำโมเดลที่เลือกมา train บน train data 2) การประเมิน โมเดลเป็นการวัดประสิทธิภาพของโมเดลด้วย test data 3) การนำโมเดลที่ได้นำไปใช้งาน โดย Data set ที่ได้มาจะถูก

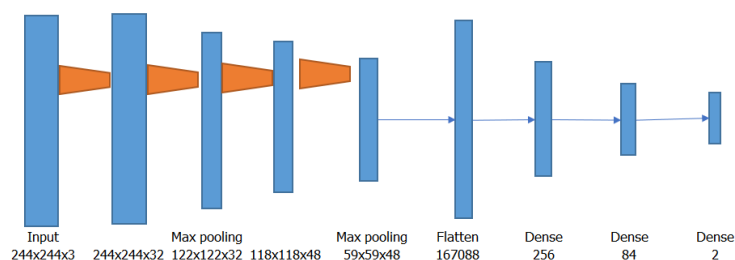
แบ่งเป็น 3 ส่วนสำหรับ Train, Validation, Test หลังจากนั้นจึงผ่านกระบวนการ Data Augmentation เป็นการแปลงข้อมูลให้มีความหลากหลายมากขึ้น ในกรณีที่นี้เป็นข้อมูลภาพจะทำการแปลงข้อมูลภาพ เช่นย่อ ขยาย หมุนซ้าย หมุนขวา กลับด้านซ้ายและขวา ปรับสี เพิ่มหรือลดข้อมูลรบกวน ทำให้โมเดล ที่เกิดจากการ train ลดปัญหา overfitting และมีความหลากหลายมากขึ้น จึงทำให้โมเดลที่เกิดขึ้น สามารถนำไปใช้โดยมีความเสถียรมากขึ้น หลังจากนั้นจึงโมเดลดังกล่าวไปพัฒนาระบบต่อไป



ภาพที่ 7 Data Flow สำหรับ Deep Learning project

โมเดลที่ใช้ในการทดสอบมี 2 โมเดล

- 1) โมเดล Customized โดยการสร้างโมเดลแสดงดังภาพ



Model: "sequential"		
Layer (type)	Output Shape	Param #
=====		
conv1d (Conv2D)	(None, 244, 244, 32)	2832
max_pooling2d (MaxPooling2D)	(None, 122, 122, 32)	0
conv2d_1 (Conv2D)	(None, 118, 118, 48)	28448
max_pooling2d_1 (MaxPooling2D)	(None, 59, 59, 48)	0
flatten (Flatten)	(None, 167988)	0
dense (Dense)	(None, 256)	42774784
dense_1 (Dense)	(None, 84)	21588
dense_2 (Dense)	(None, 2)	170
=====		
Total params: 42,837,422		
Trainable params: 42,837,422		
Non-trainable params: 0		

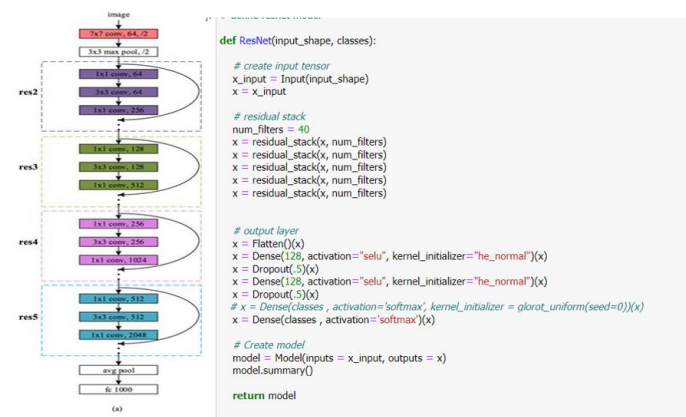
```

model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(5,5), padding='same', activation='relu', input_shape=(244, 244, 3)))
model.add(MaxPool2D(strides=2))
model.add(Conv2D(filters=48, kernel_size=(5,5), padding='valid', activation='relu'))
model.add(MaxPool2D(strides=2))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(84, activation='relu'))
model.add(Dense(2, activation='softmax'))
model.build()
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=[accuracy])
model.summary()

```

ภาพที่ 8 แสดง โมเดล Customized ที่ใช้ในการทดสอบการ Classification

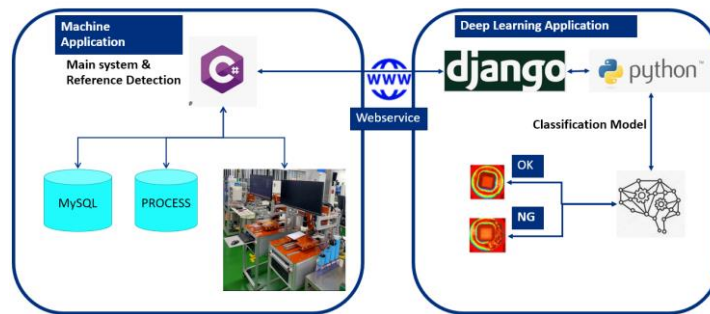
2) Resnet โมเดล โดยการสร้างโมเดลแสดงดังภาพ



ภาพที่ 9 แสดง โมเดล ResNet ที่ใช้ในการทดสอบการแยกประเภทงาน

3. การนำไปใช้งาน

ระบบที่ทำการพัฒนาประกอบไปด้วยสามส่วน หลัก ส่วนแรกเป็นการเก็บข้อมูลกระบวนการทำงานของเครื่องจักรทั้งองค์กร และการรับส่งข้อมูลกระทำผ่าน service ที่เรียกโดย dll library ส่วนที่สอง ส่วนระบบ Desktop Application ที่ใช้ในการควบคุมเครื่องจักรและกลไกต่าง ๆ การ setting parameter ของเครื่อง และอื่น ๆ ที่เกี่ยวข้องกับเครื่องจักร ส่วนที่สาม HTTP webservice ที่พัฒนาผ่าน Restful framework และติดตั้งบน Django ที่รองรับการใช้งานภาษา python ในการประมวลผลด้าน AI Deep learning โดย ระบบ Desktop application จะทำการถ่ายรูปและส่งข้อมูลให้ HTTP webservice ประมวลผลข้อมูลดังกล่าวและทำการตัดสินใจข้อมูลนั้นกลับมายัง Desktop application

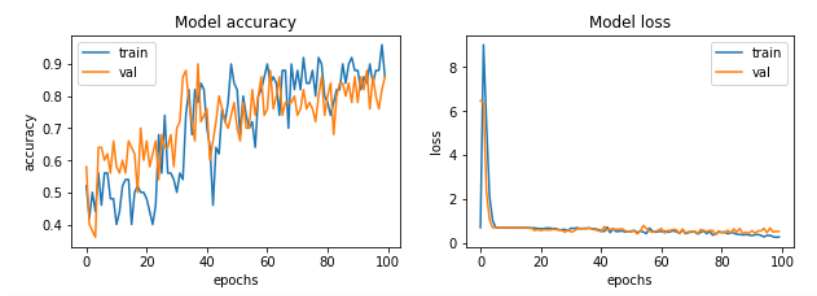


ภาพที่ 10 ระบบเครื่องจักรที่พัฒนาเพื่อตรวจชิ้นงาน

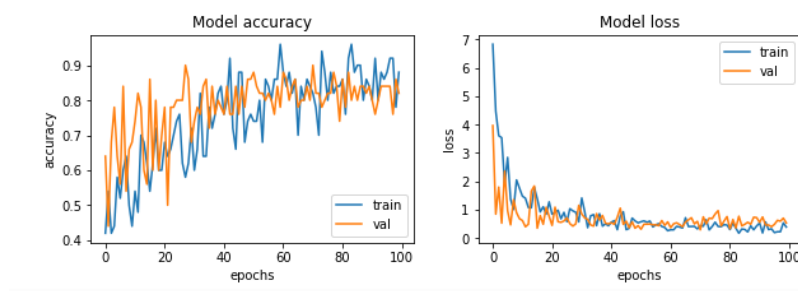
ผลการวิจัย

1. ผลการทดสอบ Validation Dataset

การสร้างโมเดลในการตรวจชิ้นงาน พบว่า โมเดล Customized validation accuracy อยู่ที่ 86% และ RestNet101 อยู่ที่ 82%



ภาพที่ 11 กราฟการเทรนข้อมูลโมเดล Customized



ภาพที่ 12 กราฟการเทรนข้อมูลโมเดล RestNet101

ตารางที่ 1 Validation accuracy

Accuracy	
Customized	RestNet 101
86%	82%

2. ผลการทดสอบ Test Dataset

พบว่าการตัดสินใจโดยใช้ Threshold > 0.5 ในส่วนของคลาสงานเสีย พบว่า โมเดล Customized accuracy อยู่ที่ 92% และ Restnet101 อยู่ที่ 97%

ตารางที่ 2 Test dataset accuracy Threshold ในการระบุงานดี $> 50\%$

Accuracy	
Customized	RestNet 101
92%	97%

3. ผลการปรับ Threshold Test Dataset

เนื่องจากผลโมเดล Test Restnet101 ได้ผลค่อนข้างดีกว่าโมเดล Customized ระบบจึงตัดสินใจใช้ Restnet101 มาใช้ในการปรับ Threshold พบว่าการตัดสินใจโดยใช้ Threshold > 0.8 ในส่วนของคลาสงานเสีย Accuracy Restnet101 อยู่ที่ 99%

ตารางที่ 3 Test dataset accuracy Threshold ในการระบุงานดี $> 50\%$

Accuracy	
Customized	RestNet 101
N/A	99%

ตารางที่ 4 แสดง Test dataset accuracy Threshold >50% และ >80%

Threshold 0.5 for judgement			Threshold 0.5 for judgement			Threshold 0.8 for judgement			Threshold 0.8 for judgement		
Class	OK		Class	DEWET		Class	OK		Class	DEWET	
Predict	Dewet	OK	Predict	Dewet	OK	Predict	Dewet	OK	Predict	Dewet	OK
1	0.30	0.70	1	0.97	0.03	1	0.30	0.70	1	0.97	0.03
2	0.25	0.75	2	0.98	0.02	2	0.25	0.75	2	0.98	0.02
3	0.14	0.86	3	0.97	0.03	3	0.14	0.86	3	0.97	0.03
4	0.29	0.71	4	0.95	0.05	4	0.29	0.71	4	0.95	0.05
5	0.75	0.25	5	0.97	0.03	5	0.75	0.25	5	0.97	0.03
6	0.11	0.89	6	0.97	0.03	6	0.11	0.89	6	0.97	0.03
7	0.50	0.50	7	0.95	0.05	7	0.50	0.50	7	0.95	0.05
8	0.16	0.84	8	0.97	0.03	8	0.16	0.84	8	0.97	0.03
9	0.09	0.91	9	0.95	0.05	9	0.09	0.91	9	0.95	0.05
10	0.25	0.75	10	0.97	0.03	10	0.25	0.75	10	0.97	0.03
11	0.44	0.56	11	0.95	0.05	11	0.44	0.56	11	0.95	0.05
12	0.16	0.84	12	0.97	0.03	12	0.16	0.84	12	0.97	0.03
13	0.39	0.61	13	0.94	0.06	13	0.39	0.61	13	0.94	0.06
14	0.06	0.94	14	0.97	0.03	14	0.06	0.94	14	0.97	0.03
15	0.05	0.95	15	0.97	0.03	15	0.05	0.95	15	0.97	0.03
16	0.14	0.86	16	0.96	0.04	16	0.14	0.86	16	0.96	0.04
17	0.14	0.86	17	0.97	0.03	17	0.14	0.86	17	0.97	0.03
18	0.07	0.93	18	0.95	0.05	18	0.07	0.93	18	0.95	0.05
19	0.10	0.90	19	0.98	0.02	19	0.10	0.90	19	0.98	0.02
20	0.16	0.84	20	0.96	0.04	20	0.16	0.84	20	0.96	0.04
21	0.29	0.71	21	0.97	0.03	21	0.29	0.71	21	0.97	0.03
22	0.22	0.78	22	0.95	0.05	22	0.22	0.78	22	0.95	0.05
23	0.62	0.38	23	0.97	0.03	23	0.62	0.38	23	0.97	0.03
24	0.36	0.64	24	0.94	0.06	24	0.36	0.64	24	0.94	0.06
25	0.27	0.73	25	0.98	0.02	25	0.27	0.73	25	0.98	0.02
26	0.20	0.80	26	0.94	0.06	26	0.20	0.80	26	0.94	0.06
27	0.07	0.93	27	0.97	0.03	27	0.07	0.93	27	0.97	0.03
28	0.20	0.80	28	0.96	0.04	28	0.20	0.80	28	0.96	0.04
29	0.69	0.31	29	0.95	0.05	29	0.69	0.31	29	0.95	0.05
30	0.09	0.91	30	0.06	0.94	30	0.09	0.91	30	0.06	0.94
31	0.09	0.91	31	0.95	0.05	31	0.09	0.91	31	0.95	0.05
32	0.35	0.65	32	0.95	0.05	32	0.35	0.65	32	0.95	0.05
33	0.17	0.83	33	0.97	0.03	33	0.17	0.83	33	0.97	0.03
34	0.28	0.72	34	0.95	0.05	34	0.28	0.72	34	0.95	0.05
35	0.61	0.39	35	0.95	0.05	35	0.61	0.39	35	0.95	0.05
36	0.10	0.90	36	0.97	0.03	36	0.10	0.90	36	0.97	0.03
37	0.39	0.61	37	0.95	0.05	37	0.39	0.61	37	0.95	0.05
38	0.10	0.90	38	0.97	0.03	38	0.10	0.90	38	0.97	0.03
39	0.14	0.86	39	0.95	0.05	39	0.14	0.86	39	0.95	0.05
40	0.15	0.85	40	0.97	0.03	40	0.15	0.85	40	0.97	0.03
41	0.68	0.32	41	0.95	0.05	41	0.68	0.32	41	0.95	0.05
42	0.30	0.70	42	0.97	0.03	42	0.30	0.70	42	0.97	0.03
43	0.20	0.80	43	0.95	0.05	43	0.20	0.80	43	0.95	0.05
44	0.16	0.84	44	0.97	0.03	44	0.16	0.84	44	0.97	0.03
45	0.08	0.92	45	0.95	0.05	45	0.08	0.92	45	0.95	0.05
46	0.27	0.73	46	0.97	0.03	46	0.27	0.73	46	0.97	0.03
47	0.34	0.66	47	0.97	0.03	47	0.34	0.66	47	0.97	0.03
48	0.44	0.56	48	0.95	0.05	48	0.44	0.56	48	0.95	0.05
49	0.05	0.95	49	0.97	0.03	49	0.05	0.95	49	0.97	0.03
50	0.07	0.93	50	0.95	0.05	50	0.07	0.93	50	0.95	0.05
MAX	0.75	0.25	MAX	0.98	0.02	MAX	0.75	0.25	MAX	0.98	0.02
MIN	0.05	0.95	MIN	0.06	0.94	MIN	0.05	0.95	MIN	0.06	0.94
ACCURACY	88%		ACCURACY	98%		ACCURACY	100%		ACCURACY	98%	

สรุปและอภิปรายผลการวิจัย

การสร้างโมเดลด้วยการเรียนรู้เชิงลึก ในโมเดล Customized มี accuracy ที่ 92% ส่วน โมเดล RestNet101 ที่ 97% โดยมีค่า Threshold ที่ใช้ในการตัดสินใจในของคลาสงานเสียอยู่ที่ >50% และหลังจากการเปลี่ยนค่า Threshold และเพิ่มกฎโดยให้การตัดสินใจในของคลาสงานเสียอยู่ที่ > 80% พบว่า RestNet101 มี accuracy ที่ 99% ซึ่งจากผลการศึกษาระบบที่พัฒนาสามารถลดการตัดสินใจคลาดเคลื่อนได้จาก 10% เหลือเพียง 1% เมื่อมีการประยุกต์ใช้เครื่องจักรและระบบที่ออกแบบในการศึกษานี้ ร่วมกับเครื่องจักรตรวจชิ้นงานที่ใช้อยู่ในโรงงานกรณีศึกษา

แนวทางการพัฒนาในอนาคต

แนวทางที่ผู้วิจัยจะนำไปพัฒนาต่อมีดังนี้ 1) ระบบที่พัฒนาควรจะผนวกขั้นตอนการเทรนเข้าไปในระบบด้วย เพื่อรองรับงานหลายๆ โมเดล โดยไม่ต้องไปรัน Python script โดย User ไม่ต้องมี Experience ในการ Programming 2) เพื่อเพิ่มความยืดหยุ่นระบบควรจะสามารถเพิ่ม คลาสในการตัดสินใจเพิ่มเติมตามจำนวนที่ระบุ กรณีที่นำเครื่องไปใช้กับสายการผลิตอื่น ๆ 3) นำระบบที่พัฒนาในส่วนการตรวจจับวัตถุ ไประบุตำแหน่งชิ้นงาน ที่ต้องการให้หุ่นยนต์จับโดยใช้ระบบวิชั่น 4) นำระบบที่พัฒนาไปประยุกต์ใช้งานที่หลากหลายมากขึ้น

บรรณานุกรม

Madhav Moganti, Fikret Ercal, Cihan H. Dagli, Shou Tsunekawa : “Automatic PCB Inspection Algorithms”,1996

Qingsheng Jiang , Dapeng Tan , Yanbiao Li, Shiming Ji *, Chaopeng Cai and Qiming Zheng : “Object Detection and Classification of Metal Polishing Shaft Surface Defects Based on Convolutional Neural Network Deep Learning” applied sciences Article, 20 Dec 2019

Stjepan Ložnjak, Tin Kramberger, Ivan Cesar, Renata Kramberger : “Automobile Classification Using Transfer Learning on Resnet Nueral Network Architecture,” Polytechnic & Design ,2020,pp58-64

Yoshinori Konishi, Yoshihisa Ijiri, Masaki Suwa, Masato Kawade: “Textureless object detection using cumulative orientation feature,” ResearchGate 2017