

บทที่ 1 บทนำ

1.1 ความสำคัญและที่มาของงานวิจัย

ปัจจุบัน นวัตกรรม เทคโนโลยีและความคิดสร้างสรรค์ได้เข้ามามีบทบาทอย่างมากในสังคมไทย เนื่องจากปัจจุบันประเทศไทยมีการปฏิวัติอุตสาหกรรม ครั้งที่ 4 เรียกอีกอย่างว่า ยุคไทย 4.0 ทำให้อุตสาหกรรมด้านต่างๆขยายตัวเติบโตเพิ่มมากขึ้น และหนึ่งในนั้นคือ อุตสาหกรรมหุ่นยนต์ อุตสาหกรรมหุ่นยนต์เป็นอุตสาหกรรมพื้นฐานสำคัญที่ช่วยให้อุตสาหกรรมด้านต่างๆพัฒนาได้อย่างรวดเร็ว ดังนั้นในแวดวงอุตสาหกรรมจึงเห็นค่าความสำคัญของอุตสาหกรรมหุ่นยนต์และนำไปประยุกต์ และปรับใช้กับอุตสาหกรรมต่างๆมากขึ้น เช่น อุตสาหกรรมอิเล็กทรอนิกส์ อุตสาหกรรมเกษตร อุตสาหกรรมยานยนต์ เป็นต้นทำให้กระบวนการทำงานรวดเร็วและมีประสิทธิภาพเพิ่มมากขึ้น และยังสามารถลดจำนวนแรงงานคนในบริษัท อีกทั้งสามารถต่อ ยอดการทำงานได้เพิ่มมากขึ้น ดังนั้นบริษัทต่างๆจึงใช้หุ่นยนต์เข้ามามีส่วนร่วมและทดแทนการทำงานของแรงงานคน เพื่อช่วยลดค่าใช้จ่าย และเพิ่มการทำงานให้มีประสิทธิภาพมากยิ่งขึ้น

บริษัทไทยออปติคอลล กรุ๊ป จำกัด (มหาชน) Thai Optical Group หรือ TOG เป็นบริษัทที่ผลิตและจัดจำหน่ายเลนส์แว่นตา และทำการส่งออกทั้งภายนอกและภายในประเทศ ทางบริษัทได้เล็งเห็นถึงความสำคัญของการใช้หุ่นยนต์ในส่วนของการผลิต เนื่องจากในบริษัทยังมีการใช้แรงงานมากถึง 80 % จึงทำให้การทำงานยังล่าช้ามีความผิดพลาดเกิดขึ้น ดังนั้นทางบริษัทจึงมีความต้องการนำหุ่นยนต์อัตโนมัติมาใช้ในการกระบวนการทำงานมากขึ้น ในกระบวนการต่างๆเช่น การผลิต จัดเก็บและประกอบปะเก็น (Gasket) การฉีดสารโมโนเมอร์ที่ใช้ทำเลนส์เข้าโมล การฝนริมขอบเลนส์ การบรรจุเลนส์เข้าซองใส่เลนส์ การตรวจสอบคุณภาพเลนส์ เป็นต้น ในส่วนของการตรวจสอบคุณภาพเลนส์ ทางบริษัทยังคงใช้แรงงานเป็นคนตรวจสอบ ทำให้ยังเกิดข้อผิดพลาดในการทำงานสูง ทางบริษัทจึงมีความต้องการที่จะใช้หุ่นยนต์อัตโนมัติในการคัดแยกเลนส์ เพื่อความแม่นยำในการคัดแยกและยังสามารถลดจำนวนคนงานได้อีกด้วย ทางบริษัทเล็งเห็นถึงผลกระทบที่จะเกิดขึ้น ถ้าหากลูกค้าได้รับเลนส์ที่มีคุณภาพไม่ได้มาตรฐาน ดังนั้นการนำระบบการเรียนรู้ด้วยเครื่องมาใช้ในการตรวจสอบคุณภาพของเลนส์ จึงเป็นหนึ่งในวิธีที่สามารถแก้ไขปัญหาข้างต้นได้

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อออกแบบและพัฒนากระบวนการตรวจสอบเลนส์ สำหรับการคัดแยกเลนส์ที่ไม่ได้คุณภาพ
- 2) เพื่อศึกษาการตรวจสอบคุณภาพเลนส์ โดยใช้การเรียนรู้ของเครื่อง (Machine Learning)

- 3) เพื่อเปรียบเทียบตัวแปรที่มีผลต่อปัจจัยในการแยกเลนส์ที่ไม่ได้คุณภาพ

1.3 ประโยชน์และผลคาดว่าจะได้รับของงานวิจัย

- 1) สามารถคัดแยกเลนส์ที่ไม่ได้คุณภาพ
- 2) สามารถลดจำนวนพนักงานในการคัดแยกเลนส์ และลดค่าใช้จ่าย
- 3) สามารถเพิ่มประสิทธิภาพของการคัดแยกเลนส์ที่ไม่ได้คุณภาพ
- 4) สามารถนำความรู้ที่เรียนมาประยุกต์ใช้ในการพัฒนาสายการผลิตในระบบอุตสาหกรรมให้เป็นระบบอัตโนมัติ

1.4 ขอบเขตงานวิจัย

- 1) โปรแกรมที่พัฒนาใช้สำหรับตรวจสอบคุณภาพเลนส์ที่ไม่ได้คุณภาพ 2 แบบ คือ แบบ 40 และ แบบ 41
- 2) ตรวจสอบเลนส์ แบบ SEMI และ FINISH
- 3) ระบบที่ออกแบบสามารถนำไปใช้ได้จริง จากการนำชิ้นงานระบบของพนักงาน และสามารถแสดงผลว่าเป็นเลนส์คุณภาพ หรือ ไม่ได้คุณภาพ

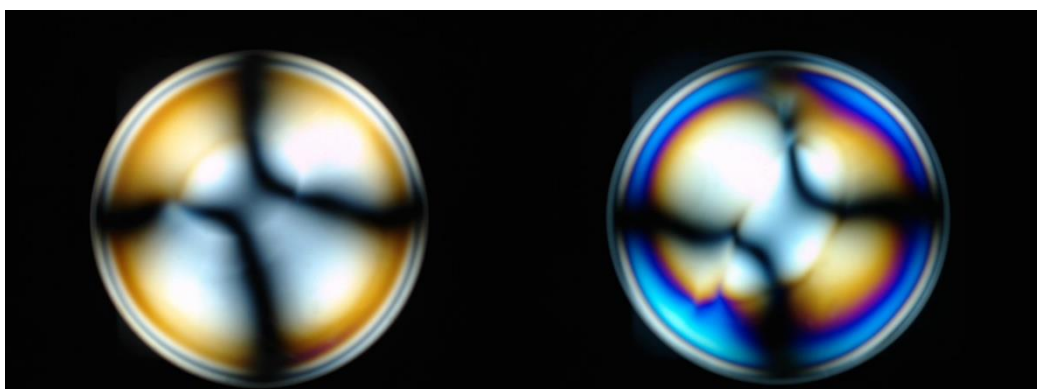
1.5 ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ระยะเวลาการดำเนินงาน (เดือน) พ.ศ.2561			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.
1. การสำรวจผลงานวิจัยและทฤษฎีที่เกี่ยวข้อง				
2. เก็บข้อมูลของเลนส์ที่ได้คุณภาพและไม่ได้คุณภาพ				
3. เขียนโปรแกรมตรวจสอบคุณภาพเลนส์				
4. ทดลอง ปรับปรุงระบบและวิเคราะห์ผลการทดสอบ				
5. สรุปผลวิจัย				
6. จัดทำวิทยานิพนธ์และนำเสนอผลงาน				

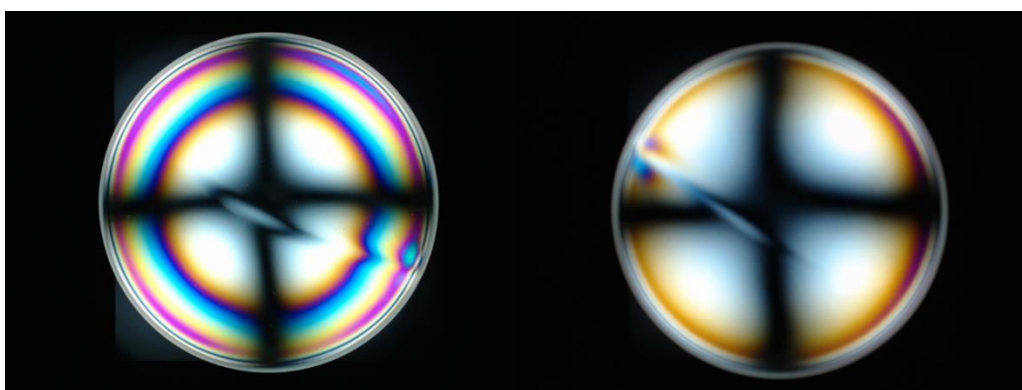
บทที่ 2 ทฤษฎี/งานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

การตรวจคุณภาพเลนส์ด้วยวิธีการโพลาไรซ์เซชัน โดยจะมีความแม่นยำและถูกต้องสูง จึงนำการเรียนรู้ด้วยเครื่อง (Machine Learning) มาใช้ช่วยในการตัดสินใจ เพื่อช่วยเพิ่มประสิทธิภาพในการตรวจคุณภาพของเลนส์ โดย เลนส์ที่ไม่ได้คุณภาพ จะมีอยู่ 2 ประเภท ด้วยกัน คือเลนส์ ที่ไม่ได้คุณภาพแบบมีวงกลมในส่วนของเนื้อเลนส์หรือเรียกว่า 40 และ เลนส์ที่ไม่ได้คุณภาพแบบมีเส้นในส่วนของเนื้อเลนส์ หรือเรียกว่า 41



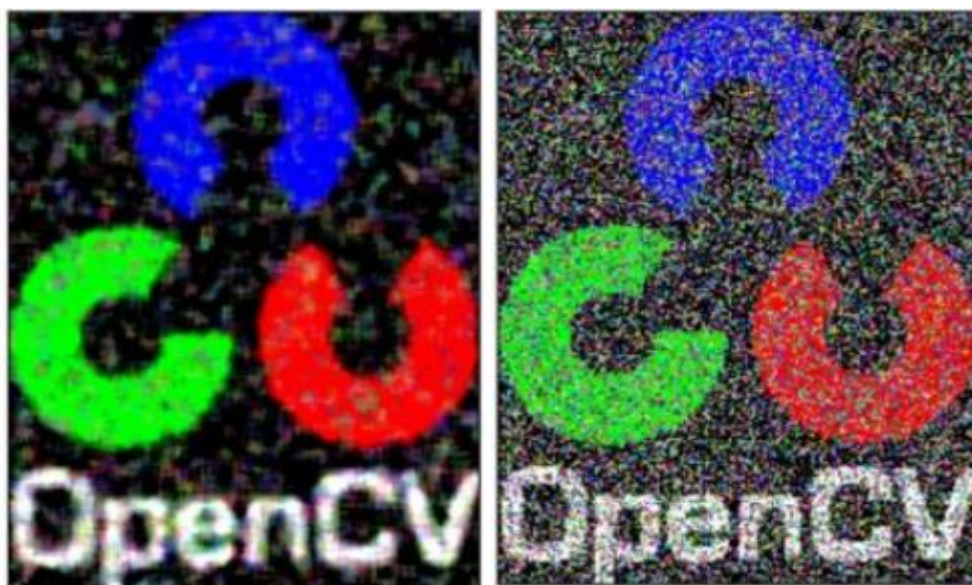
รูปที่ 2.1 ตัวอย่างเลนส์ที่ไม่ได้คุณภาพแบบวงกลม



รูปที่ 2.2 ตัวอย่างเลนส์ที่ไม่ได้คุณภาพแบบมีเส้น

2.1.1 OpenCV

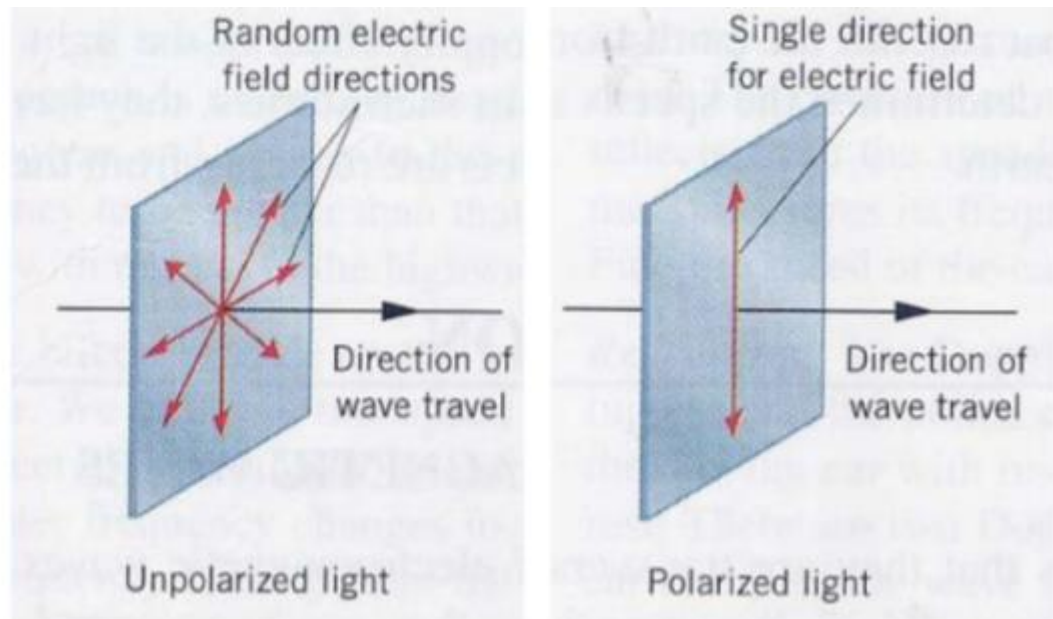
OpenCV (Open Source Computer Vision) เป็นไลบรารีสำหรับรวบรวมคำสั่งหรือฟังก์ชันสำหรับการประมวลผลภาพ โดยมีการพัฒนาโปรแกรมบนภาษาที่หลากหลาย เช่น C, C++, Python เป็นต้น โดยเราจะสามารถเรียกใช้ ฟังก์ชันเหล่านี้ได้ เช่น การลด Noise ของภาพลง



รูปที่ 2.3 ภาพตัวอย่างก่อนและหลังทำการประมวลผลภาพของ OpenCV [6]

2.1.2 โพลาริเซชัน (Polarization)

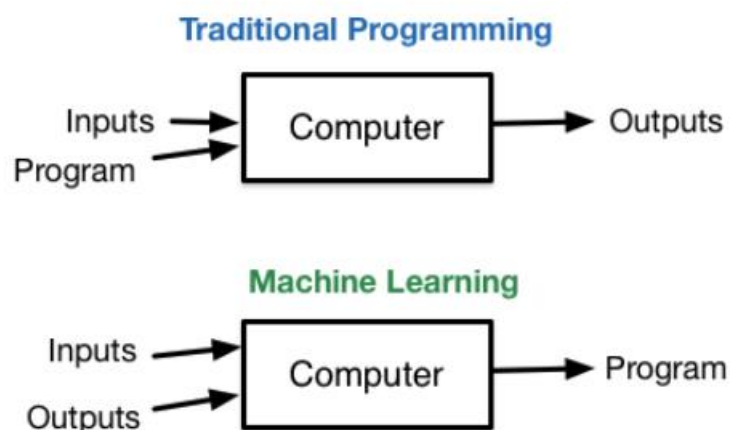
โพลาริเซชัน คือการที่คลื่นนั้นมีระนาบการสั่นของอนุภาค หรือ สนามแม่เหล็ก หรือ สนามไฟฟ้ามีระนาบเดียว โดยทั่วไปแสงจะกระจายไปหลากหลายทิศทางแต่หาก ผ่านการโพลาริเซชันแล้วแสงจะผ่านไปเพียงแกนเดียว โดยใช้โพลาริเซชันเป็นประเภทดูดกลืน โดยผ่านแผ่นโพลาริซ์แสงที่ผ่านจะมีระนาบตรงกับแผ่นของโพลาริซ์เพียงระนาบเดียว



รูปที่ 2.4 แสงที่ผ่านแผ่นโพลาไรซ์เพียงระนาบเดียว [7]

2.1.3 การเรียนรู้ด้วยเครื่อง (Machine Learning)

Machine Learning นั้นแตกต่างจาก การเขียนโปรแกรมแบบเดิม (Traditional Programming) โดยที่ ปกติแล้วโปรแกรมที่เขียนจะต้องเขียนสูตรในการหา Output เอง เช่น $Y = X + 2$ ส่วนที่จะต้องเขียนโปรแกรมลงไปคือ กำหนด $X + 2$ โปรแกรมจะสามารถหาค่า Y มาให้ แต่ในส่วนของ Machine Learning คือการนำ X และ Y ที่มีจำนวนเยอะๆ เข้าไปใส่ในโปรแกรม Machine Learning จะสามารถทำนายสมการออกมาให้ คือ $Y = X + 2$



รูปที่ 2.5 แสดงการเปรียบเทียบระหว่าง Traditional Programming กับ Machine Learning [12]

โดย Machine Learning จะแบบออกเป็น 2 ประเภทคือ

Supervised Learning คือการเรียนรู้แบบมีผู้สอน โดยเราจะเป็นคนคอยสอน Machine Learning ว่าถ้า Inputs เข้ามาในโปรแกรมรูปแบบนี้ Output ของโปรแกรมควรจะเป็นแบบไหนเช่น ให้ Machine Learning เรียนรู้เรื่องการสอบผ่านหรือสอบตกของนักเรียน

ชื่อ	คะแนน	Label
A	90	Pass
B	25	Fail
C	73	Pass
D	51	Pass
E	40	Fail
F	85	?

ตารางที่ 2.1 แสดง Input สำหรับ Supervised Learning [12]

โดย Label ทั้งหมดคือการสอน Machine Learning ว่า คะแนนเท่าไหนควรจะเป็น Pass หรือคะแนนเท่าไหนควรจะเป็น Fail แล้ว Machine Learning ก็จะสามารถรู้ว่า นักเรียนชื่อ F จะมี Label เป็น Pass
Unsupervised Learning คือการเรียนรู้แบบไม่มีผู้สอน Machine learning จะเรียนรู้เองโดยที่เราต้องคอยกำหนดว่าเราต้องแบ่ง Label เป็นทั้งหมดกี่ประเภทแล้วหลังจากนั้น Machine Learning จะเป็นคนกำหนดเองว่าคะแนนเท่าใดควรจะเป็นกลุ่มไหน

ชื่อ	คะแนน
A	90
B	25
C	73
D	51
E	40
F	85

ตารางที่ 2.2 แสดง Input สำหรับ Unsupervised Learning [12]

ชื่อ	คะแนน	Output
A	90	1
B	25	2
C	73	1
D	51	1
E	40	2
F	85	1

ตารางที่ 2.3 แสดง Output สำหรับ Unsupervised Learning [12]

2.1.4 Python

เป็นหนึ่งในภาษาของการเขียนโปรแกรมที่ติดอันดับการเขียน Programming ยอดฮิตประจำปี 2018 โดย Python นั้นจะได้รับความนิยมสูงสุด เพราะภาษานี้มีความหลากหลายมากกว่าภาษาอื่นๆ โดยสามารถเข้าถึงภาษาอื่นๆ ด้วยฟังก์ชันต่างๆ ในภาษา Python เองโดยครอบคลุมการใช้งานสำหรับ Web, Mobile, Enterprise และ Embedded



รูปที่ 2.6 แสดง 10 อันดับความนิยมของภาษา Programming [3]

2.2 งานวิจัยที่เกี่ยวข้อง

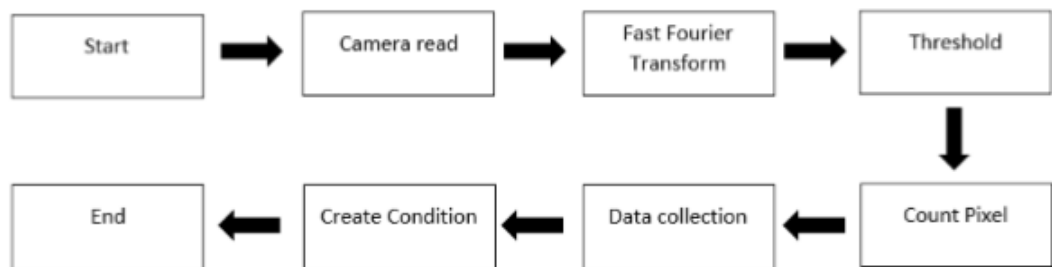
2.2.1 การตรวจสอบชนิดเลนส์โดยการประเมินผลภาพ

ถนัดดา ถาวรกุล, 2560, ได้ศึกษาและเก็บข้อมูลเลนส์ โดยเลนส์จะสามารถแบ่งได้ตามคุณสมบัติ ค่าความโค้ง ขนาดของเลนส์ ค่าสายตาเป็นต้น ซึ่งในการตรวจสอบคุณภาพเลนส์ด้วยเครื่องโพลาริเซชันนั้น จะมีความไม่สมบูรณ์ได้ 2 แบบคือ มีเส้นปรากฏบนเลนส์ (41) และมีวงกลมปรากฏบนเลนส์ (40) โดยมีการตรวจสอบโดยใช้การประมวลผลภาพทั้ง 2 แบบคือ

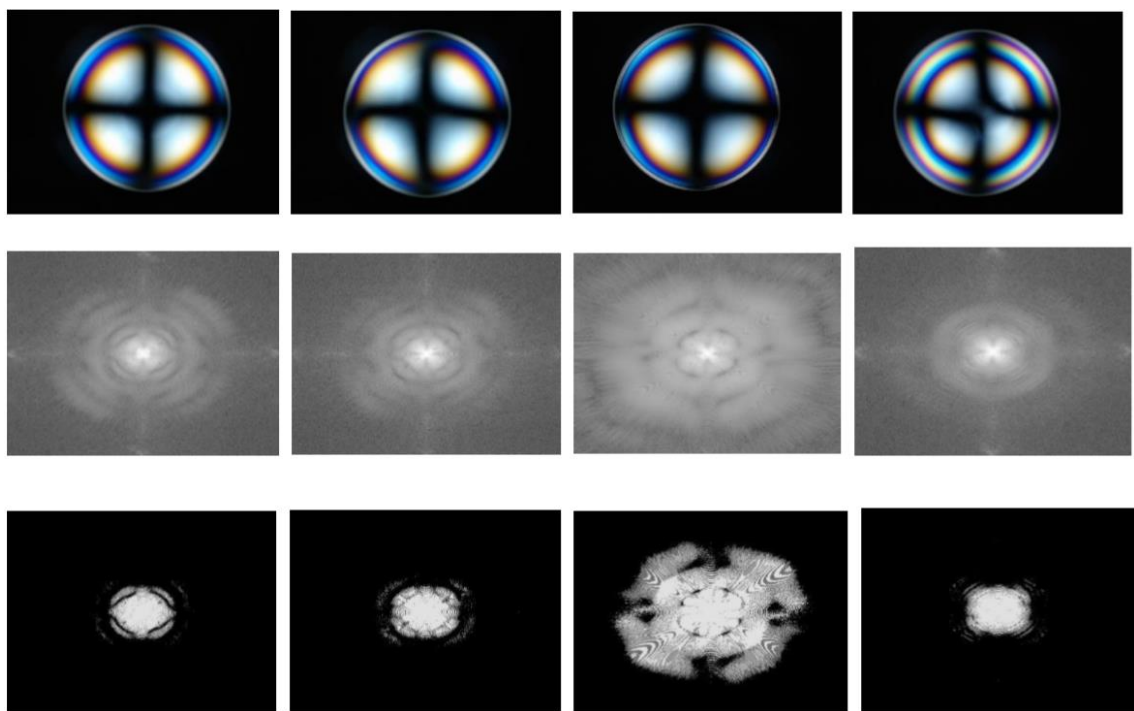
2.2.1.1 การตรวจสอบด้วยวิธีการนับจำนวนพิกเซลของภาพ

1) วิธีการตรวจสอบด้วยวิธีการหาพิกเซลบนภาพจะเป็นการทำ FFT (Fast Fourier Transform) และนำภาพที่ได้จากนั้นกำหนดค่าแสงและสีไว้เพื่อไปตัดภาพส่วนนั้นทิ้งหลัง

จสกนั้น จึงนับค่าพิกเซลในภาพต่อเพื่อที่จะวิเคราะห์ว่าเป็นเลนส์ที่มีคุณภาพ หรือ ไม่มีคุณภาพ ด้วย Python



รูปที่ 2.7 แผนผังการทำงานของ การตรวจสอบด้วยวิธีการหาพิกเซลบนภาพ [7]



รูปที่ 2.8 การตรวจสอบด้วยวิธีการนับจำนวนพิกเซลของภาพ [7]

Input \ Output	เลนส์ดี	เลนส์เสีย	รวม
เลนส์ดี	44	356	400
เลนส์เสียแบบ 41	0	500	500
เลนส์เสียแบบ 40	0	300	300
รวม	44	1,156	1,200
% ความถูกต้อง	11	100	

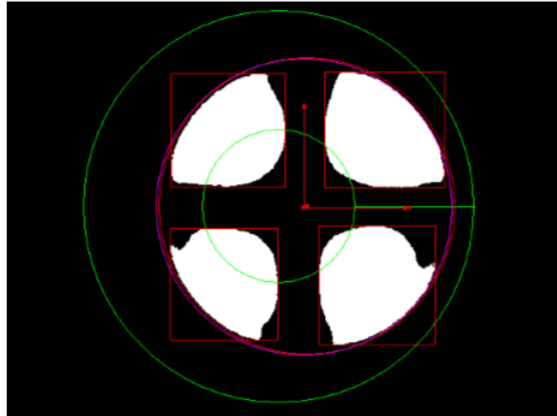
ตารางที่ 2.4 การตรวจสอบด้วยวิธีการนับจำนวนพิกเซลของภาพ [7]

2.2.1.2 การตรวจสอบด้วยวิธีการหาวัตถุบนภาพ

1) วิธีการตรวจสอบด้วยวิธีการหาวัตถุบนภาพ เป็นการตรวจสอบโดยใช้โปรแกรม NI Vision Builder โดยทำการกำหนดลอจิกเป็นเอ็กซคลูซีฟออร์ (XOR) กำหนดสีที่ให้ สีในภาพชัดขึ้น หลังจากนั้นจึงดึงโครงร่างแสงโคนสนใจเฉพาะพื้นที่ที่สว่าง แล้วนำไป Gaussian ภาพ เป็นการปรับให้ภาพเบลอแล้วนำภาพที่ได้ไป Edge Detection เพื่อวาดเส้นส่วนที่ชัดแล้วนำไป Low Pass เพื่อลด Noise เล็กๆลงแล้วนำภาพที่ได้ทั้งหมดไป Clustering โดยเลือกจุดที่สนใจเป็นสีขาวจะทำให้ภาพเด่นชัดขึ้นแล้วนำภาพที่ได้ไป Erode เพื่อลดขนาดให้เล็กลงรวมถึง Noise ของภาพจะหายไปด้วย หลังจากนั้นจึงนำภาพที่ได้ไปหาขนาดของเลนส์ที่ไม่ได้คุณภาพ หลังจากนั้นนำเลนส์มาหาขนาดของพิกเซลที่กำหนด ว่ามี อยู่ในภาพหรือไม่ หากไม่มี พิกเซลที่กำหนดจะถือว่าเป็นเลนส์ที่ไม่มีคุณภาพ



รูปที่ 2.9 แผนภาพแสดงการทำงานของ การตรวจสอบคุณภาพเลนส์ด้วยวิธีการหาวัตถุที่มีขนาดใกล้เคียงกันที่เกิดบนภาพ [7]



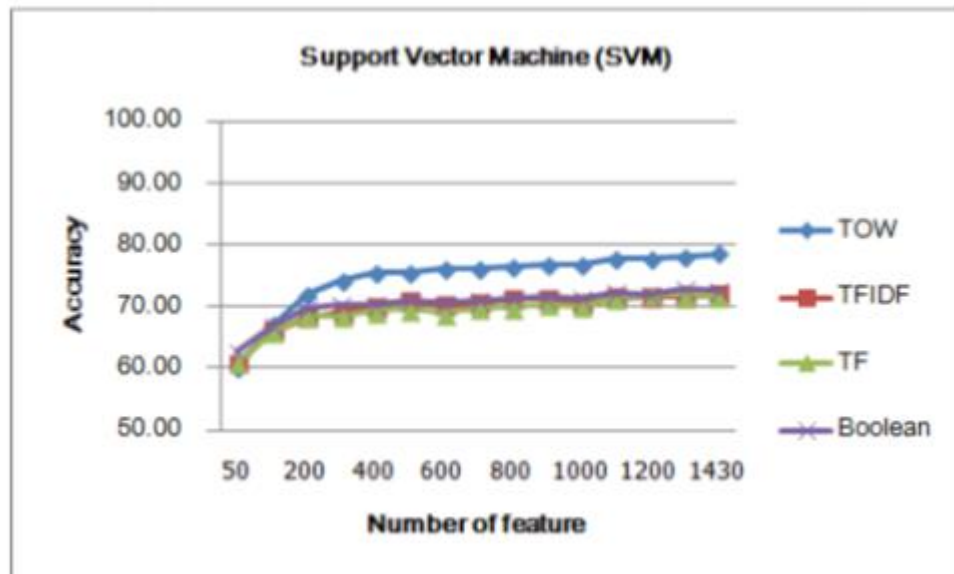
รูปที่ 2.10 การตรวจสอบด้วยวิธีการหาวัตถุบนภาพ [7]

Input \ Output	เลนส์ดี	เลนส์เสีย	รวม
เลนส์ดี	112	288	400
เลนส์เสียแบบ 41	20	480	500
เลนส์เสียแบบ 40	63	237	300
รวม	195	1,005	1,200
% ความถูกต้อง	28	90	

ตารางที่ 2.5 การตรวจสอบด้วยวิธีการหาวัตถุบนภาพ [7]

2.2 การพัฒนาประสิทธิภาพแบบจำลองของการจำแนกอารมณ์จากข้อความภาษาไทยโดยใช้เทคนิคปรับปรุงดัชนีของคำรวมกับการเรียนรู้ของเครื่อง

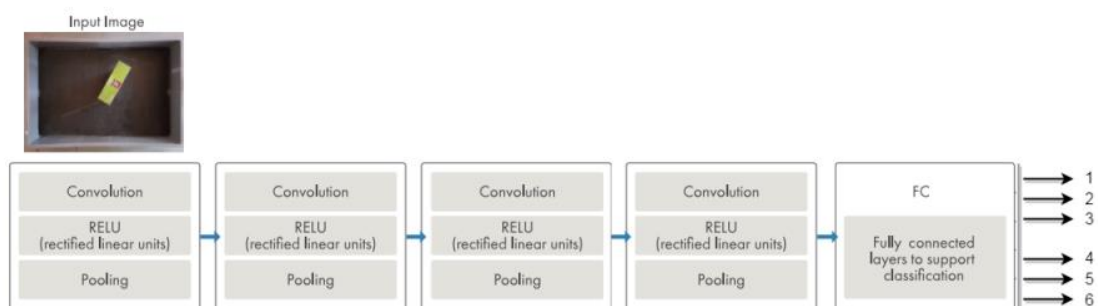
นิเวศ วิระวชิตชัย ,2557, ได้ศึกษาเกี่ยวกับการจำแนกอารมณ์โดยการปรับปรุงดัชนีของคำรวมนั้นเป็นการหาวิธีคำนวณค่าน้ำหนักต่างๆที่มีผลกับค่าความแม่นยำที่มากที่สุด เพื่อนำไปประยุกต์ใช้กับการเลือกอัลกอริทึมและการเลือกค่าดัชนีให้เหมาะสมกับชิ้นงาน



ตารางที่ 2.6 การเปรียบเทียบประสิทธิภาพอัลกอริทึมซัพพอร์ตเวกเจอร์แมชชีน [8]

2.3 A Deep Learning Prediction Model for Object Classification

N.E. Sahla,2018, เป็นการหาวัตถุบนภาพโดยใช้โปรแกรม MATLAB ในการทำ Deep Learning โดยใช้ CNN เข้ามาในการหาวัตถุบนภาพ โดยมีการเปรียบเทียบการทำงานระหว่าง CPU และ GPU ทำให้เห็นถึงประสิทธิภาพและความแตกต่างรวมถึงรูปแบบของ Dataset ที่ทำให้วิเคราะห์ภาพจากการ Train ข้อมูลได้



รูปที่ 2.11 แสดงวิธีการทำงานด้วย MATLAB ของ Deep Learning [9]



รูปที่ 2.12 แสดงตำแหน่งการวางกล้อง [9]

2.4 การประยุกต์ใช้ระเบียบวิธีการเรียนรู้เชิงลึกเพื่อจำแนกประเภทการใช้ที่ดิน ด้านเกษตรกรรมในประเทศไทยโดยใช้ซอฟต์แวร์คาเฟ

วุฒิชัย บุญพุก และคณะ, ได้มีการเปรียบเทียบหาความแตกต่างระหว่างการใช้โครงข่ายประสาทเทียม ALEXNET, CAFFENET และ FLICKNET โดยมีการรบกวนในรูปแบบที่ต่างกันไป เช่น Salt and Pepper หรือ Gaussian จนเห็นความแตกต่างระหว่าง Model โดยใช้ Activate Function ได้แก่ BNLL และ ELU

	CAFFENET			ALEXNET			FLICKNET		
	BNLL	ELU	ReLU	BNLL	ELU	ReLU	BNLL	ELU	ReLU
Exp	67	71	67	26	21	61	20	20	21
Fixed	94	93	96	20	20	96	20	20	95
Inv	96	96	96	20	20	98	20	20	96
Multistep	96	95	94	20	20	97	20	20	95
Poly	96	94	96	20	20	94	20	20	91
Sigmoid	95	95	94	20	20	97	20	20	98
Step	89	93	91	20	20	77	20	20	82

ตารางที่ 2.7 แสดงการเปรียบเทียบแต่ละ โมเดลใน Activate Function ที่ต่างกัน [10]

2.5 Computer Vision and Deep Learning in Autonomous Drones

Markus Teigen Pike, 2559, ได้ศึกษาเกี่ยวกับการใช้ Computer Vision และใน โดรนซึ่งมีการศึกษาเกี่ยวกับ Computer Vision ไว้หลายรูปแบบโดยอธิบายเกี่ยวกับการใช้

2.5.1 Deep learning models

- 1) Convolutional Neural Networks (CNNs)
- 2) AlexNet
- 3) Googles Inception Model
- 4) VGGNet
- 5) ResNet
- 6) Comparison of Classification Networks

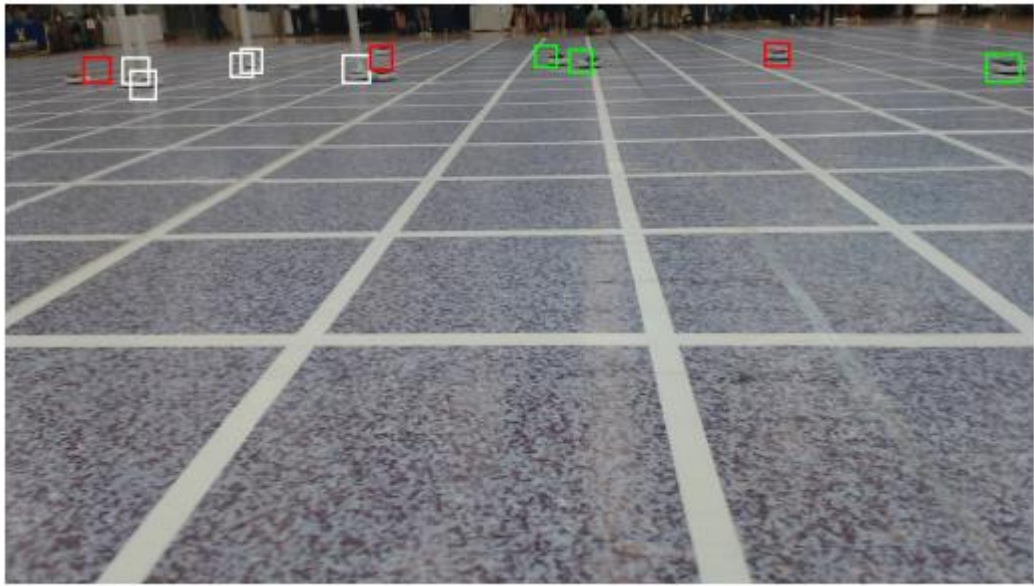
2.5.2. Object Detection Networks

- 1) Region-based Convolutional Neural Networks (R-CNN)
- 2) YOLO - You Only Look Once
- 3) SSD - Single Shot Multibox Detection
- 4) YOLOv2
- 5) Comparison of Detection Networks
- 6) Dlib and Max-Margin Object-Detection (MMOD)

2.5.3.Semantic Segmentation

- 1) Sliding Window
- 2 Early stage of CNN Image Segmentation
- 3) FCNs - Fully Convolutional Networks
- 4) Deconvolutional Networks for Semantic Segmentation
- 5) SegNet
- 6) DeepLab
- 7) Comparing Segmentation Networks

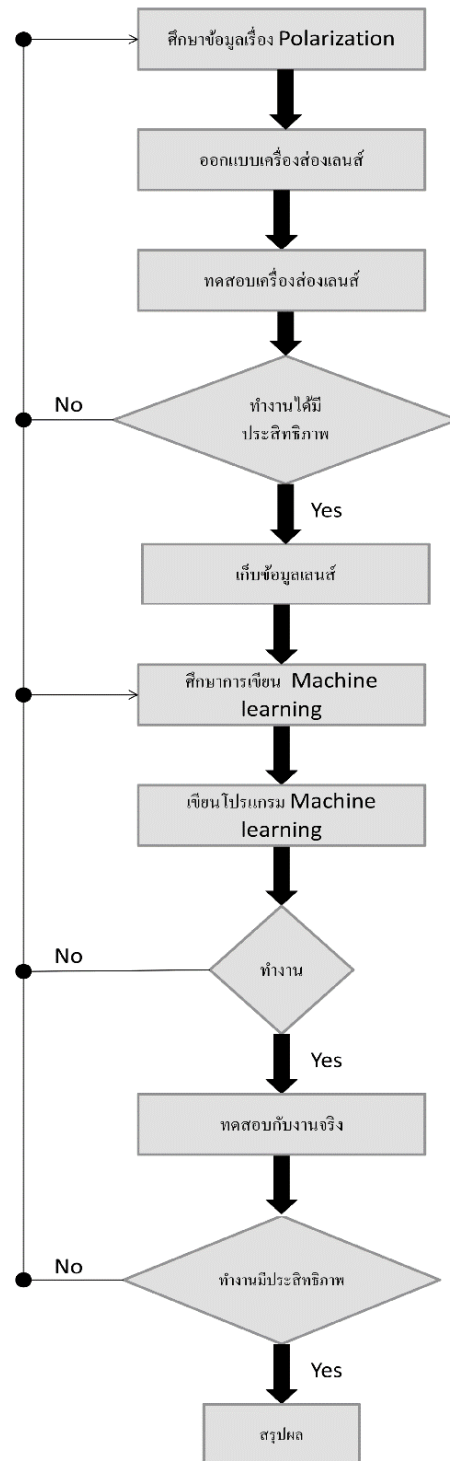
โดยวิธีทั้งหมดได้มีการทดลองและการทำงานตัวอย่างรวมถึง รูปแบบการทำงานของโปรแกรมทั้งหมด โดยมี Accuracy สูงถึง 85%



รูปที่ 2.13 แสดงการตรวจจับหุ่นยนต์จากระยะไกล [11]

บทที่ 3 ระเบียบวิธีวิจัย

3.1 ขั้นตอนการดำเนินงาน



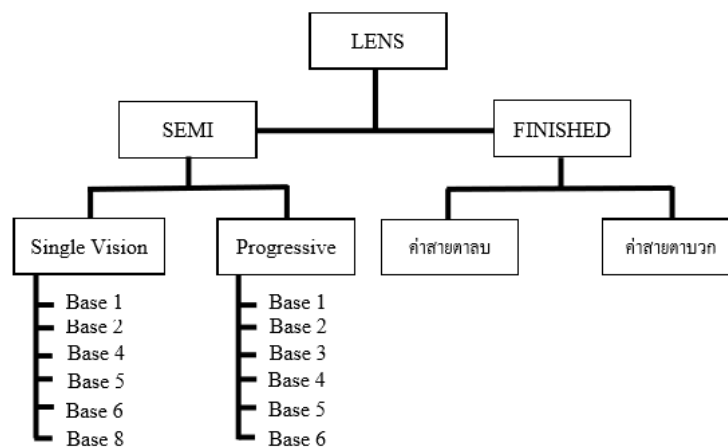
รูปที่ 3.1 แสดงขั้นตอนการทำงานระบบ

3.2 การสำรวจสภาพปัจจุบัน

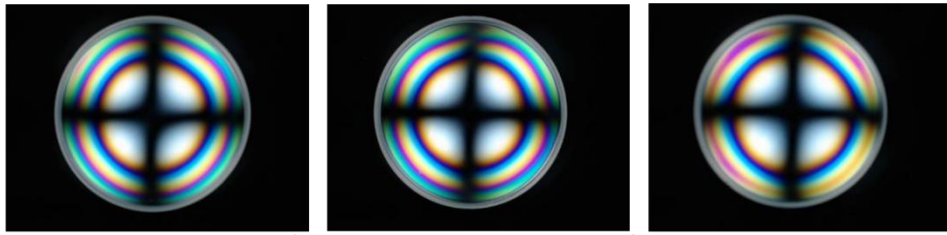
เป็นการศึกษาปัญหาของเลนส์ที่มีอยู่ในปัจจุบันของโรงงานอุตสาหกรรมในขั้นตอนการคัดแยกเลนส์

3.2.1 ศึกษาข้อมูลเกี่ยวกับเลนส์

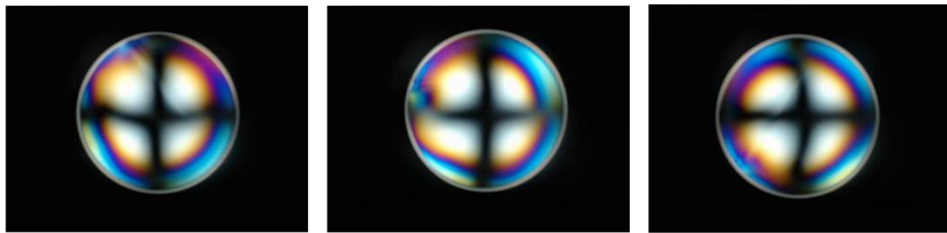
โดยการผลิตของเลนส์ในบริษัทนั้น จะมีเลนส์หลายรูปแบบต่างกันไปโดยแบ่งเป็น SEMI และ FINISHED โดยในเลนส์ SEMI จะแบ่งเป็น Single Vision 6 ประเภท และ Progressive 6 ประเภท ส่วนเลนส์ FINISHED จะแบ่งเป็น ค่าสายตาเลนและบวก



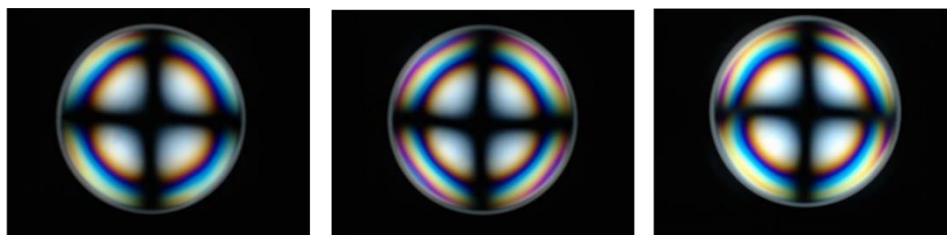
รูปที่ 3.2 เป็นการแสดงประเภทของเลนส์ที่บริษัททำการผลิตทั้งหมด



รูปที่ 3.3 ตัวอย่างเลนส์ SEMI Base 1 ที่ได้คุณภาพ



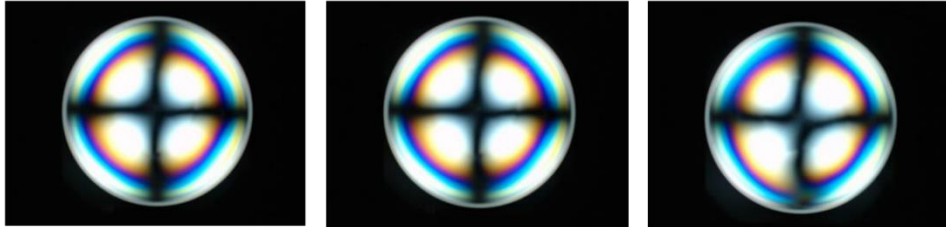
รูปที่ 3.4 ตัวอย่างเลนส์ SEMI Base 1 ที่ไม่ได้คุณภาพแบบมีเส้น (41)



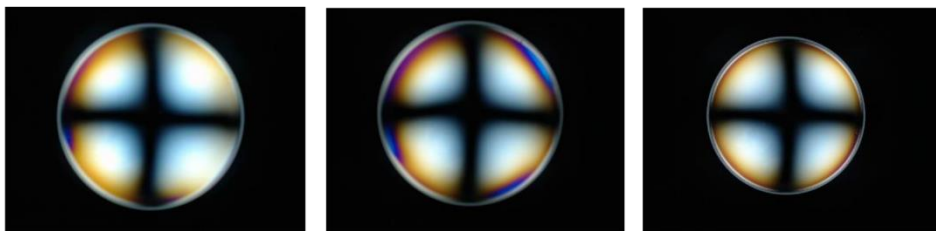
รูปที่ 3.5 ตัวอย่างเลนส์ SEMI Base 2 ที่ได้คุณภาพ



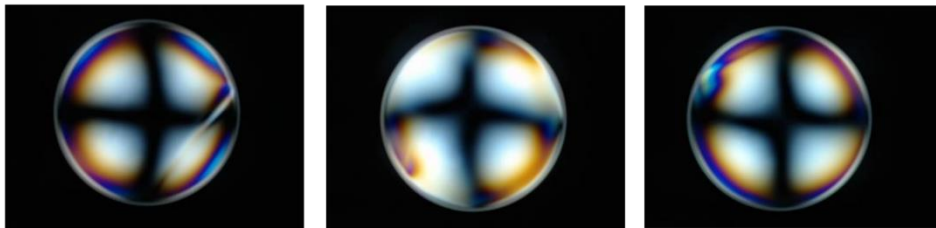
รูปที่ 3.6 ตัวอย่างเลนส์ SEMI Base 2 ที่ไม่ได้คุณภาพแบบมีเส้น (41)



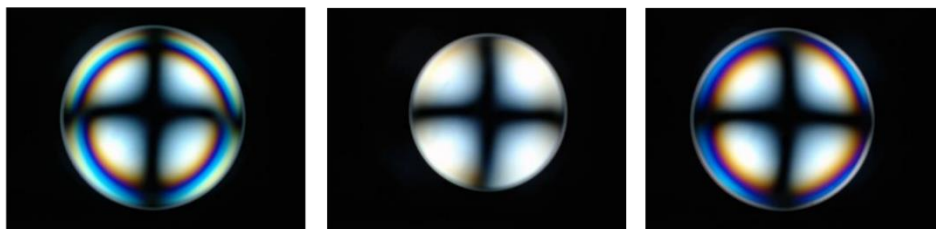
รูปที่ 3.7 ตัวอย่างเลนส์ SEMI Base 2 ที่ไม่ได้คุณภาพแบบมีวงกลม (40)



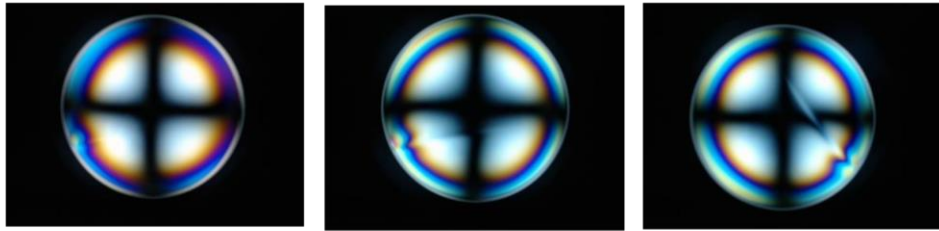
รูปที่ 3.8 ตัวอย่างเลนส์ SEMI Base 4 ที่ได้คุณภาพ



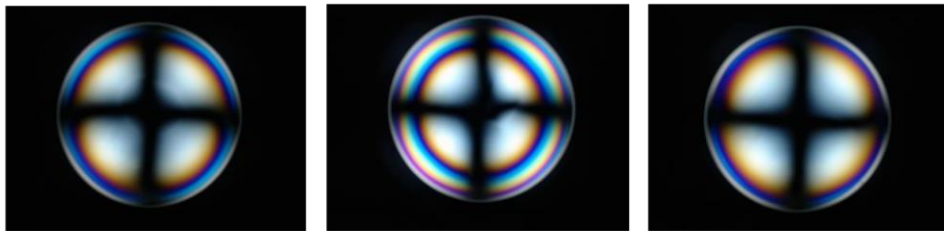
รูปที่ 3.9 ตัวอย่างเลนส์ SEMI Base 4 ที่ไม่ได้คุณภาพแบบมีเส้น (41)



รูปที่ 3.10 ตัวอย่างเลนส์ SEMI Base 5 ที่ได้คุณภาพ



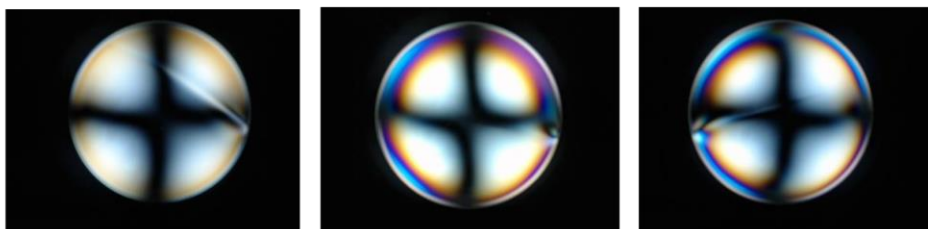
รูปที่ 3.11 ตัวอย่างเลนส์ SEMI Base 5 ที่ไม่ได้คุณภาพแบบมีเส้น (41)



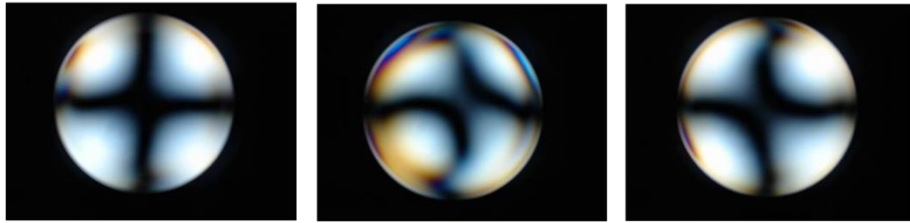
รูปที่ 3.12 ตัวอย่างเลนส์ SEMI Base 5 ที่ไม่ได้คุณภาพแบบมีวงกลม (40)



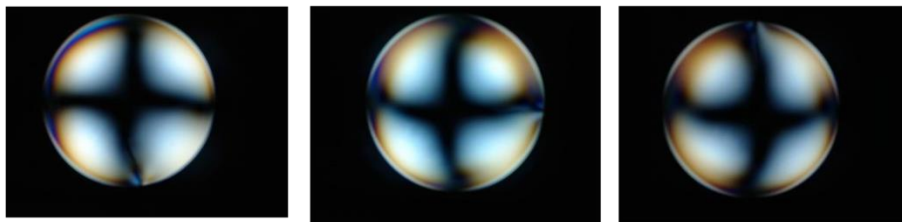
รูปที่ 3.13 ตัวอย่างเลนส์ SEMI Base 6 ที่ได้คุณภาพ



รูปที่ 3.14 ตัวอย่างเลนส์ SEMI Base 6 ที่ไม่ได้คุณภาพแบบมีเส้น (41)



รูปที่ 3.15 ตัวอย่างเลนส์ SEMI Base 8 ที่ได้คุณภาพ



รูปที่ 3.16 ตัวอย่างเลนส์ SEMI Base 8 ที่ไม่ได้คุณภาพแบบมีเส้น (41)

3.3 การออกแบบเครื่องส่องเลนส์



รูปที่ 3.17 เครื่องส่องเลนส์ที่สร้างขึ้นมาเพื่อใช้ในการวิจัยนี้

เครื่องส่องเลนส์ที่สร้างขึ้นมามีสำหรับใช้ในการวิจัยนี้จะใช้โดยการนำเลนส์เข้าไปวางบนแผ่นอะคริลิกแล้วเปิดไฟ LED เพื่อให้ส่องขึ้นมาผ่านแผ่นโพลาไรซ์ หลังจากนั้นจะสามารถดูเลนส์ผ่านแผ่นโพลาไรซ์ชั้นบนอีกที

3.4 การออกแบบทางกล

3.4.1 แผ่นอะคริลิก

เป็นแผ่นทึบแสง ทนทานต่อการใช้งานและง่ายต่อการทำความสะอาดและมีราคาถูก จึงนิยมใช้ในโรงงานอุตสาหกรรม

3.4.2 แผ่นโพลาไรซ์

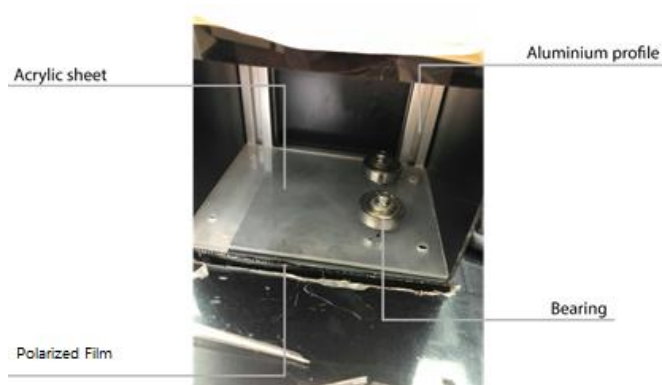
เป็นหนึ่งในอุปกรณ์สำหรับการบังคับแสงให้ออกมาเพียงระนาบเดียวเป็นองค์ประกอบสำหรับการทำโพลาไรเซชัน

3.4.3 แบร์ริง

เป็นอุปกรณ์สำหรับการหมุนเลนส์เพื่อเพิ่มประสิทธิภาพในการรับภาพได้หลากหลายองศา

3.4.4 อลูมิเนียมโปรไฟล์

เป็นหนึ่งในโครงสร้างของเครื่องจักรที่นิยมใน โรงงานอุตสาหกรรม มีน้ำหนักเบา ติดตั้งได้ง่ายและแข็งแรง



รูปที่ 3.19 แสดงการออกแบบทางกล



รูปที่ 3.20 แสดงอุปกรณ์ทั้งหมดของระบบ

3.5 การออกแบบทางไฟฟ้าและอิเล็กทรอนิกส์

3.5.1 เครื่อง NUVO-5501 INDUSTRIAL

เป็นคอมพิวเตอร์สำหรับโรงงานอุตสาหกรรมโดยเฉพาะ ที่มีอายุการใช้งานมากกว่าคอมพิวเตอร์ทั่วไปและมีคุณสมบัติ ทนทาน รวมถึง มีที่ช่วยระบายความร้อน

3.5.2 ไฟ LED Lamp

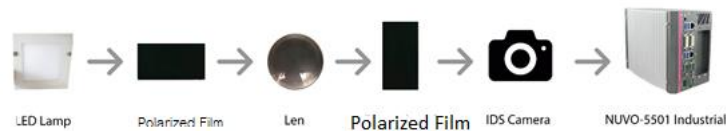
เป็นอุปกรณ์สำหรับการให้แสงสว่าง โดยเป็นองค์ประกอบสำหรับการทำโพลาไรเซชัน

3.5.3 กล้อง UI-3240LE-NIR CAMERA

เป็นกล้องสำหรับจับวัตถุมีหน้าที่ส่งสัญญาณภาพผ่านคอมพิวเตอร์มีไว้สำหรับการใช้งานทั้งการรับภาพนิ่งและภาพเคลื่อนไหว สำหรับเขียนโปรแกรมต่อไปในภายหลัง

3.6 ภาพรวมของระบบ

จะมีการใช้ไฟ LED Lamp ส่องผ่าน ฟิล์มโพลาไรซ์ เพื่อที่จะทำให้แสงเป็นระนาบเดียวกัน โดยหลังจากนั้นแสงที่ผ่านฟิล์มจะผ่านเลนส์และเข้าไปยังกล้องเพื่อที่จะได้ใช้กล้องในการรวบรวมภาพที่สามารถตรวจจับได้ และส่งข้อมูลไปยังคอมพิวเตอร์ เพื่อไปประมวลผลสำหรับ Machine Learning หลังจากนั้น ระบบจะแสดงออกมาว่าผลผลิตดังกล่าวได้คุณภาพหรือไม่



รูปที่ 3.21 แสดงภาพรวมของระบบ

3.7 การออกแบบโปรแกรม

จะมี 2 ขั้นตอนหลักๆคือ

- 1) Preprocess ภาพ คือ การนำภาพที่ได้จากกล้องไปปรับแต่งภาพเพื่อนำไปเป็นข้อมูล ในการทำ Machine Learning ในขั้นต่อไปโดยทั้งหมดใช้โปรแกรม Python โดยใช้ ร่วมกับ OpenCV
- 2) Machine Learning คือ Machine Learning จะเป็นการนำภาพที่ถูก Preprocess ทั้งหมด ไปเรียนรู้เพื่อนำไปวิเคราะห์คุณภาพของเลนส์และตัดสินใจในการคัดแยกออกทั้งหมด โดยใช้โปรแกรม Python โดยใช้ Fast-CNN เข้ามาเป็น Model สำหรับการเรียนรู้

3.8 Machine Learning Detail

ในการทำ Machine Learning นั้นจะมีหลักการสำคัญในการแบ่งข้อมูลเพื่อทดสอบอยู่ 3 วิธีการ

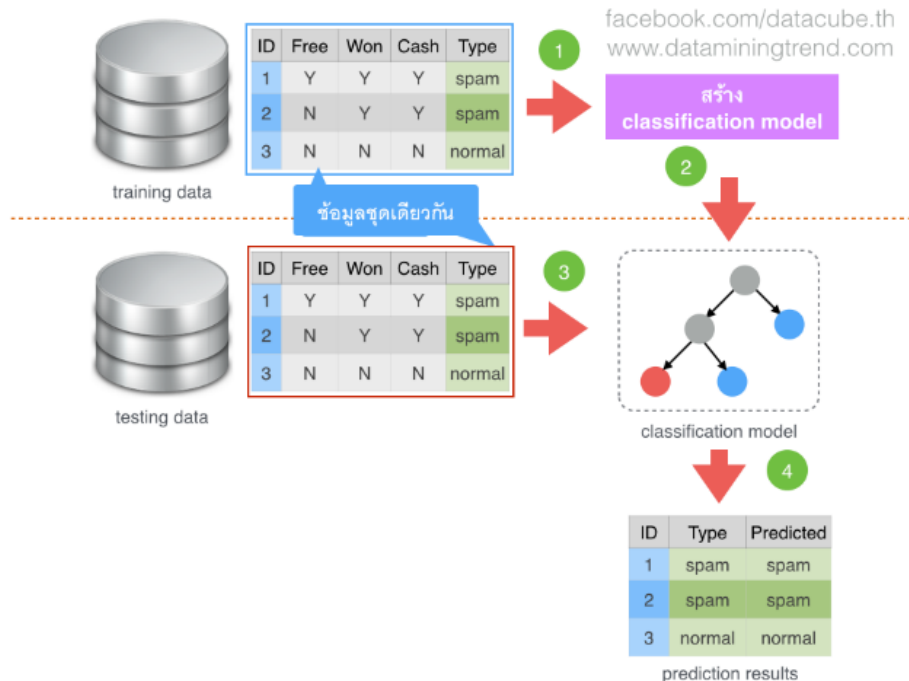
1. Self-Consistency Test

โดยวิธีนี้จะเป็นการนำข้อมูลที่ Train เข้ามา ไป Test กับข้อมูลชุดเดิม โดยไม่มีการแบ่งข้อมูล โดยวิธีนี้เป็นวิธีที่ง่ายที่สุดสำหรับทดสอบว่า Code ของเรามี ปัญหาหรือไม่ โดยวิธีนี้จะมีประสิทธิภาพที่สูงมากใกล้เคียง 100% แต่ในกรณีนี้จะหมายความว่าเราจะไม่สามารถรู้ได้เลยว่า Model ของเรานั้น Overfit หรือไม่ แต่สามารถทดสอบ Code และได้ว่าเราเขียนแล้วสามารถ Train ออกมาแล้วใช้ได้หรือไม่ เพื่อที่จะนำ Code ที่ทำได้ไปพัฒนาต่อในวิธีอื่นๆ รวมถึงดูแนวโน้มของข้อมูลได้ ถ้าหากว่าทำวิธีนี้แล้วมีประสิทธิภาพน้อยแสดงว่าเราเลือก Model ไม่เหมาะสมกับข้อมูลที่เรามี จะไม่เสียเวลาไปทดสอบกับวิธีการแบ่งข้อมูลแบบอื่นๆให้เสียเวลา

Self-consistency Test

(data)³
base | warehouse | mining

- ใช้ข้อมูล training ในการทดสอบประสิทธิภาพของโมเดล



รูปที่ 3.22 แสดงการทำ Self-consistency Test [7]

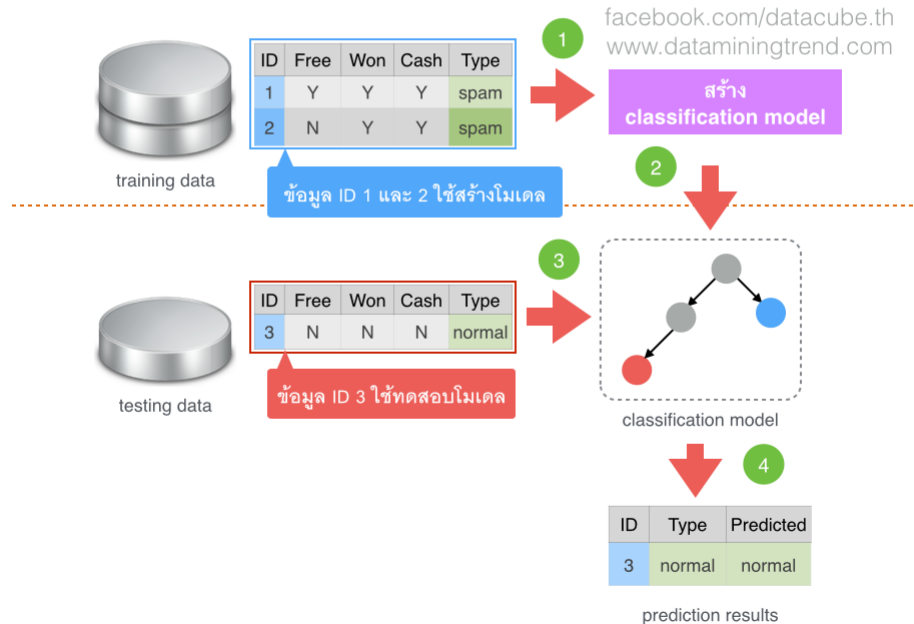
2. Split-Test

วิธีนี้จะเป็นการสุ่มแบ่งข้อมูลออกมาเป็น 2 ส่วน เช่น Test 20% Train 80% โดยข้อมูลส่วน Train จะเป็นข้อมูลสำหรับสอนคอมพิวเตอร์ ส่วนข้อมูลส่วน Test จะเป็นข้อมูลส่วนของการทดสอบ แต่วิธีนี้จะไม่เหมาะกับการแบ่งข้อมูลที่มีความแตกต่างกันมากๆ เนื่องจาก เป็นการสุ่ม อาจจะทำให้สุ่มได้ข้อมูลที่ตรงกับ ส่วนที่คล้ายกับ Model ที่สร้างแต่ในทางตรงกันข้ามถ้าหากข้อมูลที่สุ่มมามีลักษณะไม่คล้ายกับข้อมูลที่สอนคอมพิวเตอร์จะออกมามีประสิทธิภาพได้ดี ดังนั้นเราควร จะทำการ Split Test หลายๆ ครั้งแต่ข้อดี ของ วิธีนี้คือมี ความเร็วที่รวดเร็วเหมาะ กับข้อมูลที่มีชุดขนาดใหญ่หลายๆ และผมได้เลือกวิธีนี้เพราะว่า ข้อมูลที่เป็นส่วน ของการแยกแยะระหว่างข้อมูลที่มีความแตกต่างกันมีความแตกต่างกันที่น้อยมาก จึง ทำให้ข้อดีของมัน ถูกกลบหายไป และส่วนตัวผมมี ข้อมูลที่มากจึงเหมาะกับ วิธีการแบ่งข้อมูลแบบนี้

Split Test

(data)³
base | warehouse | mining

- แบ่งข้อมูลออกเป็น 2 ชุด
- training data สำหรับสร้างโมเดล และ testing data สำหรับทดสอบ



รูปที่ 3.23 แสดงการทำ Split Test [7]

3. Cross-Validation Test

วิธีนี้เป็นที่นิยมมากที่สุดเนื่องจาก การวัดประสิทธิภาพของ Model จะมีความน่าเชื่อถือ เพราะสามารถใช้ได้กับชุดข้อมูลทุกรูปแบบ โดยวิธีการแบ่ง Cross-Validation จะใช้ ค่า K เช่น 7-Fold-Cross-Validation โดยจะทำการแบ่งข้อมูลออกมาเป็น 7 ส่วนโดยแต่ละส่วนที่แบ่งมาจะมีขนาดเท่ากัน เช่น ชุดข้อมูล 70 ก็แบ่งออกมา ชุดละ 10 หลังจากนั้นนำข้อมูลแต่ละชุดมาทดสอบประสิทธิภาพของ Model โดยจากรูปด้านล่างเราแบ่งข้อมูลออกมาเป็น 5 ส่วนที่มีจำนวนเท่าๆกัน และทดสอบโดยจะแบ่งเป็น 5 รอบ

รอบที่ 1 จะใช้ข้อมูล 2,3,4,5 ในการ Train และใช้ข้อมูลที่ 1 ในการ Test

รอบที่ 2 จะใช้ข้อมูล 1,3,4,5 ในการ Train และใช้ข้อมูล ที่ 2 ในการ Test

รอบที่ 3 จะใช้ข้อมูล 1,2,4,5 ในการ Train และใช้ข้อมูล ที่ 3 ในการ Test

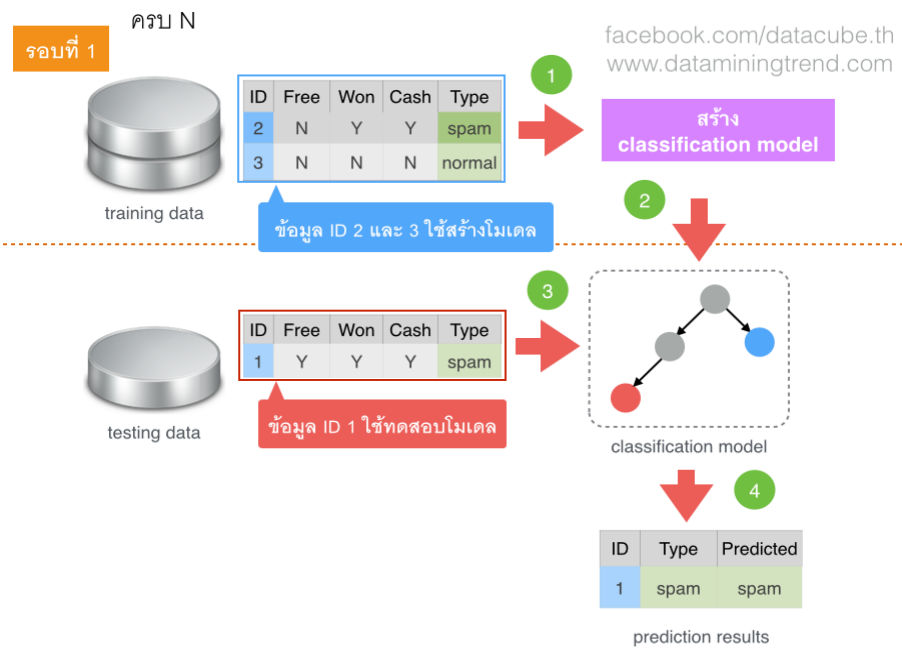
รอบที่ 4 จะใช้ข้อมูล 1,2,3,5 ในการ Train และใช้ข้อมูล ที่ 4 ในการ Test

รอบที่ 5 จะใช้ข้อมูล 1,2,3,4 ในการ Train และใช้ข้อมูล ที่ 5 ในการ Test

Cross-validation

(data)³
base | warehouse | mining

- แบ่งข้อมูลออกเป็น N ชุด เช่น N = 5 หรือ 10
- ข้อมูล N-1 ชุดสำหรับสร้างโมเดล และ ข้อมูลส่วนที่เหลือสำหรับทดสอบ วนทำจนครบ N

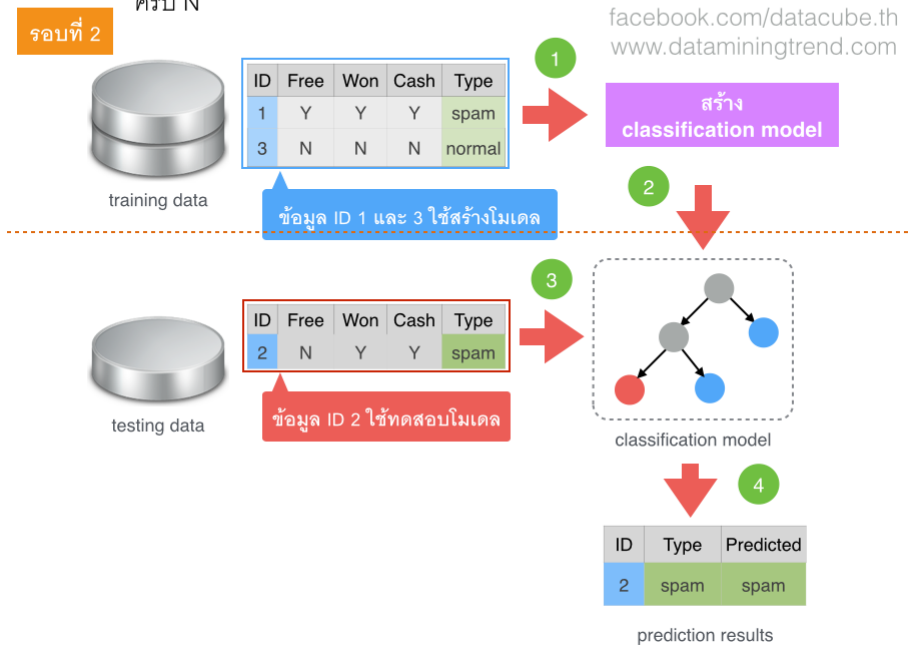


รูปที่ 3.24 แสดงการทำ Cross-Validation (1) [7]

Cross-validation

(data)³
base | warehouse | mining

- แบ่งข้อมูลออกเป็น N ชุด เช่น $N = 5$ หรือ 10
- ข้อมูล $N-1$ ชุดสำหรับสร้างโมเดล และ ข้อมูลส่วนที่เหลือสำหรับทดสอบ วนทำจนครบ N

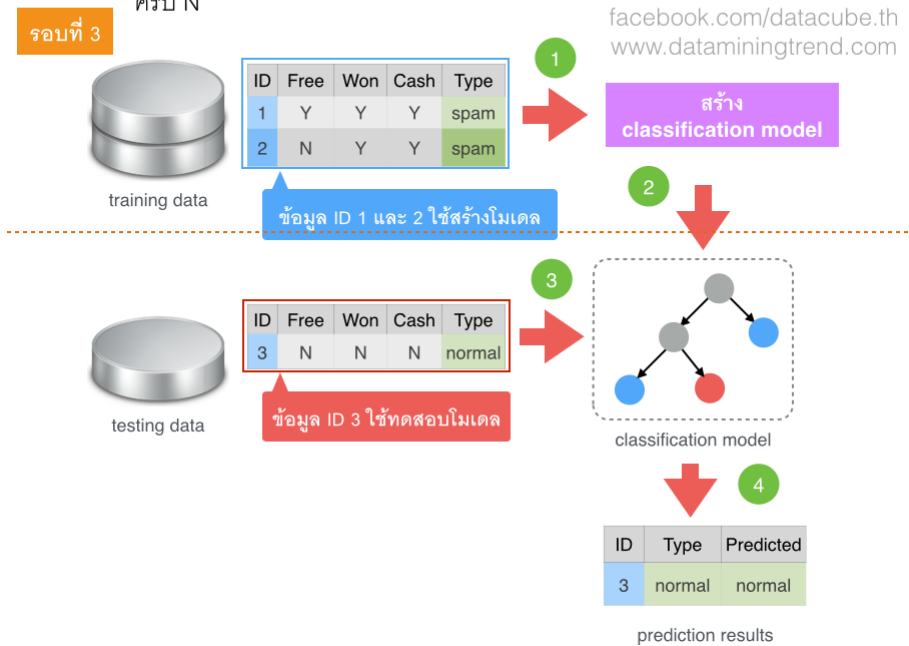


รูปที่ 3.25 แสดงการทำ Cross-Validation (2) [7]

Cross-validation

(data)³
base | warehouse | mining

- แบ่งข้อมูลออกเป็น N ชุด เช่น $N = 5$ หรือ 10
- ข้อมูล $N-1$ ชุดสำหรับสร้างโมเดล และ ข้อมูลส่วนที่เหลือสำหรับทดสอบ วนทำจนครบ N

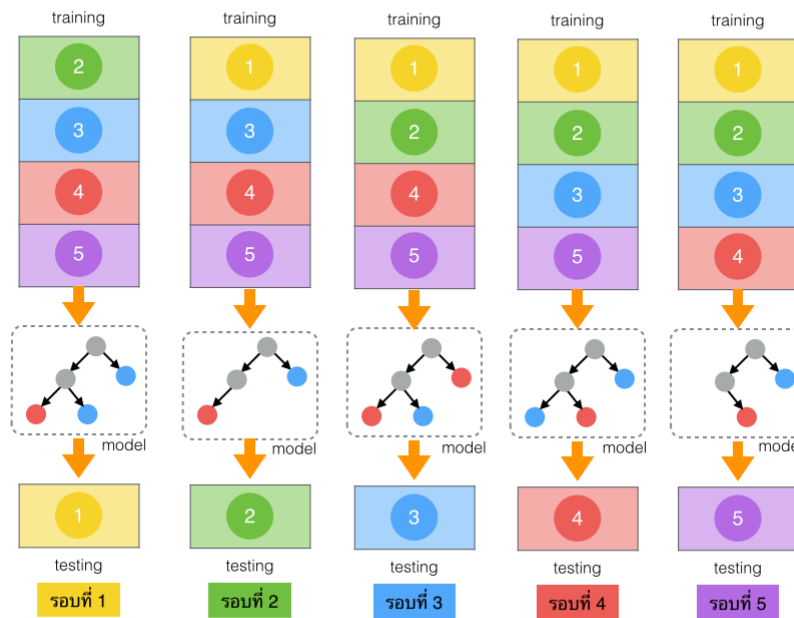


รูปที่ 3.2 แสดงการทำ Cross-Validation (3) [7]

Cross-validation



- ตัวอย่าง 5-fold cross-validation



facebook.com/datacube.th
www.dataminingtrend.com

รูปที่ 3.27 แสดงการทำ Cross-Validation (4) [7]

บทที่ 4 การทดลองและผลการทดลอง/วิจัย

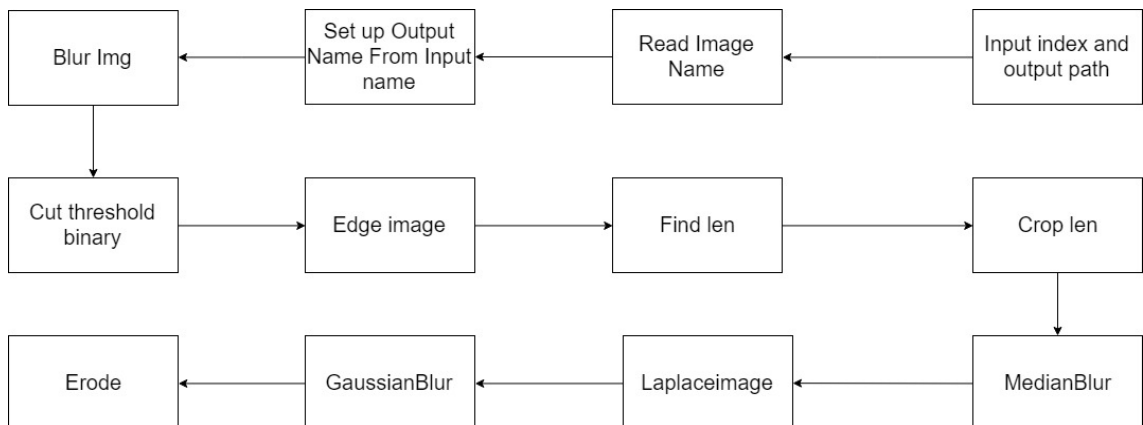
4.1 บทนำ

ในบทนี้ทางผู้วิจัยทำได้เสนอการทดลองเกี่ยวกับการเรียนรู้ของการตรวจสอบคุณภาพเลนส์ ด้วยการเรียนรู้ด้วยเครื่องเพื่อที่จะนำไปพัฒนาต่อให้มีประสิทธิภาพ และแม่นยำ สูงสุด โดยบทนี้ผู้วิจัยได้ทดลองโดยคำนึงถึงปัจจัยต่างๆที่จะมีผลต่อการทำงานของโปรแกรม โดยขั้นตอนทั้งหมดนี้จะมีขั้นตอน ดังต่อไปนี้

4.2 Machine Vision

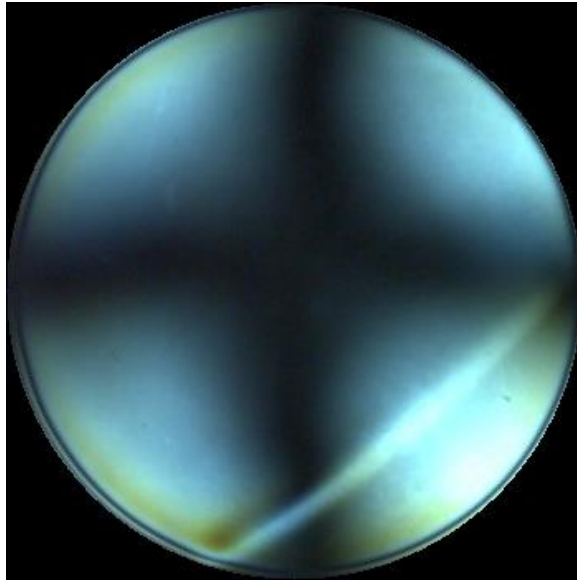
4.2.1 Image Preprocessing

ในการทดลองนี้จะใช้ภาพที่ได้จากกล้องมาตรวจสอบคุณภาพของเลนส์ว่า เป็นเลนส์ที่ไม่ได้คุณภาพแบบมีวงกลม (40) หรือ เลนส์ที่ไม่ได้คุณภาพแบบมีเส้น (41) ได้ชัดเจน โดยขั้นตอนการทำ Image Preprocessing จะมี ดังนี้



1) Crop Len

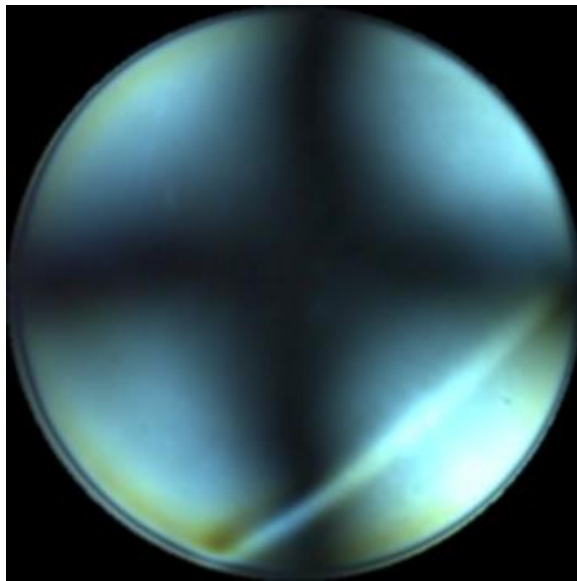
โดยขั้นตอนนี้จะทำการตัดภาพเป็นขาวดำและหาวงกลมในภาพและตัดส่วนที่เป็นวงกลมออกมาคำนวณให้ได้เป็นภาพสี่เหลี่ยมพอดี ดังรูป โดยใช้วิธีการตัด Threshold แบบ Binary ที่ (10 , 255) จากนั้น คำนวณหา Height และ Width แล้วนำไปสร้าง ภาพใหม่ โดยใช้เป็น numpy.uint8 หลังจากนั้น ตัดภาพต้นฉบับด้วย Canny ที่ Threshold (50 , 200) หลังจากนั้นหา วงกลมในภาพ โดย Set Parameter หลังจากนั้น ก็ไปหาวงกลมในภาพ แล้วนำวงกลมไป วาดใน ภาพใหม่ และ หา ภาพที่ซ้อนทับกัน และสุดท้ายนำภาพที่ซ้อนทับกัน ตัดออกมาเป็นวงกลม แล้วทำให้ภาพเป็นสี่เหลี่ยม



รูปที่ 4.1 แสดงรูปหลังผ่านขั้นตอน Crop Len

2) Blur Img

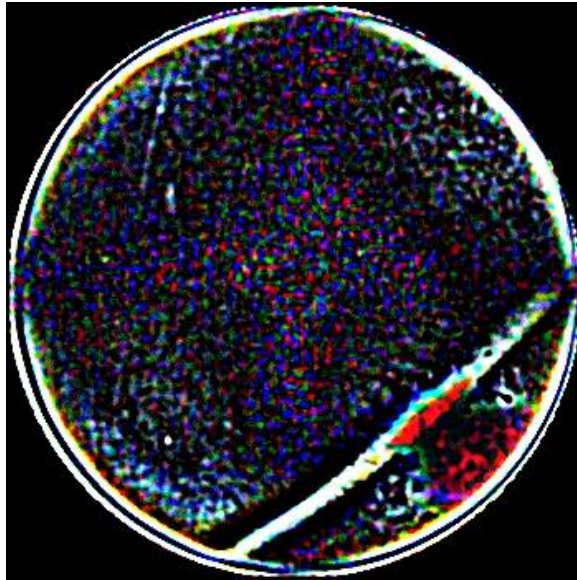
ในขั้นตอนนี้ทำการ Blur รูปภาพโดยใช้ Gaussian Blur ที่ Gaussian Kernel Size เป็น 5×5



รูปที่ 4.2 แสดงรูปหลังผ่านขั้นตอนการ Blur

3) Laplace Img

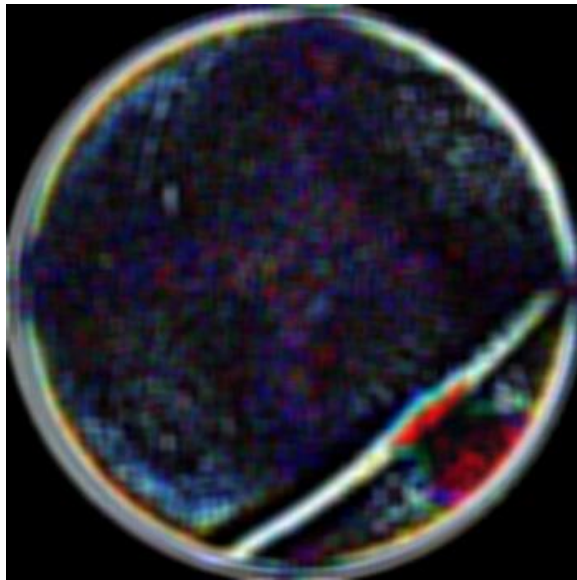
ในขั้นตอนนี้จะเป็นการนำภาพที่ได้มา Laplace โดยตั้งความลึกเป็น -1 Kernel Size เป็น แบบ 9×9



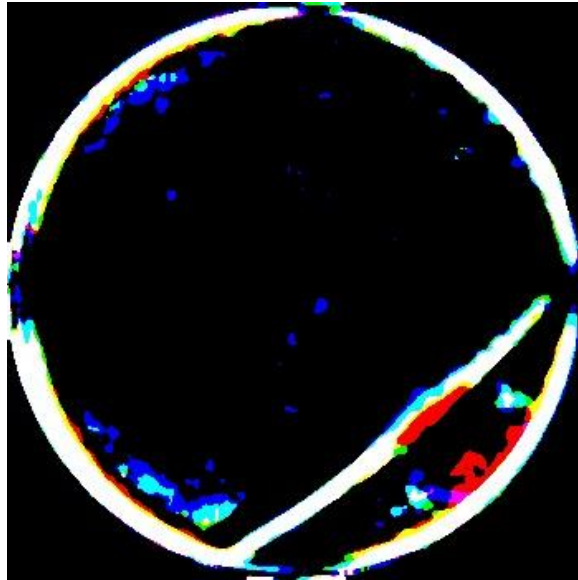
รูปที่ 4.3 แสดงรูปหลังผ่านขั้นตอนการทำ Laplace

4) Gaussian Blur

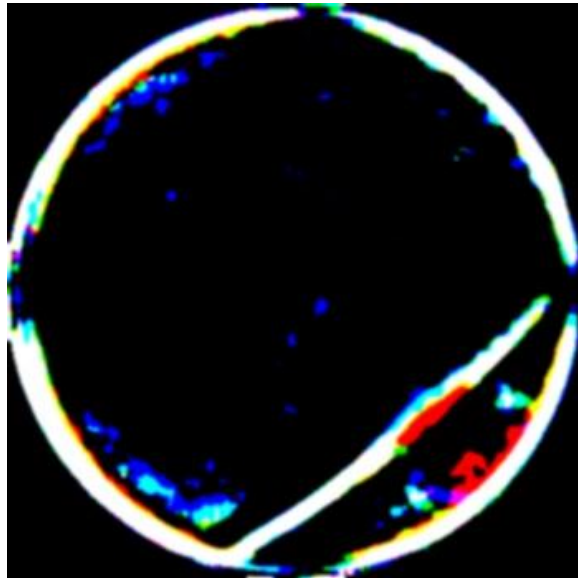
ในขั้นตอนนี้จะทำการ Gaussian Blur โดยรอบนี้ใช้เป็น Kernel Size แบบ 9×9 อีกทั้งยังนำภาพที่ได้ไปตัด Threshold แบบ Binary ที่มี Parameter (100 , 255) และนำไป Gaussian Blur อีกครั้ง ที่ Kernel Size เป็นแบบ 5×5 โดย Set Gaussian Kernel Direction เป็น 0 ทั้งแกน X และ แกน Y



รูปที่ 4.4 แสดงรูปหลังผ่านขั้นตอนการ Gaussian Blur



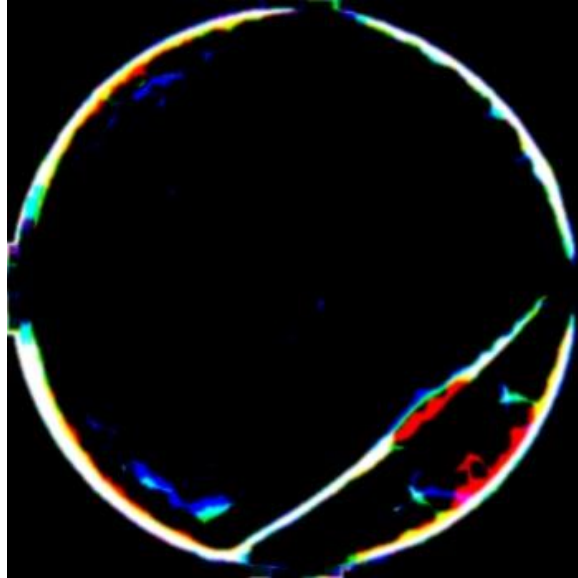
รูปที่ 4.5 แสดงรูปหลังผ่านขั้นตอนการตัด Threshold



รูปที่ 4.6 แสดงรูปหลังผ่านขั้นตอนการทำ Gaussian Blur อีกครั้ง

5) Erode

ในขั้นตอนนี้จะทำการ Erode ภาพที่ได้มาเพื่อลด Noise ในภาพโดยใช้ Kernel ที่เป็น 1 ทั้งหมดในขนาด 5*5



รูปที่ 4.7 แสดงรูปหลังผ่านขั้นตอนการทำ Erode

ในขบวนการนี้ถึงแม้ว่าจะใช้หลายวิธีการทำงานแต่สามารถทำให้ภาพที่ได้มองเห็นได้มากขึ้นและทำให้มีประสิทธิภาพที่มากขึ้นเมื่อนำไปใช้งาน

4.2.2 Machine Learning with Keras Library

โดยในการทดลองนี้ผู้วิจัยจะนำภาพที่ได้จากการทำ Preprocessing ไปเข้าสู่การเรียนรู้ด้วยเครื่องด้วยการสอน (Train) และการทดลอง (Test) ด้วยภาพทั้งหมด 5200 ภาพโดยแยกเป็นเลนส์ที่มีคุณภาพทั้งหมด 800 ภาพ และเลนส์ที่ไม่ได้คุณภาพแบบมีเส้น (41) 3000 ภาพ และ เลนส์ที่ไม่ได้คุณภาพแบบวงกลม (40) 1400 ภาพ ผ่าน Fast-CNN Model โดยเปรียบเทียบประสิทธิภาพสำหรับ Model ผ่าน 4 ตัวแปร คือ Train Accuracy, Train loss, Validation Accuracy, Validation loss โดยในขั้นตอนนี้ เราจะมี Clear Missing Value ทั้งหมดโดยการลบ Observation ที่มีข้อมูลไม่ครบออกไป โดยใช้ Pandas Library และเราจะแบ่งภาพทั้งหมด 5200 ภาพ ออกมาเป็นภาพสำหรับ Test ทั้งหมด 25% จาก 5200 ภาพ จะเป็น 1300 ภาพ สำหรับ Test โดยเราจะใช้วิธีการแบ่งเป็นแบบ Split Test ที่เราจะสุ่ม 25% เอาไว้ ทั้งหมด 42 รอบ ซึ่งหมายความว่า ใน 42 รอบนั้น ในภาพ ที่ใช้สำหรับ Test 1300 ภาพ และ Train 3900 ภาพ จะมีไม่ซ้ำกันเพื่อที่จะทำให้ Model ที่ได้นั้นไม่ Overfit หลังจากนั้นเราจะนำข้อมูลที่ได้ทีละ รอบ 1 รอบ เข้าไป Train Model โดยใน 1 รอบนั้น จะมี ภาพทั้งหมด 3900 ภาพ สำหรับการ Train โดยใน 3900 ภาพ เราจากนั้นใน 1 รอบ เราจะทำ การสอนมันโดยมีค่า Learning

Rate = 0.0001 โดยยิ่งค่า Learning Rate สูงก็จะทำให้ Model ที่ได้เกิดการเรียนรู้ที่ไวจนไม่สามารถแยกแยะภาพที่ไม่อยู่ใน Dataset หรือ อยู่ใน Dataset แต่มีความแตกต่างกันมากไป และ ใน รอบที่เรา Train เราจะทำการปรับภาพด้วยโปรแกรม(Keras Preprocessing Image Library)โดยการหมุน 45 องศา ซoomเข้าและzoomออกภาพ โดยเป็น 30% ขยับ ภาพไปทางซ้ายและขวา 30% โดยการปรับภาพนั้นจะใช้ โดยใช้เพื่อที่จะทำให้ Model ของเราสามารถแยกภาพโดยเห็นภาพที่ไม่เต็ม เพื่อที่จะทำการสอนมันและเพิ่มประสิทธิภาพของ Model โดยใน 1 รอบ นั้น จะมีทั้งหมด 42 รอบ ของ Split Test ด้วย (ใช้ Scikit Learn Model Selection Library) โดยเราจะทำแบบนี้ทั้งหมด อีก 40 รอบ หลังจากนั้นเราจะวัดค่า Train Accuracy , Train Loss , Validation Accuracy ,Validation Loss โดยใน 1 รอบของ 40 รอบ นั้นจะมี ค่า Accuracy , Loss พวกนี้ออกมา (Keras Library) และ ทำการนำแต่ละตัวแปรมาบวกกันทั้งหมด แล้วหารจำนวนรอบ เพื่อที่จะได้ค่า ทั้ง 4 ของ Model โดยค่า

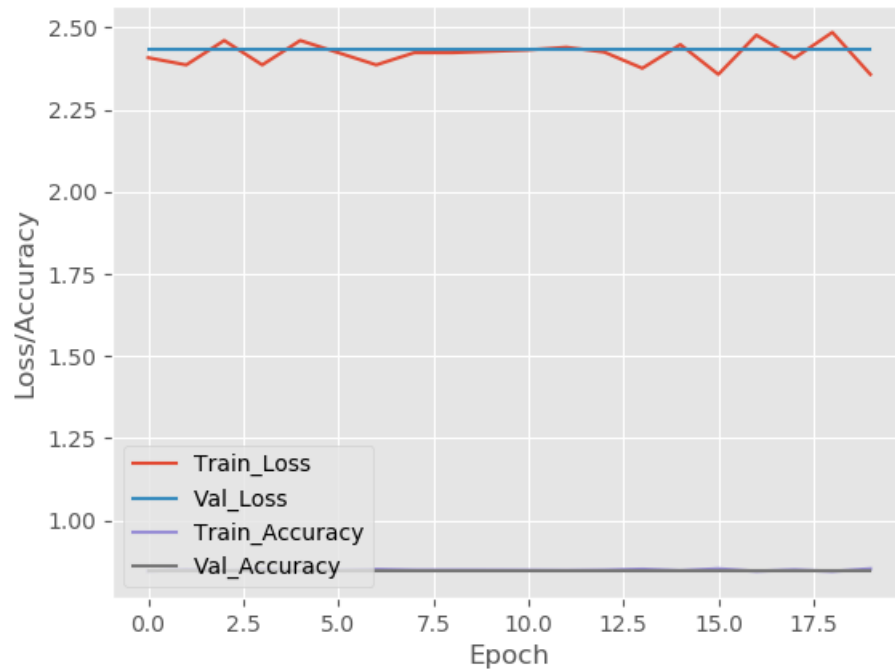
1. Train Accuracy นั้นจะมาจากการนำค่าของ Model ที่ ทำการ Prediction ของ ผัง Train ออกมาว่าถูกทั้งหมดกี่ % (ใน 3900 ภาพ) ยิ่งมากยิ่งดี
2. Train Loss จะเป็นการวิเคราะห์การเรียนรู้ของ Model ใน Trainset ที่ผิดพลาด ยิ่งน้อยยิ่งดี
3. Validation Accuracy นั้น จะมาจากการนำค่าของ Model ที่ ทำการ Prediction ของผัง Test ออกมาว่าถูกทั้งหมดกี่ % (ใน 1300 ภาพ) ยิ่งมากยิ่งดี
4. Validation Loss จะเป็นการวิเคราะห์การเรียนรู้ของ Model ใน Test Set ที่ผิดพลาด ยิ่งน้อยยิ่งดี

ขั้นตอนในการ Setup Model นั้น

ถ้าหาก Learning Rate สูง Epoch ควรจะน้อยควรใช้สำหรับภาพที่มีหลากหลาย Label และมีความจำเพาะสำหรับภาพสูง จะใช้ Dataset สำหรับการทำ แต่ละ Label น้อยเช่น แยกระหว่าง ประตูสีฟ้า ประตูสีแดง ประตูสีน้ำเงิน

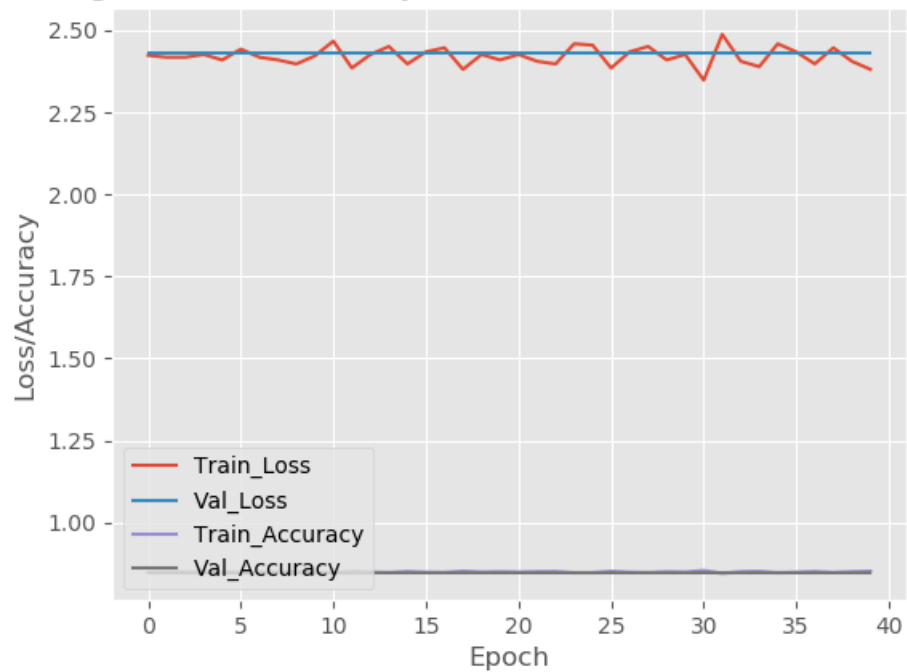
จากรูปที่ 4.8 และ 4.9 จะเห็นได้ว่า ค่า Train Loss และ ค่า Validation Loss จะสูงมาก สูงเกินกว่าค่า Accuracy เนื่องจากเราปรับค่า Learning Rate ให้สูงถึง 0.01 จนทำให้ Model ของเราเข้าใจว่าสิ่งนี้เป็นสิ่งที่ถูกต้อง ทำให้เห็นว่า โมเดลแบบนี้ไม่สามารถใช้งานได้โดยสามารถดูได้จากกราฟที่แสดงถึงค่าความแตกต่างที่มีค่า Loss ถึง 2.5 ทั้ง Train และ Validation

Training Loss and Accuracy for LEN detection HIGH LEARNIG LA⁻

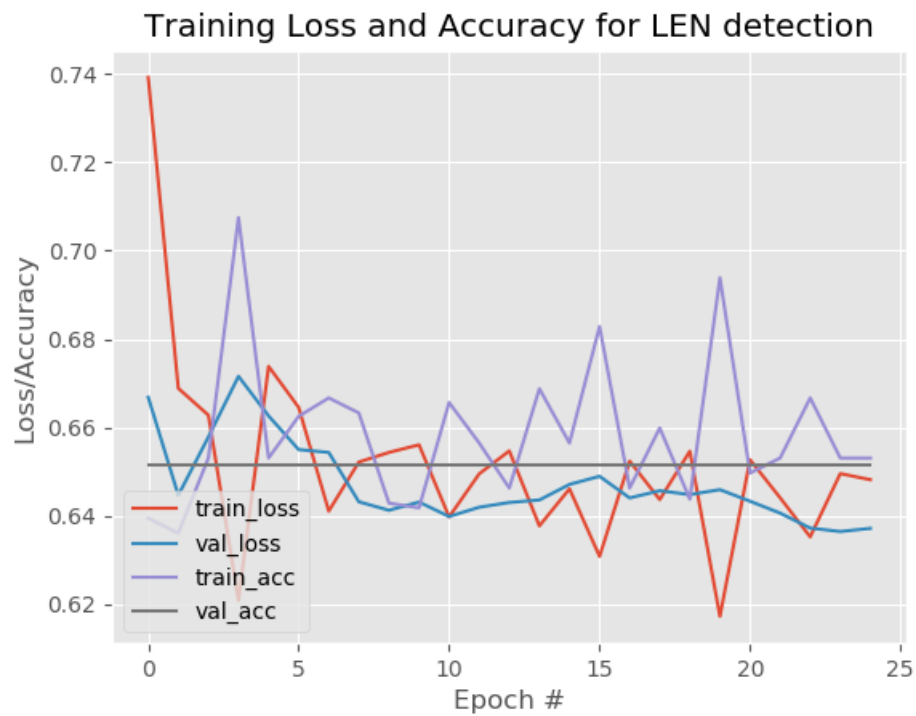


รูปที่ 4.8 ภาพแสดงประสิทธิภาพของ Model ที่มีค่า Learning Rate สูง โดยมีค่า Epoch ที่ 20

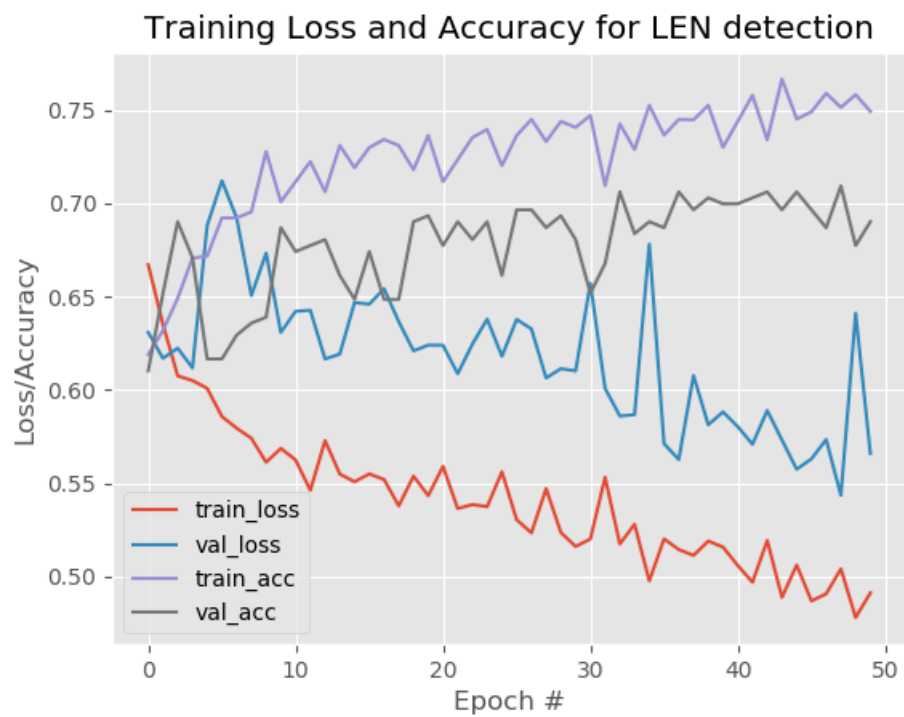
Training Loss and Accuracy for LEN detection HIGH LEARNIG LA⁻



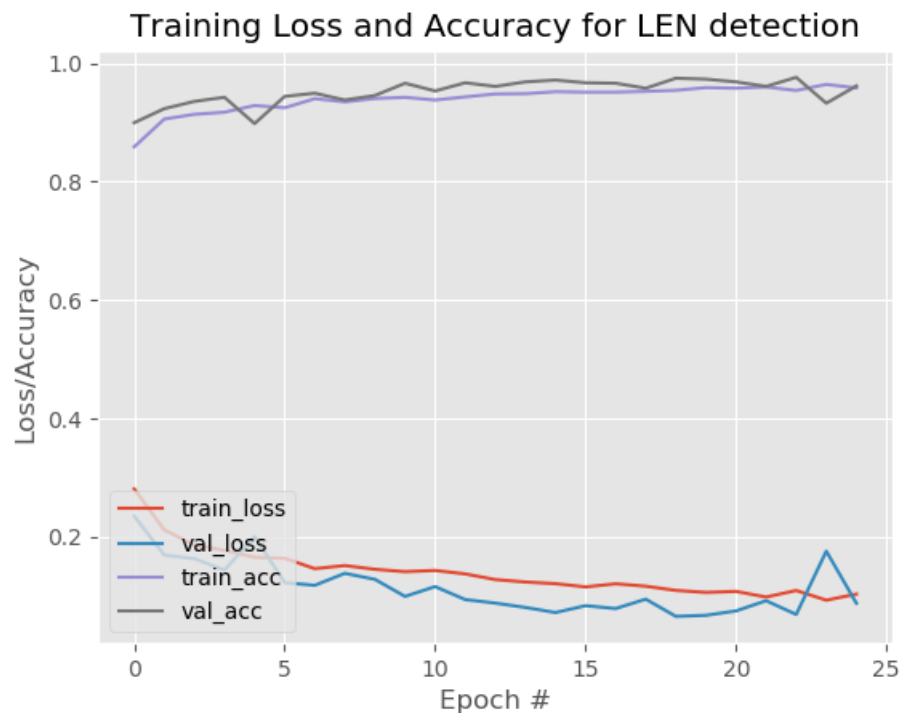
รูปที่ 4.9 ภาพแสดงประสิทธิภาพของ Model ที่มีค่า Learning Rate สูง โดยมีค่า Epoch ที่ 40



รูปที่ 4.10 ภาพแสดงประสิทธิภาพของ Model โดยที่ยังไม่ผ่านการปรับ Model และยังไม่ผ่านการทำ
Image Preprocessing



รูปที่ 4.11 กราฟแสดงประสิทธิภาพของ Model โดยที่ยังผ่านการปรับ Model แต่ยังไม่ผ่านการทำ
Image Preprocessing



รูปที่ 4.12 กราฟแสดงประสิทธิภาพของ Model โดยที่ยังผ่านการปรับ Model และผ่านการทำ Image Preprocessing

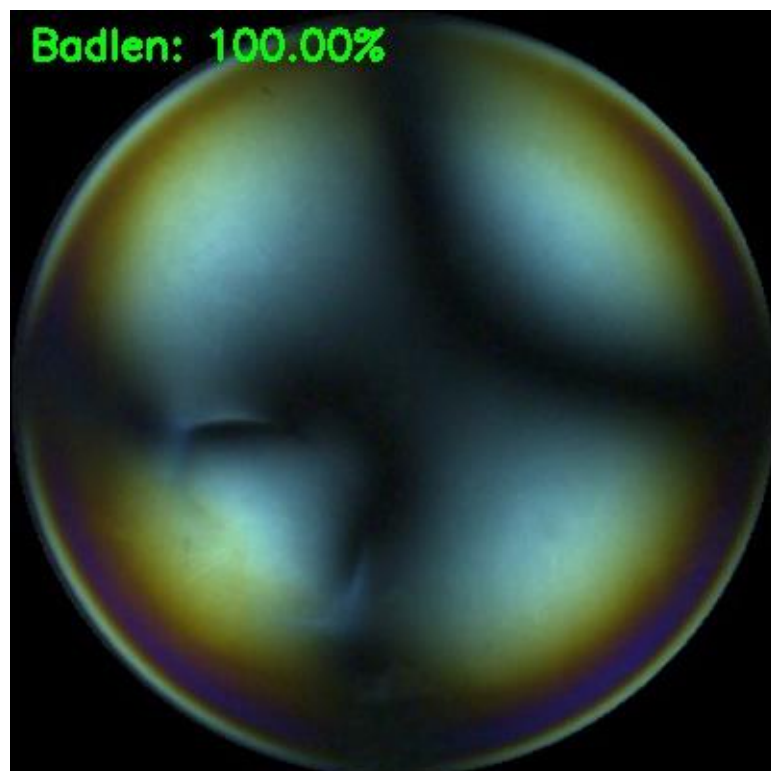
โดยในการทำขั้นตอนนี้จะทำให้สรุปได้ว่า ภาพที่ได้ จากการทำ Image Preprocessing มีประสิทธิภาพมากกว่าภาพที่ยังไม่ผ่านขบวนการทำ Image Preprocessing หลังจากที่ได้ Model ที่มีความแม่นยำถึง 97% เพื่อใช้งานต่อไป

4.2.3 Test Model

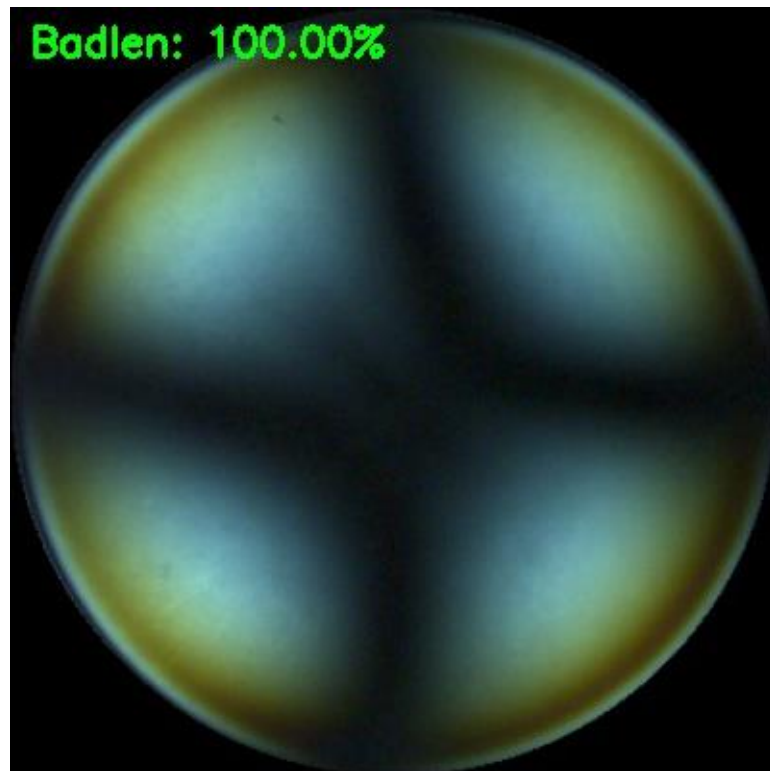
โดยในขั้นตอนนี้จะสามารถนำ Model ที่ได้จากการ Train แล้วไป Test จริงด้วยภาพที่ไม่อยู่ใน 5200 ภาพ (ไม่อยู่ใน Train , Test Model ด้านบน) เป็นภาพ Dataset ใหม่ที่นำมาสำหรับ Test Model เมื่อเจอสถานการณ์จริงเพื่อทดสอบประสิทธิภาพจริงเมื่อนำไปใช้งาน



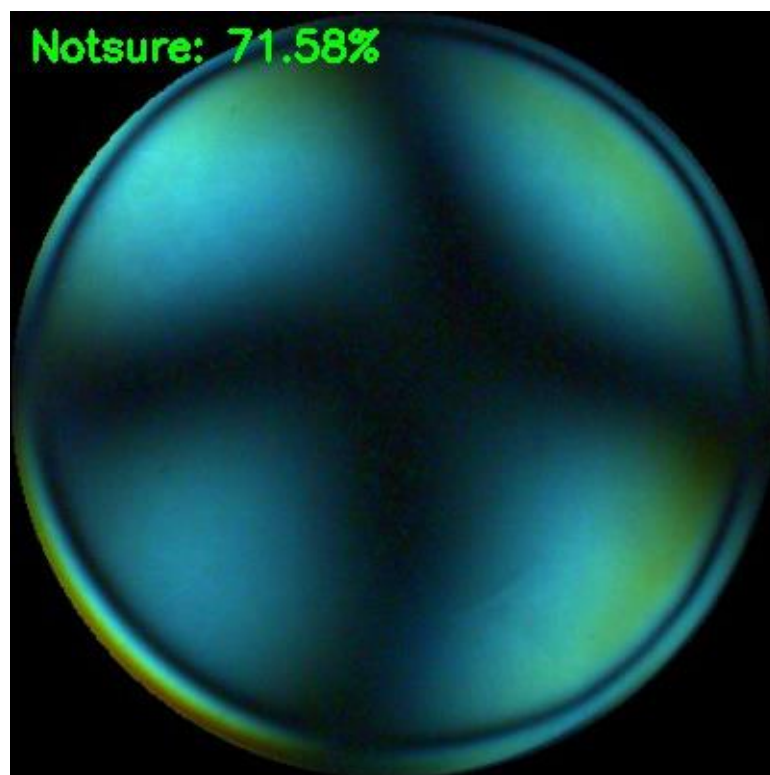
รูปที่ 4.13 ภาพที่ได้สำหรับเลนส์ที่ไม่ได้คุณภาพแบบมีเส้น (41) หลังนำไปใช้งาน



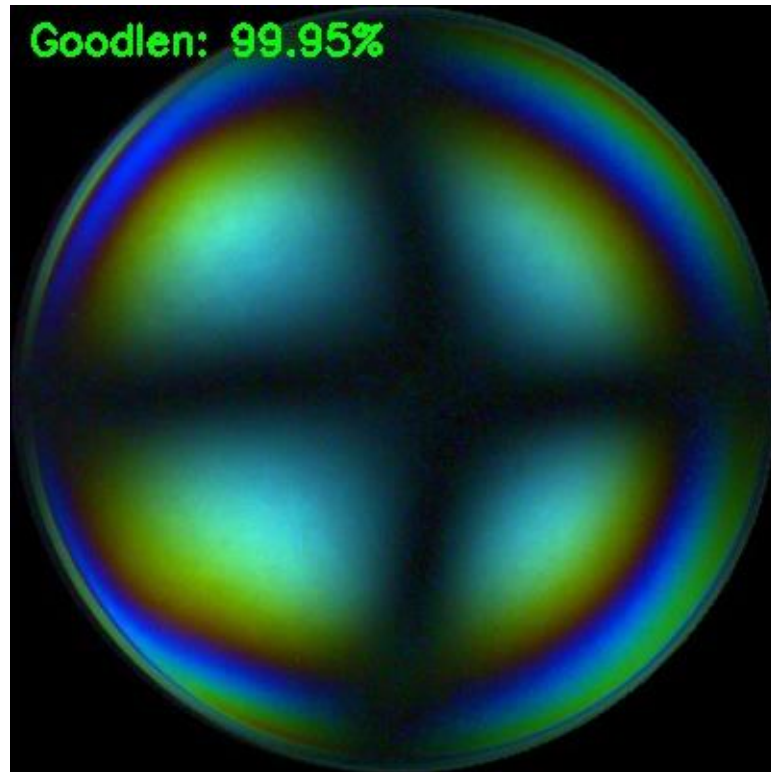
รูปที่ 4.14 ภาพที่ได้สำหรับเลนส์ที่ไม่ได้คุณภาพแบบวงกลม (40) หลังนำไปใช้งาน



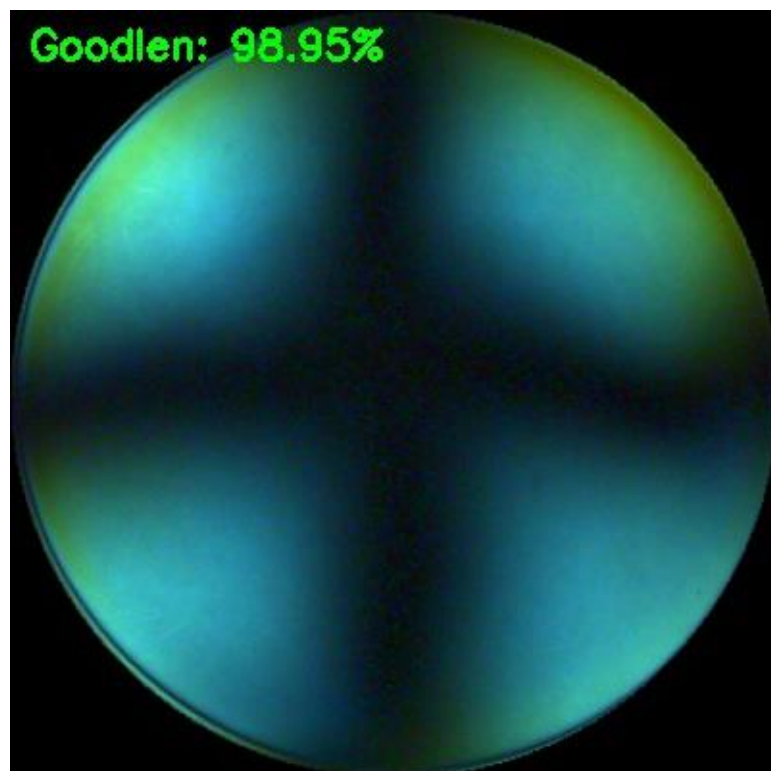
รูปที่ 4.15 ภาพที่ได้สำหรับเลนส์ที่ไม่ได้คุณภาพแบบมีเส้น (41) หลังนำไปใช้งาน โดยเป็นเส้นที่มองเห็นด้วยตาได้ยาก



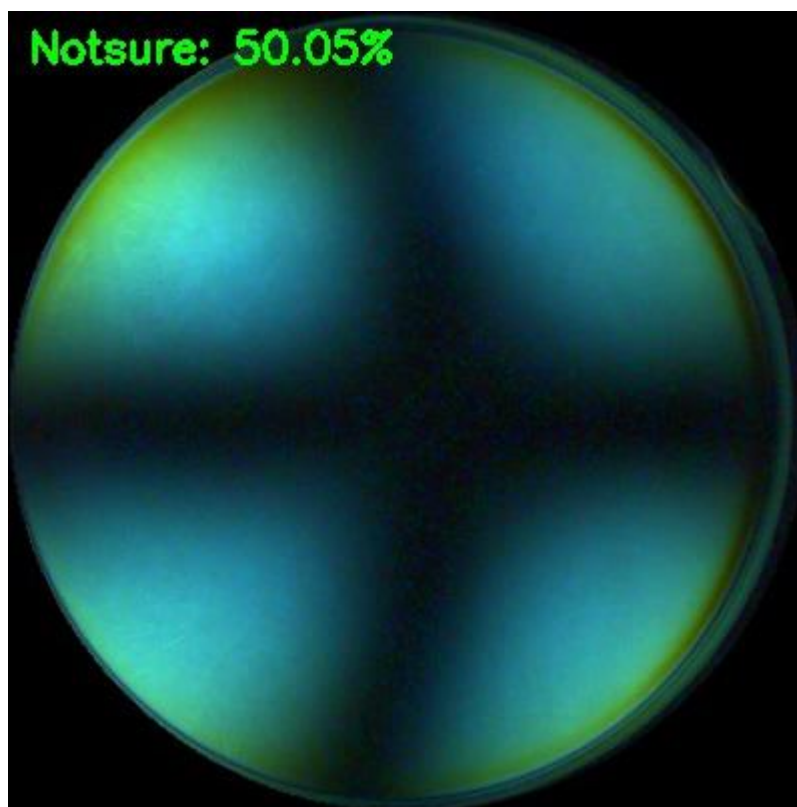
รูปที่ 4.16 ภาพที่ได้สำหรับเลนส์ที่ไม่ได้คุณภาพโดย Model ไม่สามารถแยกได้ว่า เป็นเลนส์ที่มีคุณภาพ หรือ ไม่มีคุณภาพแต่ในภาพเป็นเลนส์ที่ไม่มีคุณภาพแบบวงกลม (40)



รูปที่ 4.17 ภาพที่ได้สำหรับเลนส์ที่ได้คุณภาพหลังนำไปใช้งาน



รูปที่ 4.18 ภาพที่ได้สำหรับเลนส์ที่ได้คุณภาพหลังนำไปใช้งาน



รูปที่ 4.19 ภาพที่ได้สำหรับเลนส์ที่ได้คุณภาพหลังนำไปใช้งานโดยไม่สามารถแยกได้ว่าเป็นเลนส์ที่มีคุณภาพหรือไม่มีคุณภาพ

ประเภทเลนส์	ถูก	ผิด	ไม่สามารถระบุได้	ค่าความแม่นยำ
เลนส์ที่ได้คุณภาพ	176	3	21	94.72%
เลนส์ที่ไม่ได้คุณภาพ แบบมีเส้น(41)	499	0	1	100.00%
เลนส์ที่ไม่ได้คุณภาพ แบบมีวงกลม(40)	498	0	2	100.00%
รวม	1173	3	24	99.99%

ตารางที่ 4.1 ตารางแสดงค่าความแม่นยำของ Model หลังทดสอบประสิทธิภาพจริง

ในการทดลองนี้ ได้ทดลองจำนวน 1200 ภาพ โดยแบ่งเป็นเลนส์ที่มีคุณภาพ 200 ภาพ และเลนส์ที่ไม่มีคุณภาพแบบมีวงกลม (40) 500 ภาพ และ เลนส์ที่ไม่ได้คุณภาพแบบมีมีเส้น (41) 500 ภาพ

4.3 สรุป

ในการทดลองทั้งหมดในบทที่ 4 สามารถสรุปได้ดังนี้

การทดลองที่ 1 ในการทดลองขั้นตอนการเตรียมภาพต้นแบบเพื่อใช้ในการตรวจสอบคุณภาพของเลนส์ ภาพที่ได้หลังจากการทำการทดลองนี้มีประสิทธิภาพมากขึ้นเมื่อเทียบกับภาพต้นแบบ

การทดลองที่ 2 ในการทดลองการเรียนรู้ด้วยเครื่องมือ้นำการทดลองนี้ มาปรับเปลี่ยนค่าต่างๆ รวมถึงเพิ่มวิธีการทำงานของ Model จะเห็นได้ว่า ค่าความแม่นยำของ Model จะมีประสิทธิภาพมากยิ่งขึ้นเมื่อทำการ เตรียมภาพต้นแบบที่ดี

การทดลองที่ 3 การทดสอบความแม่นยำ พบว่าเมื่อนำ Model ที่ได้ มาใช้จริงจะพบว่าเลนส์ค่าความแม่นยำที่ได้จะมี 99.99%

บทที่ 5 สรุปและข้อเสนอแนะ

5.1 สรุปผล

สรุปผล จากการศึกษาและเก็บข้อมูลเลนส์ จะสามารถแบ่งได้ตามคุณสมบัติ เช่น รูปแบบ ค่าความโค้ง ค่าสายตาฯ ซึ่งในการตรวจสอบคุณภาพเลนส์ด้วยการเรียนรู้ของเครื่องนั้นจะสามารถตรวจสอบความสมบูรณ์ของเลนส์ด้วยการตรวจสอบด้วยเครื่องโพลาริเซชัน โดยเลนส์ที่ไม่สมบูรณ์จะสามารถแบ่งได้ 2 ประเภทคือ มีเส้นปรากฏบนเลนส์ (41) และมีวงกลมปรากฏบนเลนส์ (40) แต่ในกระบวนการผลิตจริงต้องใช้มนุษย์มาตรวจสอบ ซึ่งสามารถนำงานวิจัยเรื่องการตรวจสอบคุณภาพเลนส์ด้วยการเรียนรู้ด้วยเครื่องมาใช้ในการตัดสินใจแทนมนุษย์ จะทำให้บริษัทสามารถลดต้นทุนการผลิต และ เพิ่มประสิทธิภาพมากยิ่งขึ้น โดยแบ่งออกเป็น 3 การทดลอง มีผลลัพธ์แต่ละการทดลองดังนี้

การทดลองที่ 1 Image Preprocessing จะเห็นว่าภาพที่ได้หลังจากผ่านการทดลองนี้มีคุณภาพมากยิ่งขึ้นเมื่อนำภาพที่ได้ ไปใช้กับการทำ Machine Learning โดยสามารถดูได้จากภาพที่ได้ หรือ ค่าความแม่นยำของ Model จากค่าความแม่นยำ 66% เมื่อนำภาพที่ได้หลังจากการทำ Image Preprocessing แล้วมีค่าความแม่นยำถึง 97%

การทดลองที่ 2 การเรียนรู้ของเครื่อง โดย นำภาพที่ได้ ทั้งหมด 5200 ภาพ นำไป Train Model โดยเปรียบเทียบประสิทธิภาพจาก ค่า Train Accuracy , Train Loss , Validation Accuracy , Validation Loss โดยสนใจ Validation Accuracy เป็นหลักเพื่อที่ Model จะไม่ Overfit และหลังจากทดลองแล้ว ได้ค่าความแม่นยำถึง 97 %

การทดลองที่ 3 โดยการทดลองนี้จะทดสอบประสิทธิภาพของ Model เมื่อนำมาใช้งานจริง โดยทดสอบแล้วพบว่า เลนส์ที่ได้คุณภาพ จาก 200 ภาพ มี เลนส์ที่ทำนายผิด 3 ภาพ และ ไม่สามารถระบุได้ 21 ภาพ โดยภาพที่ไม่สามารถระบุได้เนื่องจาก เลนส์นั้นมีความคล้ายคลึงกับเลนส์ที่ไม่ได้คุณภาพมากเกินไปจนบางครั้ง คนก็ไม่สามารถระบุได้ จึงต้องนำไปตรวจสอบใหม่อีกครั้ง และ เลนส์ที่ไม่ได้คุณภาพแบบมีเส้น จาก 500 ภาพ มีเลนส์ที่ทำนายผิด 0 ภาพ และ ไม่สามารถระบุได้ 1 ภาพ และ เลนส์ที่ไม่ได้คุณภาพแบบมีวงกลม จาก 500 ภาพ มีเลนส์ที่ทำนายผิด 0 ภาพ และ ไม่สามารถระบุได้ 2 ภาพ โดย ค่า ความแม่นยำรวมของทั้งหมดคือ 99%

5.2 ข้อเสนอแนะ

ด้วยการตรวจสอบเลนส์ด้วยการเรียนรู้ของเครื่องนั้นยังไม่เป็นระบบอัตโนมัติทั้งหมดทำให้การตรวจสอบของคน ยังมี ความเร็วที่มากกว่า การนำเลนส์ไปวางบนเครื่องเพื่อตรวจสอบ ซึ่งสามารถพัฒนาต่อไปในอนาคตเพื่อที่จะทำให้ระบบตรวจสอบคุณภาพของเลนส์เป็นอัตโนมัติทั้งหมด

เอกสารอ้างอิง

- [1] Cross Validation, Available:
<https://www.kdnuggets.com/2018/01/training-test-sets-cross-validation.html>
- [2] Train Test Set, Available:
<https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>
- [3] Top 10 Programming, Available:
<https://medium.com/@dumpdatasci.th/machine-learning-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3%E0%B8%9E%E0%B8%A3%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%95%E0%B8%B1%E0%B8%A7%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%83%E0%B8%99%E0%B8%8A%E0%B8%B5%E0%B8%A7%E0%B8%B4%E0%B8%95%E0%B8%88%E0%B8%A3%E0%B8%B4%E0%B8%87-b3f8faf0b4da>
- [4] Train Test Split, Available:
<https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6>
- [5] Keras Model, Available:
<https://machinelearningmastery.com/evaluate-performance-deep-learning-models-keras/>
- [6] OpenCV, Available: https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html
- [7] ทัศนดา ถาวรกุล,2560, การตรวจสอบคุณภาพของเลนส์ ด้วยวิธีการประมวลผลภาพ, รายงานการวิจัย, สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ, มหาวิทยาลัยพระจอมเกล้าธนบุรี
- [8] นิเวศ จิระวิชิตชัย ,2557, การพัฒนาประสิทธิภาพแบบจำลองการจำแนกอารมณ์จากข้อความภาษาไทยโดยใช้เทคนิคปรับปรุงดัชนีของคำร่วมกับการเรียนรู้ของเครื่อง, รายงานการวิจัย, สาขาวิชาระบบสารสนเทศคอมพิวเตอร์, มหาวิทยาลัยศรีปทุม
- [9] N.E. Sahla,2018, A Deep Learning Prediction Model for Object Classification, Master thesis Mechanical, Maritime and Materials Engineering, TU Delft Bio Robotics, Delft University of Technology
- [10] วุฒิชัย บุญพุก และคณะ, การประยุกต์ใช้ระเบียบวิธีการเรียนรู้เชิงลึกเพื่อจำแนกประเภทการใช้ที่ดินด้านเกษตรกรรมในประเทศไทยโดยใช้ซอฟต์แวร์กาแฟ, รายงานการวิจัย,มหาวิทยาลัยการบินและอวกาศแห่งกรุงปักกิ่ง
- [11] Markus Teigen Pike,2559, Computer Vision and Deep Learning in Autonomous Drones, Masterthesis, Department of Computer Science, Norwegian University of Science and Technology

ประวัติผู้วิจัย

ชื่อ — สกุล	นายณภัทร นิธิโสภา
รหัสนักศึกษา	58340500060
วัน เดือน ปีเกิด	3 กรกฎาคม 2539
ประวัติการศึกษา	
ระดับมัธยมศึกษา	ประโยคมัธยมศึกษาตอนปลาย โรงเรียนอัสสัมชัญ ปีการศึกษา 2557
ระดับปริญญาตรี	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ สถาบันวิทยาการหุ่นยนต์ภาคสนาม ปีการศึกษา 2561
ทุนการศึกษา หรือทุนวิจัย	ทุนสนับสนุนแบบไม่เต็มจำนวน หลักสูตรปริญญาตรี สาขา วิศวกรรมหุ่นยนต์และระบบอัตโนมัติ

