



การปรับปรุงประสิทธิภาพการทำงานความเข้มข้นของเครื่อง
สเปกโตรโฟโตมิเตอร์แบบพกพาด้วยการเรียนรู้ของเครื่อง

นายกิตตินันท์ จันเพชร
นายณภัฏ จันทปัญญา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเคมี ภาควิชาวิศวกรรมเคมี

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CONCENTRATION PREDICTION USING MACHINE LEARNING
ENHANCED PORTABLE SPECTROPHOTOMETER

MISTER KITINAN JANPETCH
MISTER NAPHON CHANTHAPANYA

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE DEGREE OF BACHELOR OF ENGINEERING IN CHEMICAL
ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG


ACADEMIC YEAR 2023

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

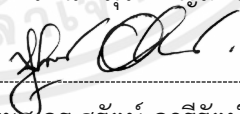
ปริญญาานิพนธ์เรื่อง การปรับปรุงประสิทธิภาพการทำนายความเข้มข้นของเครื่องสเปกโตรโฟโต
มิเตอร์แบบพกพาด้วยการเรียนรู้ของเครื่อง
โดย นายกิตตินันท์ จันเพชร รหัสนักศึกษา 63010078
นายณภนต์ จันทปัญญา รหัสนักศึกษา 63010490
ปริญญา วิศวกรรมศาสตร์บัณฑิต
สาขาวิชา วิศวกรรมเคมี
ปีการศึกษา 2566
อาจารย์ที่ปรึกษา รศ.ดร.อมตะ อนันต์พินิจวัฒนา
ผศ.ศิริพันธ์ มุรธาธัญลักษณ์

ปริญญาานิพนธ์นี้ได้รับการพิจารณาอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมเคมี

คณะกรรมการตรวจสอบปริญญาานิพนธ์


ประธานกรรมการ
(รศ.ดร.อมตะ อนันต์พินิจวัฒนา)


กรรมการ
(ผศ.ศิริพันธ์ มุรธาธัญลักษณ์)


กรรมการ
(ผศ.ดร.สุรัตน์ อารีรัตน์)

ปริญญาานิพนธ์เรื่อง	การปรับปรุงประสิทธิภาพการทำนายความเข้มข้นของเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพาด้วยการเรียนรู้ของเครื่อง		
โดย	นายกิตตินันท์	จันเพชร	รหัสนักศึกษา 63010078
	นายณภัฏ	จันทปัญญา	รหัสนักศึกษา 63010490
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมเคมี		
ปีการศึกษา	2566		
อาจารย์ที่ปรึกษา	รศ.ดร.อมตะ อนันต์พิณีวัฒนา		
	ผศ.ศิริพันธ์ มุรธาธัญลักษณ์		

บทคัดย่อ

การวิเคราะห์ความเข้มข้นของสารกำจัดศัตรูพืชต้องใช้อุปกรณ์ที่มีราคาสูง เช่น เครื่องโครมาโทกราฟีของเหลวสมรรถนะสูงหรือสเปกโตรโฟโตมิเตอร์ การใช้สเปกโตรโฟโตมิเตอร์แบบพกพาจึงเป็นทางเลือกที่มีราคาประหยัดและพกพาได้ง่าย จากผลงานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” โดย Chotika และคณะพัฒนาสเปกโตรโฟโตมิเตอร์ชนิดพกพาที่ให้ค่าความคลาดเคลื่อนสูงในการทำนายความเข้มข้นของโพแทสเซียมเปอร์แมงกาเนต ($KMnO_4$) และชุดทดลองคาร์โบซิลแพน ($C_{20}H_{32}N_2O_3S$) เป็นร้อยละ 22.88 และ 21.88 ตามลำดับ เมื่อเทียบกับเครื่องสเปกโตรโฟโตมิเตอร์แบบทั่วไปที่ร้อยละ 4.17 ดังนั้นงานวิจัยนี้จะมุ่งเน้นการปรับปรุงตรรกะการวิเคราะห์ภาพ และการประยุกต์ใช้การเรียนรู้ของเครื่องเพื่อลดค่าความคลาดเคลื่อนจากการทำนายความเข้มข้นของภาพจากสเปกโตรโฟโตมิเตอร์พกพาโดยแบ่งกระบวนการทำงานเป็นสองส่วนหลัก คือ 1) การเตรียมและแปลงรูปภาพเป็นข้อมูลตัวเลข 2) การประยุกต์ใช้การเรียนรู้ของเครื่องเพื่อทำนายความเข้มข้น จากข้อมูลภาพที่ผ่านการประมวลผลแล้ว ผลการทำนายความเข้มข้นของโพแทสเซียมเปอร์แมงกาเนต ($KMnO_4$) และคาร์โบซิลแพน ($C_{20}H_{32}N_2O_3S$) ด้วยชุดข้อมูลเดิม พบว่าความคลาดเคลื่อนลดลงเหลือเพียงร้อยละ 6.03 และ 18.52 ตามลำดับ นอกจากนี้มีการทดลองเพิ่มเติม เกลือ, น้ำตาล และสารผสมเกลือ-น้ำตาล ได้เป็นร้อยละ 32.34, 21.65 และ 39.77 ตามลำดับ

คำสำคัญ : สารกำจัดศัตรูพืช, คาร์บาเมต, คาร์โบซิลแพน, สเปกโตรโฟโตมิเตอร์, การเรียนรู้ของเครื่อง, การเตรียมภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Concentration prediction using machine learning enhanced portable spectrophotometer.		
By	Kittinan	Janpetch	Student ID 63010078
	Naphon	Chanthapanya	Student ID 63010490
Degree	Bachelor of Engineering		
Program	Chemical Engineering		
Academic Year	2566		
Thesis Advisor	Assoc. Prof. Dr. Amata Anantpinijwatna		
	Asst. Prof. Siripan Murathathunyaluk		

Abstract

The analysis of pesticide concentrations requires expensive equipment such as high-performance liquid chromatography or spectrophotometers. Using a portable spectrophotometer is a more economical and portable alternative. From the research work "Development of a Device for Analyzing Carbamate Pesticides Using Light Absorption Principle and Digital Image Analysis" by Chotika et al. who developed a portable spectrophotometer with prediction errors of 22.88% and 21.88% for potassium permanganate (KMnO_4) and carbosulfan ($\text{C}_{20}\text{H}_{32}\text{N}_2\text{O}_3\text{S}$) respectively, which are still high errors compared to a conventional spectrophotometer at 4.17%. Therefore, this research aims to improve the image analysis logic and apply machine learning to reduce the prediction error of concentration from portable spectrophotometer images. The work is divided into two main processes: 1) image preparation and conversion to numerical data, and 2) applying machine learning to predict concentration from the processed image data. The predicted concentrations of potassium permanganate (KMnO_4) and carbosulfan ($\text{C}_{20}\text{H}_{32}\text{N}_2\text{O}_3\text{S}$) using the same dataset showed reduced errors of 6.03% and 18.52% respectively. Additionally, further experiments were conducted with salt, sugar, and salt-sugar mixtures, resulting in errors of 32.34%, 21.65% and 39.77% respectively.

Keywords: pesticide, carbamate, carbosulfan, spectrophotometer, Machine learning, image processing, Spectral band

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง “การปรับปรุงประสิทธิภาพการทำนายความเข้มข้นของเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพาด้วยการเรียนรู้ของเครื่อง” สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากการได้รับความกรุณา และความช่วยเหลือจาก รศ.ดร.อมตะ อนันต์พิณจิวัฒนา และ ผศ.ศิริพันธ์ มุรธาธัญลักษณ์ อาจารย์ที่ปรึกษางานวิจัยที่คอยให้คำแนะนำและคำปรึกษาในทุกขั้นตอนการทำวิจัย ตลอดจนความรู้ทางวิชาการที่เป็นประโยชน์ต่องานวิจัยให้สำเร็จบรรลุตามวัตถุประสงค์

ขอขอบคุณบุคลากร คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมเคมี สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่อำนวยความสะดวกให้สถานที่ในการศึกษาและค้นคว้าทำการทดลอง ทั้งด้านอุปกรณ์การทดลอง สารเคมีและสาธารถุบัติและขอขอบคุณแหล่งค้นคว้าข้อมูลสำคัญที่เป็นประโยชน์แก่งานวิจัย ทั้งหนังสือ ตัวอย่างงานวิจัยต่างๆ เว็บไซต์ และบทความ วารสารทางวิชาการ ที่คณะผู้วิจัยได้นำมาใช้ประกอบการทำเล่มวิจัยฉบับนี้

สุดท้ายนี้ ทางคณะผู้จัดทำวิจัย ขอขอบคุณผู้มีพระคุณที่ช่วยเหลือและสนับสนุนทุนการวิจัยอันมีค่าแก่งานวิจัยนี้ และหวังว่างานวิจัยเล่มนี้จะเป็นประโยชน์และเป็นแนวทางให้กับผู้ที่สนใจศึกษาและหากผิดพลาดประการใดทางผู้วิจัยขออภัย มา ณ ที่นี้ด้วยพร้อมน้อมรับคำแก้ไขทุกประการ

กิตตินันท์ จันเพชร

นภนต์ จันทปัญญา

ผู้วิจัย

สารบัญ

หน้า

บทคัดย่อ.....	ii
Abstract.....	iii
กิตติกรรมประกาศ.....	iv
สารบัญ.....	v
สารบัญรูป.....	ix
สารบัญตาราง.....	xiii
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	2
1.3 ขอบเขตงานวิจัย.....	2
1.4 ประโยชน์ที่ได้รับ.....	2
บทที่ 2 ทฤษฎี และงานวิจัยที่เกี่ยวข้อง.....	3
2.1 วิธีการวิเคราะห์ความเข้มข้นของสารละลาย.....	3
2.1.1 สเปกโตรโฟโตมิเตอร์.....	3
2.1.1.1 กฎของเบียร์และแลมเบิร์ต (Beer-Lambert law).....	3
2.1.2 ความแตกต่างระหว่างยูวี-วิสิเบิล สเปกโตรโฟโตมิเตอร์ (UV-VIS Spectrophotometer) กับ สเปกโตรโฟโตมิเตอร์แบบพกพา.....	4
2.2 ระบบสี (Color Space).....	5
2.2.1 RGB.....	5
2.2.2 LAB.....	5
2.3 ภาษาคอมพิวเตอร์ (Computer language).....	6
2.3.1 ภาษาไพทอน (Python).....	6
2.3.2 Pandas.....	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.3.3 NumPy.....	6
2.3.4 OpenCV.....	6
2.3.5 Scikit-Learn.....	7
2.3 ทฤษฎีการเตรียมรูปภาพก่อนการวิเคราะห์ (Image Pre-processing).....	7
2.3.1 การขยาย (Dilation).....	7
2.3.2 การกัดกร่อน (Erosion).....	7
2.3.3 การปิด (Closing).....	7
2.3.4 การเปิด (Opening).....	8
2.3.5 การเบลลอภาพด้วยฟังก์ชันเกาส์เซียน (Gaussian blur).....	8
2.3.6 การจับรูปทรงของภาพ (Contour Detection).....	8
2.4 การเรียนรู้ของเครื่อง (Machine Learning).....	9
2.5 อัลกอริทึมการเรียนรู้ของเครื่อง.....	10
2.5.1 Random Forest (RF).....	10
2.5.2 Support Vector Regression (SVR).....	11
2.5.3 Multilayer Perceptron (MLP).....	12
2.5.4 Extreme Gradient Boosting (XGBoost).....	13
2.6 การเพิ่มประสิทธิภาพ.....	13
2.6.1 MAPE (Mean Absolute Percentage Error).....	13
2.6.2 MAE (Mean Absolute Error).....	13
2.6.3 R-square (Coefficient of Determination).....	14
2.6.4 MSE (Mean Squared Error):.....	14
2.6.5 Cross Validation.....	14
2.7 งานวิจัยที่เกี่ยวข้อง.....	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.7.1	งานวิจัยตรวจสอบปริมาณฟอสฟอรัสในดินด้วยสเปกโตรโฟโตมิเตอร์ต้นทุนต่ำและการเรียนรู้ของเครื่อง	15
2.7.2	งานวิจัยการวัดปริมาณความเข้มข้นของอสุจิของมนุษย์โดยใช้เครื่องสเปกโตรโฟโตมิเตอร์แบบอาศัยการเรียนรู้ของเครื่อง.....	16
บทที่ 3 วิธีการดำเนินงาน		17
3.1	การเตรียมรูปภาพและการแปลงรูปภาพเป็นข้อมูลตัวเลข	17
3.1.1	งานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล”	17
3.1.1.1	อ่านรูปจากโฟลเดอร์	19
3.1.1.2	เลือกพื้นที่แถบสีที่ต้องการวิเคราะห์	20
3.1.1.3	เบลอรูปภาพ.....	20
3.1.1.4	การแปลงรูปภาพเป็นข้อมูลตัวเลข	20
3.1.2	งานวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต”	21
3.1.2.1	อ่านรูปจากโฟลเดอร์	24
3.1.2.2	การเลือกพื้นที่แถบสี.....	24
3.1.2.3	หาพิกัดมุมของกรอบสี่เหลี่ยมแถบสีแสงฟุ้งที่ไม่มีสารละลาย	25
3.1.2.4	เฉลี่ยความสว่างของแถบสีแสงฟุ้งที่ไม่มีสารละลาย	26
3.1.2.5	แปลงพิกัดมุมที่มี 8 พารามิเตอร์ ให้เหลือ 5 พารามิเตอร์.....	26
3.1.2.6	ย้ายจุดกึ่งกลางของสี่เหลี่ยมแถบสีแสงฟุ้งที่ไม่มีสารละลาย ไปอยู่ในฝั่งที่มีสารละลาย 27	
3.1.2.7	สร้างกรอบสี่เหลี่ยมใหม่ ปรับเปลี่ยนมุมมองภาพและเลื่อนตำแหน่ง	27
3.1.2.8	เก็บค่าชื่อของรูปภาพ และแปลงข้อมูลรูปภาพเป็นตัวเลข.....	28
3.2	การเรียนรู้ของเครื่องและการเลือกใช้โมเดล.....	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2.1	การเลือกชุดข้อมูลตัวแปรต้นด้วย SelectKBest.....	29
3.2.2	การสร้างความเท่าเทียมให้กับโมเดลด้วย K-fold.....	29
3.2.3	การทำนายความเข้มข้นด้วยการเรียนรู้ของเครื่อง.....	29
บทที่ 4	ผลการทดลองและวิเคราะห์ผลการทดลอง.....	31
4.1	ผลการทดลองข้อมูลจากงานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืช กลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล”	31
4.1.1	ผลจากกระบวนการเตรียมรูปภาพ.....	31
4.1.2	ผลจากการประยุกต์ใช้การเรียนรู้ของเครื่อง.....	32
4.2	ผลการทดลองข้อมูลจากงานวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพา ราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต”	34
4.2.1	ชุดการทดลองสารละลายคาร์โบซิลแฟน.....	34
4.2.1.1	ผลจากกระบวนการเตรียมรูปภาพ.....	34
4.2.1.2	ผลจากการประยุกต์ใช้การเรียนรู้ของเครื่อง.....	34
4.2.2	ชุดการทดลองสารละลายเกลือ, น้ำตาล และ เกลือ-น้ำตาล.....	37
4.2.2.1	ผลจากกระบวนการเตรียมรูปภาพ.....	37
4.2.2.2	ผลจากการประยุกต์ใช้การเรียนรู้ของเครื่อง.....	38
4.3	วิเคราะห์ผลการทดลอง	48
บทที่ 5	สรุปผลการทดลองและข้อเสนอแนะ	51
5.1	สรุปผลการทดลอง	51
5.2	ข้อเสนอแนะ	52
เอกสารอ้างอิง		53
ภาคผนวก ก.....		57

สารบัญรูป

รูปที่	หน้า
2.1 ระบบสี RGB.....	5
2.2 ระบบสี LAB.....	6
2.3 รูปภาพแสดงตัวอย่างการดำเนินการปิดทั้งก่อนและหลังการดำเนินการ.....	7
2.4 รูปภาพแสดงตัวอย่างการดำเนินการเปิดก่อนและหลังการดำเนินการ.....	8
2.5 ชนิดของการเรียนรู้ของเครื่อง.....	9
2.6 Random Forest (RF).....	11
2.7 Support Vector Regression (SVR).....	12
2.8 Multilayer Perceptron (MLP).....	12
2.9 การ Cross Validation แบบ k-fold.....	15
2.10 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set จากงานวิจัย ตรวจสอบปริมาณฟอสฟอรัสในดินด้วยสเปกโตรโฟโตมิเตอร์ต้นทุนต่ำและการเรียนรู้ของเครื่อง	16
2.11 อุปกรณ์ที่ใช้ในการวิเคราะห์และเก็บค่าความเข้มข้นของอสุจิ	16
3.1 ฟังก์ชันการดำเนินงาน.....	17
3.2 ตัวอย่างรูปจากชุดการทดลอง (ก)โพแทสเซียมเปอร์แมงกาเนต และ(ข)คาร์โบซิลแฟน	18
3.3 แผนภาพแสดงขั้นตอนการเตรียมรูปภาพและแปลงเป็นตัวเลขชุดการทดลอง “การพัฒนา อุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการ วิเคราะห์ภาพดิจิทัล”	19
3.4 ชุดการทดลอง KMnO ₄ “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดย ใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” หลังการตัดรูปภาพ	20
3.5 ชุดการทดลอง KMnO ₄ “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดย ใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” หลังการเบลอรูปภาพ	20
3.6 ตัวอย่างรูปจากชุดการทดลอง (ก)คาร์โบซิลแฟน (ข)เกลือ (ค)น้ำตาล และ (ง)เกลือ-น้ำตาล	22
3.7 แผนภาพแสดงขั้นตอนการเตรียมรูปภาพและแปลงเป็นตัวเลขชุดการทดลอง “การพัฒนา อุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภท คาร์บาเมต”	23
3.8 กราฟแสดงค่า L เฉลี่ยในแต่ละแถวของรูปภาพ.....	24

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.9 กราฟแสดงค่า L เฉลี่ยในแต่ละหลักของรูปภาพ.....	24
3.10 รูปภาพที่ผ่านการเลือกพื้นที่แถบสีอ้างอิง.....	25
3.11 รูปภาพหลังจากเปลี่ยนระบบสีจาก RGB เป็น Greyscale.....	25
3.12 รูปภาพหลังจากตั้งเกณฑ์ของค่า Greyscale.....	25
3.13 รูปภาพหลังจากใช้เทคนิคสัณฐานวิทยาแบบเปิด.....	26
3.14 รูปภาพหลังจากใช้เทคนิคสัณฐานวิทยาแบบปิด.....	26
3.15 รูปภาพแสดงตำแหน่งกรอบสี่เหลี่ยมที่แถบสีอ้างอิง.....	27
3.16 รูปภาพแสดงตำแหน่งกรอบสี่เหลี่ยมจากแถบสีอ้างอิงและตำแหน่งกรอบสี่เหลี่ยมที่เกิดจากการ ฉาย.....	27
3.17 ตัวอย่างภาพหลังการปรับเปลี่ยนมุมมองชุดการทดลองคาร์โบซิลแฟน “การพัฒนาอุปกรณ์สเปก โตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต”	28
3.18 ฟังงานการเรียนรู้ของเครื่องและการเลือกใช้โมเดล.....	29
4.1 รูปภาพสารละลายโพแทสเซียมเปอร์แมงกาเนต 5 ppm ก่อนกระบวนการ.....	31
4.2 รูปภาพสารละลายโพแทสเซียมเปอร์แมงกาเนต 5 ppm หลังกระบวนการ.....	31
4.3 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูล โพแทสเซียมเปอร์แมงกาเนตจากโมเดล SVR.....	32
4.4 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูล โพแทสเซียมเปอร์แมงกาเนตจากโมเดล RF.....	32
4.5 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูล โพแทสเซียมเปอร์แมงกาเนตจากโมเดล MLP.....	33
4.6 รูปภาพสารละลายคาร์โบซิลแฟน 300 ppm ก่อนกระบวนการ.....	34
4.7 รูปภาพสารละลายคาร์โบซิลแฟน 300 ppm หลังกระบวนการ.....	34
4.8 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูล คาร์โบซิลแฟนจากโมเดล SVR.....	35
4.9 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูล คาร์โบซิลแฟนจากโมเดล RF.....	35

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.10 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลคาร์โบซัลเฟนจากโมเดล MLP	36
4.11 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลคาร์โบซัลเฟนจากโมเดล XGBoost	36
4.12 สารละลายเกลือ 5 %w/v	37
4.13 สารละลายน้ำตาล 5 %w/v	37
4.14 สารละลายเกลือ 5 %w/v และ น้ำตาล 2.5 %w/v	38
4.15 รูปภาพสารละลายเกลือ 5 %w/v หลังกระบวนการ.....	38
4.16 รูปภาพสารละลายน้ำตาล 5 %w/v หลังกระบวนการ.....	38
4.17 รูปภาพสารละลายเกลือ 5 %w/v และ น้ำตาล 2.5 %w/v หลังกระบวนการ.....	38
4.18 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือจากโมเดล SVR.....	39
4.19 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือจากโมเดล RF	39
4.20 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือจากโมเดล MLP	40
4.21 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือจากโมเดล XGBoost	40
4.22 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลน้ำตาลจากโมเดล SVR.....	41
4.23 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลน้ำตาลจากโมเดล RF.....	42
4.24 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลน้ำตาลจากโมเดล MLP	42
4.25 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลน้ำตาลจากโมเดล XGBoost.....	43
4.26 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือ-น้ำตาลจากโมเดล SVR.....	44

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.27 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือ-น้ำตาลจากโมเดล RF	45
4.28 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือ-น้ำตาลจากโมเดล MLP.....	45
4.29 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือ-น้ำตาลจากโมเดล XGBoost.....	46
4.30 แถบสีอ้างอิงคาร์โบซิลแพนความเข้มข้น 0 ppm (ก) และ 300 ppm (ข).....	48
4.31 ค่าความสว่างของแถบสีอ้างอิงจากการทดลองสารละลายคาร์โบซิลแพน.....	49
4.32 ค่าความสว่างของแถบสีอ้างอิงจากการทดลองสารละลายคาร์โบซิลแพนหลังการใช้ฟังก์ชัน.....	49

สารบัญตาราง

ตารางที่	หน้า
3.1 สรุปชุดข้อมูลของงานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมต โดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล”	18
3.2 สรุปชุดข้อมูลของงานวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัด สำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต”	21
3.3 การปรับ Hyperparameter ของแต่ละโมเดล	30
4.1 Evaluation metrics ของแต่ละโมเดล	33
4.2 Evaluation metrics ของแต่ละโมเดล	37
4.3 Evaluation metrics ของแต่ละโมเดล	41
4.4 Evaluation metrics ของแต่ละโมเดล	43
4.5 Evaluation Metrics ของแต่ละโมเดล	46
4.6 Hyperparameter ที่ใช้ในแต่ละชุดการทดลอง	47
4.7 สรุปโมเดลที่ดีที่สุดและค่าความคาดเคลื่อนของแต่ละการทดลอง	48

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ปัจจุบันมีการใช้งานสารเคมีกำจัดศัตรูพืชกันอย่างแพร่หลายเพื่อเพิ่มผลผลิตทางการเกษตรตามความต้องการของผู้บริโภค สารเคมีกำจัดศัตรูพืชเป็นที่รู้จักกันประกอบด้วยกลุ่มของสารกำจัดแมลง (Insecticides) สารกำจัดหนูและสัตว์ฟันแทะ (Rodenticides) และ สารกำจัดเชื้อรา (Fungicides) ในบรรดาสารเหล่านี้สารกำจัดแมลงเป็นสารก่อให้เกิดภาวะพิษ เป็นอันตรายต่อสุขภาพที่พบบ่อยที่สุดเมื่อเทียบกับสารเคมีกำจัดศัตรูพืชกลุ่มอื่นๆ ซึ่งพบในสารกลุ่มออร์แกโนฟอสเฟตและคาร์บาเมต หากได้รับสารเคมีกลุ่มนี้สะสมในปริมาณมากเป็นระยะเวลาอันยาวนานอาจส่งผลกระทบต่อสุขภาพ เช่น ระบบทางเดินหายใจผิดปกติ ระบบประสาทและสมองถูกทำลาย ระบบกล้ามเนื้อ กระดูกและอ่อนแรง ระบบหลอดเลือดหัวใจทำงานผิดปกติ อีกทั้งสารเคมีกลุ่มนี้ยังเป็นสารก่อให้เกิดโรคมะเร็งอีกด้วย [1]

เทคโนโลยีในปัจจุบันสำหรับตรวจสอบความเข้มข้นของสารเคมีมีหลายวิธีไม่ว่าจะเป็นเทคนิคโครมาโตกราฟีของเหลวสมรรถนะสูง (High Performance Liquid Chromatography, HPLC) หรือเทคนิคการวัดการดูดกลืนแสงหรือรังสีในช่วงอัลตราไวโอเล็ตและวิสิเบิล (Ultra Violet-Visible Spectrophotometry, UV-Vis) เทคนิคเหล่านี้มีประสิทธิภาพสูงในการวิเคราะห์เชิงคุณภาพและเชิงปริมาณแต่อย่างไรก็ตาม เทคนิคเหล่านี้ใช้อุปกรณ์ขนาดใหญ่ ราคาแพงและมีความซับซ้อนในการเตรียมตัวอย่างสำหรับการตรวจสอบ ซึ่งไม่สะดวกที่จะนำไปใช้วิเคราะห์ภาคสนามที่ต้องการความรวดเร็วและง่ายต่อการใช้งาน ซึ่งจากผลงานการวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” [2] นางสาวรุ่งรวิพรหมจันทร์ และ นางสาวโชติกา เขยกลิ่นเทศ ได้ประดิษฐ์เครื่องเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพา เพื่อใช้หาค่าความเข้มข้นของสารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยการวิเคราะห์จากภาพถ่ายสเปกตรัม มีเปอร์เซ็นต์ความผิดพลาด 21.88% ซึ่งยังไม่เพียงพอเมื่อเทียบกับการใช้เครื่องยูวี-วิสิเบิลสเปกโตรโฟโตมิเตอร์ที่มีค่าเท่ากับ 4.17% [2]

ดังนั้นผู้วิจัยจึงต้องการพัฒนางานวิจัยนี้ต่อร่วมกับกลุ่มวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต” ซึ่งพัฒนาด้านทฤษฎีและการออกแบบของอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพา สำหรับงานวิจัยนี้จะกล่าวถึงด้านการวิเคราะห์ผลลัพธ์ที่ได้จากเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพาใหม่เพื่อพัฒนาความแม่นยำด้วยการประยุกต์ใช้การเรียนรู้ของเครื่อง (Machine Learning)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

1. เพื่อปรับปรุงกระบวนการเตรียมและแปลงภาพเป็นข้อมูลตัวเลขให้เหมาะสมกับการเรียนรู้ของเครื่อง
2. เพื่อประยุกต์ใช้การเรียนรู้ของเครื่องเพื่อทำนายความเข้มข้น
3. เพื่อเพิ่มความแม่นยำในการทำนายความเข้มข้นของสารบิสฟีนอล A สำหรับเครื่องอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพา

1.3 ขอบเขตงานวิจัย

1. วิเคราะห์ความเข้มข้นของข้อมูลจาก
 - ชุดการทดลองสารละลายโพแทสเซียมเปอร์แมงกาเนต (KMnO_4) ของงานวิจัย “การพัฒนา อุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” ความเข้มข้น 5, 8, 10, 15 และ 20 ppm
 - ชุดการทดลองสารละลายคาร์โบซิลแฟน ($\text{C}_{20}\text{H}_{32}\text{N}_2\text{O}_3\text{S}$), เกลือ, น้ำตาล และสารผสมเกลือ-น้ำตาลของงานวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต” ความเข้มข้นแต่ละชุดทดลองเป็น [300, 600, 900, 1200 และ 1500 ppm], [5, 10, 12, 15 และ 20 %w/v], [5, 8, 10, 15 และ 20 %w/v] และ [(2.5,5), (2.5,7.5), (5,2.5), (5,5), (7.5,5) และ (5,7.5)] %w/v ตามลำดับ
2. ใช้ฟังก์ชันในการปรับภาพจาก OpenCV ด้วยโปรแกรมภาษา Python
3. ประยุกต์ใช้การเรียนรู้ของเครื่องด้วยวิธี Support vector regression, Random forest, Multilayer perceptron และ Extreme gradient boosting เพื่อทำนายความเข้มข้น
4. ใช้การตรวจสอบประสิทธิผลของการเรียนรู้ของเครื่องด้วย Mean Absolute Percentage Error, Mean Absolute Error, R Square และ Mean Square Error

1.4 ประโยชน์ที่ได้รับ

พัฒนาการเรียนรู้ของเครื่องให้สามารถวิเคราะห์ความเข้มข้นที่ถ่ายภาพมาจากเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพาของงานวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต” ได้อย่างแม่นยำ รวดเร็ว และสามารถปรับปรุงข้อมูลประเภทรูปภาพให้เหมาะสมแก่การวิเคราะห์ด้วยการเรียนรู้ของเครื่อง

บทที่ 2

ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องสำหรับการปรับปรุงประสิทธิภาพการทำนายความเข้มข้นของเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพาด้วยการเรียนรู้ของเครื่องประกอบไปด้วย วิธีการวิเคราะห์ความเข้มข้นของสารละลาย, ระบบสี (Color Space), ภาษาคอมพิวเตอร์, ทฤษฎีการเตรียมรูปภาพก่อนการวิเคราะห์ (Image Pre-processing) และการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งมีรายละเอียดดังต่อไปนี้

2.1 วิธีการวิเคราะห์ความเข้มข้นของสารละลาย

การวิเคราะห์ความเข้มข้นของสารละลายสามารถทำได้หลากหลายวิธี ไม่ว่าจะเป็นเทคนิคโครมาโตกราฟีของเหลวสมรรถนะสูง (High Performance Liquid Chromatography, HPLC), เทคนิคการวัดการดูดกลืนแสงหรือรังสีในช่วงอัลตราไวโอเล็ตและวิสิเบิล (Ultra Violet-Visible Spectrophotometry, UV-Vis) หรือ การวัดค่าด้วยเครื่อง Brix Refractometer โดยในงานวิจัยนี้ และงานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” [2] จะศึกษาและพัฒนาด้วยเทคนิคการวัดการดูดกลืนแสงหรือรังสีในช่วงอัลตราไวโอเล็ตและวิสิเบิล (Ultra Violet-Visible Spectrophotometry, UV-Vis)

2.1.1 สเปกโตรโฟโตมิเตอร์

สเปกโตรโฟโตมิเตอร์เป็นเครื่องมือที่อาศัยค่าการดูดซึมของแสงสำหรับสารมีสี ซึ่งมีวิธีการประมวลผลสอดคล้องกับกฎของเบียร์และแลมเบิร์ตดังต่อไปนี้ สเปกโตรโฟโตมิเตอร์เป็นเครื่องมือที่อาศัยค่าการดูดซึมของแสงสำหรับสารมีสี ซึ่งมีวิธีการประมวลผลสอดคล้องกับกฎของเบียร์และแลมเบิร์ตดังต่อไปนี้

2.1.1.1 กฎของเบียร์และแลมเบิร์ต (Beer-Lambert law)

กฎของแลมเบิร์ต (Lambert's law) มีใจความว่า “เมื่อมีแสงที่มีความยาวคลื่นเดี่ยว (monochromatic light) ผ่านตัวกลางเนื้อเดียว สัดส่วนของความเข้มของแสงที่ถูกตัวกลางนั้นดูดกลืนไว้ไม่ขึ้นอยู่กับความเข้มของแสงที่กระทบตัวกลางนั้น และความเข้มของแสงจะถูกแต่ละชั้นของตัวกลางดูดกลืนไว้ในสัดส่วนที่เท่ากัน” ดังสมการที่ (1)

$$\log \frac{I_0}{I} = \frac{k}{2.303} \quad (1)$$

เมื่อ	I_0	คือ	ความเข้มแสงเมื่อเริ่มตกกระทบสาร
	I	คือ	ความเข้มแสงหลังผ่านสารตัวกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L	คือ	ความหนาของตัวกลาง
k	คือ	ค่าคงตัวปฏิกิริยา (proportional constant)

กฎของเบียร์ (Beer's law) มีใจความว่า “เมื่อแสงที่มีความยาวคลื่นเดียว ผ่านตัวกลางเนื้อเดียว สัดส่วนของความเข้มของแสงที่ถูกตัวกลางนั้นดูดกลืนไว้จะแปรผันโดยตรงกับ ปริมาณของตัวกลางที่ดูดกลืนแสงนั้น” ดังสมการที่ (2)

$$A = \epsilon Lc \quad (2)$$

เมื่อ	A	คือ	ค่าการดูดกลืนแสง (Absorbance)
	ϵ	คือ	ค่าโมลาร์แอบซอร์ปทิวิตี (molar absorptivity)
	c	คือ	ความเข้มข้นของสารละลาย

เมื่อเราวัดการดูดกลืนแสงของสารละลาย ปริมาณความเข้มของแสงที่ถูกดูดกลืนจะขึ้นอยู่กับทั้งความเข้มข้นของสารละลายและความหนาของสารละลายที่ลำแสงต้องผ่าน จึงจำเป็นต้องรวมกฎของเบียร์และกฎของแลมเบิร์ต เรียกเป็น กฎของเบียร์-แลมเบิร์ต (Beer-Lambert law) ดังสมการที่ (3) [3]

$$A = \log_{10} \frac{I_0}{I} = \epsilon Lc \quad (3)$$

เมื่อ	A	คือ	ค่าการดูดกลืนแสง (Absorbance)
	I_0	คือ	ความเข้มแสงเมื่อเริ่มตกกระทบสาร
	I	คือ	ความเข้มแสงหลังผ่านสารตัวกลาง
	ϵ	คือ	ค่าโมลาร์แอบซอร์ปทิวิตี (molar absorptivity)
	L	คือ	ความหนาของตัวกลาง
	c	คือ	ความเข้มข้นของสารละลาย

การวัดค่าการดูดกลืนแสงของสารตัวอย่างเราสามารถทำได้โดยให้ลำแสง ผ่านเข้าไปในตัวอย่าง (Incident light: I_0) แล้ววัดปริมาณแสงที่เหลือผ่านออกมา (I) โดยเทียบกับแสง ที่ผ่านออกมาเมื่อไม่มีสารตัวอย่าง [4]

2.1.2 ความแตกต่างระหว่างยูวี-วิสิเบิล สเปกโตรโฟโตมิเตอร์ (UV-VIS Spectrophotometer) กับ สเปกโตรโฟโตมิเตอร์แบบพกพา

ความแตกต่างที่เห็นได้ชัดคือยูวี-วิสิเบิล สเปกโตรโฟโตมิเตอร์ สามารถหาค่าการดูดกลืนรังสีของสารที่อยู่ในช่วง Ultra violet (UV) และ Visible (VIS) ความยาวคลื่นประมาณ 190-1000 nm ได้ [5] ในขณะที่สเปกโตรโฟโตมิเตอร์ที่งานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

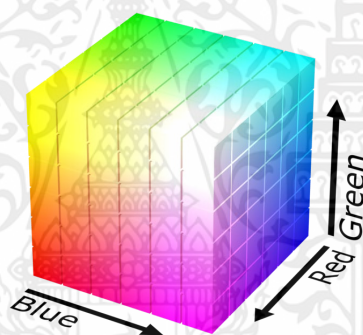
กำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” [2] พัฒนาขึ้นนั้นจะได้ผลลัพธ์ออกมาเป็นรูปภาพของสเปกตรัม โดยความยาวคลื่นในการวิเคราะห์นั้นขึ้นอยู่กับหลอดไฟภายในอุปกรณ์

2.2 ระบบสี (Color Space)

ในการแปลงรูปภาพให้กลายเป็นตัวเลขนั้นสามารถทำได้โดยการใช้ระบบสีต่างๆ ไม่ว่าจะเป็น CMYK, RGB, HSV หรือ LAB ซึ่งแต่ละระบบสี มีวิธีการในการวิเคราะห์ที่แตกต่างกัน ในงานวิจัยนี้มีการใช้สองระบบสี ประกอบด้วย RGB และ LAB

2.2.1 RGB

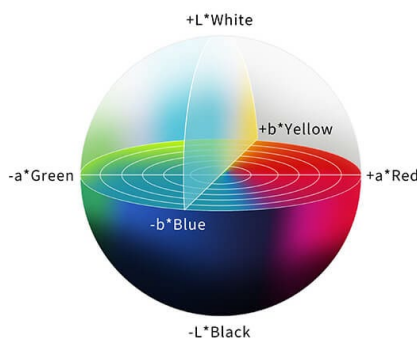
คือระบบสีที่ประกอบด้วย 3 ค่า คือ R-สีแดง(Red), G-สีเขียว(Green) และ B-สีน้ำเงิน(Blue) แต่ละค่าจะมีค่าอยู่ในช่วง 0-255 เป็นกระบวนการในการผสมสีจาก 3 แม่สี คือสีแดง สีเขียว และสีน้ำเงิน ดังรูปที่ 2.1 [6]



รูปที่ 2.1 ระบบสี RGB

2.2.2 LAB

คือระบบสีที่ประกอบด้วย 3 ค่า คือ L-บ่งบอกถึงแกนสีตั้งแต่สีขาวจนถึงสีดำ มีค่าอยู่ในช่วง 0-100, A-บ่งบอกถึงแกนสีตั้งแต่สีเขียวจนถึงสีแดง มีค่าอยู่ในช่วง -127 - 127 แล B-บ่งบอกถึงแกนสีตั้งแต่สีน้ำเงินจนถึงสีเหลือง มีค่าอยู่ในช่วง -127 - 127 เป็นระบบสีที่นิยมใช้กับอุปกรณ์สเปกโตรโฟโตมิเตอร์ ดังรูปที่ 2.2 [7]



รูปที่ 2.2 ระบบสี LAB

2.3 ภาษาคอมพิวเตอร์ (Computer language)

ภาษาคอมพิวเตอร์คือภาษาที่ออกแบบมาสำหรับการสื่อสารระหว่างมนุษย์กับคอมพิวเตอร์ ในปัจจุบันมีการศึกษาภาษาคอมพิวเตอร์อย่างแพร่หลาย ยกตัวอย่างเช่น ภาษาไพทอน (Python Language), ภาษาจาวา (Java Language) และภาษาซี (C Language) เป็นต้น สำหรับภาษาที่ผู้วิจัยเลือกใช้ในงานวิจัยนี้คือ “ภาษาไพทอน”

2.3.1 ภาษาไพทอน (Python)

ภาษาไพทอนเป็นหนึ่งในภาษาคอมพิวเตอร์ มีการใช้งานมาตั้งแต่ปี ค.ศ. 1994 ซึ่งถูกออกแบบและพัฒนาโดย กิโด ฟาน รอสซัม ชาวเนเธอร์แลนด์ เป็นภาษาที่สามารถเรียนรู้ได้ง่าย และมีประสิทธิภาพ สามารถทำงานบนแพลตฟอร์มต่างๆ ได้หลากหลาย ภาษาไพทอนยังมีชุดคำสั่งสำเร็จรูปที่ช่วยในการจัดการข้อมูล สำหรับงานวิจัยนี้ มีการใช้งานชุดคำสั่งสำเร็จรูปได้แก่ Pandas, NumPy, OpenCV และ Scikit-learn มีความสำคัญดังนี้ [8]

2.3.2 Pandas

Pandas เปิดให้ใช้งานเมื่อปี ค.ศ. 2009 ออกแบบและพัฒนาโดย เวส แมคคินนีย์ นักพัฒนาซอฟต์แวร์ชาวอเมริกัน เป็นชุดคำสั่งสำเร็จรูปสำหรับจัดการข้อมูลประเภทตาราง ซึ่งมีความจำเป็นในการวิเคราะห์ข้อมูลทั้งก่อน ระหว่าง และหลังการใช้งาน Machine Learning [9]

2.3.3 NumPy

NumPy เปิดให้ใช้งานเมื่อปี ค.ศ. 2005 ออกแบบและพัฒนาโดย ทราวิส โอลิเฟนท์ นักวิทยาศาสตร์ข้อมูลชาวอเมริกัน เป็นชุดคำสั่งสำเร็จรูปที่ใช้ในการคำนวณทางคณิตศาสตร์ เพื่อใช้ในการจัดการกับข้อมูลทางตัวเลข [10]

2.3.4 OpenCV

OpenCV เปิดให้ใช้งานเมื่อปี ค.ศ. 2002 นำโดยแกรี แบริดสกีจากบริษัท intel เป็นชุดคำสั่งสำเร็จรูป OpenCV สำหรับการจัดการข้อมูลประเภทรูปภาพ เช่น การแปลงภาพเป็นตัวเลข การปรับสี เป็นต้น [11]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5 Scikit-Learn

Scikit-Learn เปิดให้ใช้งานเมื่อปี ค.ศ. 2010 ออกแบบและพัฒนาโดยเดวิด กูร์นาโปก นักวิทยาศาสตร์ข้อมูลชาวฝรั่งเศส Scikit-Learn เป็นชุดคำสั่งสำเร็จรูปที่เก็บรวบรวมหลักการ กระบวนการวิเคราะห์ผลของ Machine Learning ชนิดต่างๆเอาไว้ [12]

2.3 ทฤษฎีการเตรียมรูปภาพก่อนการวิเคราะห์ (Image Pre-processing)

ในการเตรียมภาพก่อนการวิเคราะห์ในงานวิจัยนี้ประกอบด้วย การปิดทางด้านสัญญาณวิทยา และการเบลอภาพด้วยฟังก์ชันเกาส์เซียน ทฤษฎีทั้งสองนี้เป็นทฤษฎีที่เหมาะสมสำหรับการเลือกวิเคราะห์เฉพาะส่วนของภาพ

2.3.1 การขยาย (Dilation)

การขยายเป็นขั้นตอนที่ขยายขอบเขตของวัตถุในภาพ (โดยทั่วไปเป็นสีขาว) โดยใช้เอลิเมนต์โครงสร้าง (Structuring Element) ซึ่งเป็นเมทริกซ์ที่กำหนดผลกระทบของพิกเซลใกล้เคียง ในกระบวนการนี้ พิกเซลในภาพจะกลายเป็น '1' หากมีพิกเซลใดพิกเซลหนึ่งภายใต้เอลิเมนต์โครงสร้างเป็น '1'

2.3.2 การกัดกร่อน (Erosion)

การกัดกร่อนเป็นกระบวนการตรงข้ามกับการขยาย โดยพิกเซลในภาพดั้งเดิม (ไม่ว่าจะเป็น 1 หรือ 0 ในภาพทวิภาพ) จะถูกพิจารณาว่าเป็น 1 ก็ต่อเมื่อทุกพิกเซลที่อยู่ใต้เอลิเมนต์โครงสร้างเป็น 1 มิฉะนั้นจะถูกกัดกร่อน (ทำให้เป็นศูนย์)

2.3.3 การปิด (Closing)

การปิดเป็นการดำเนินการต่อจากการขยายและการกัดกร่อน มีประโยชน์ในการปิดช่องว่างเล็กๆ ภายในวัตถุหรือจุดดำเล็กๆ บนวัตถุ การปิดสามารถมองเป็นวิธีในการเติมช่องว่างและแก้ปัญหาในภาพ โดยเอลิเมนต์โครงสร้างกำหนดวิธีการเติม: รูปทรงและขนาดของมันสามารถส่งผลต่อผลลัพธ์ได้ [13], [14] ซึ่งให้ผลลัพธ์ดังรูปที่ 2.3



รูปที่ 2.3 รูปภาพแสดงตัวอย่างการดำเนินการปิดทั้งก่อนและหลังการดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 การเปิด (Opening)

การเปิดเป็นการดำเนินการต่อจากการกัดกร่อนและการขยาย มีประโยชน์ในการปิดช่องว่างเล็กๆ ภายนอกของภาพมราสนใจ มีประโยชน์ในการลดสีลบกวนโดยรอบแสดงดังรูปที่ 2.4



รูปที่ 2.4 รูปภาพแสดงตัวอย่างการดำเนินการเปิดก่อนและหลังการดำเนินการ

2.3.5 การเบลอภาพด้วยฟังก์ชันเกาส์เซียน (Gaussian blur)

การทำ Gaussian blur เป็นเทคนิคที่นิยมใช้ในการปรับปรุงภาพถ่ายหรือภาพดิจิทัล โดยเฉพาะในการลดแสงรบกวนหรือเพิ่มความเท่ากันของสี หรือลดความคมชัดของภาพ ซึ่งมีประโยชน์ในการปรับปรุงคุณภาพของภาพที่มีรายละเอียดจำนวนมาก

หลักการทำงานของ Gaussian blur คือการนำเอาค่าเฉลี่ยของพิกเซลที่เป็นจุดศูนย์กลาง ร่วมกับค่าของพิกเซลที่อยู่รอบๆ จุดนั้น ๆ โดยใช้น้ำหนักที่คำนวณจากฟังก์ชัน Gaussian ซึ่งให้ความสำคัญมากกว่าพิกเซลที่อยู่ห่างจากจุดศูนย์กลางมากขึ้น นั่นคือพิกเซลที่อยู่ใกล้จุดศูนย์กลางจะมีน้ำหนักสูงกว่าพิกเซลที่อยู่ห่างจากจุดศูนย์กลางมากขึ้น ทำให้ภาพที่ผ่านการประมวลผลด้วย Gaussian blur จะมีค่าสีที่เบลอลดลง และส่งผลให้ภาพดูนุ่มละเอียดขึ้น

การประมวลผล Gaussian blur สามารถทำได้โดยใช้หลายวิธี ได้แก่ การใช้ convolution ด้วย kernel ที่มีลักษณะ Gaussian, การใช้ Fast Fourier Transform (FFT) เพื่อให้การประมวลผลเร็วขึ้น หรือใช้วิธีการอื่น ๆ ที่ใช้ประโยชน์จากคุณสมบัติของ Gaussian function ในการกระจายค่าความเข้มของภาพ [15], [16]

2.3.6 การจับรูปทรงของภาพ (Contour Detection)

การจับรูปทรงของภาพ (Contour Detection) เป็นเทคนิคการประมวลผลภาพดิจิทัลที่ใช้ในการค้นหาเส้นโค้งที่แยกวัตถุออกจากพื้นหลัง อัลกอริธึมที่ใช้ในการจับรูปทรงภาพ แบ่งออกเป็น 2 ประเภทหลัก ได้แก่:

อัลกอริธึมแบบติดตามเส้นขอบ (Boundary Tracing Algorithms): อัลกอริธึมประเภทนี้จะติดตามเส้นขอบของวัตถุที่ละพิกเซล โดยเริ่มจากจุดเริ่มต้นและติดตามเส้นขอบไปจนกว่าจะครบรอบวัตถุ อัลกอริธึมที่นิยมใช้ ได้แก่:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

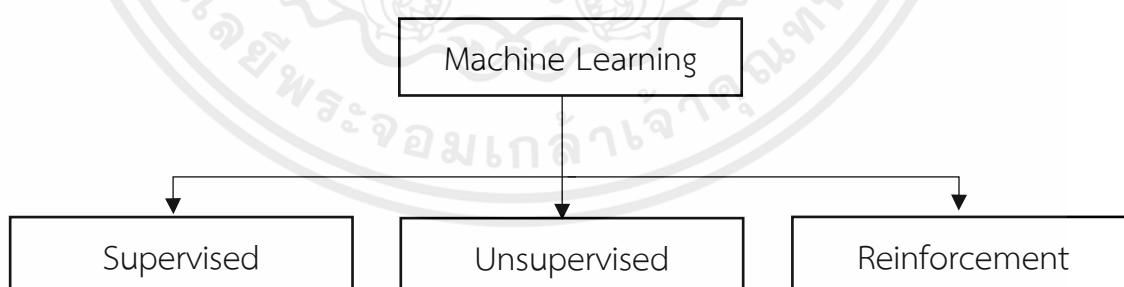
- อัลกอริธึมแบบติดตามขอบ (Chain Code)
- อัลกอริธึมแบบเติมเต็มเส้น (Flood Fill)
- อัลกอริธึมแบบติดตามขอบแบบ Freeman

อัลกอริธึมแบบหาเส้นโค้ง (Curve Fitting Algorithms): อัลกอริธึมประเภทนี้จะหาเส้นโค้งที่เหมาะสมที่สุดกับเส้นขอบของวัตถุ อัลกอริธึมที่นิยมใช้ ได้แก่:

- อัลกอริธึมแบบ Ramer-Douglas-Peucker
- อัลกอริธึมแบบ Least Squares Fitting
- อัลกอริธึมแบบ B-spline Fitting [17]

2.4 การเรียนรู้ของเครื่อง (Machine Learning)

คือการทำให้คอมพิวเตอร์เรียนรู้และพัฒนาจากข้อมูลที่ระบบได้รับ โดยไม่จำเป็นต้องให้มนุษย์มาคอยกำกับหรือเขียนโปรแกรมเพิ่มเติม ในภายหลังหากคอมพิวเตอร์ได้รับข้อมูลใหม่ๆจะสามารถตอบสนองได้เอง การเรียนรู้ของเครื่องมีการถูกใช้งานครั้งแรกในปี ค.ศ. 1950 เมื่อนักวิทยาศาสตร์หาวิธีการสอนให้คอมพิวเตอร์เล่นหมากฮอส [18] การให้คอมพิวเตอร์เรียนรู้นั้นมีตัวแปรที่สำคัญ 2 อย่างคือ ตัวแปรต้น และตัวแปรตาม หรือในทางการเรียนรู้ของเครื่อง (Machine Learning) เรียกว่า “Feature” และ “Label data” ตามลำดับ รูปแบบการเรียนรู้ของเครื่องแบ่งเป็น 3 รูปแบบประกอบด้วย การเรียนรู้แบบมีผู้สอน (Supervised Learning), การเรียนรู้โดยไม่มีผู้สอน (Unsupervised Learning) และการเรียนรู้ผ่านการให้รางวัล (Reinforcement Learning) สำหรับการเรียนรู้ที่ผู้วิจัยเลือกใช้ในงานวิจัยนี้คือ “การเรียนรู้แบบมีผู้สอน (Supervised Learning)” ซึ่งแสดงเป็นแผนภาพได้ดังรูปที่ 2.5



รูปที่ 2.5 ชนิดของการเรียนรู้ของเครื่อง

การเรียนรู้แบบมีผู้สอนมีความแตกต่างจากการเขียนโปรแกรมแบบดั้งเดิมคือ การเขียนโปรแกรมแบบดั้งเดิมจำเป็นต้องมีหลักการ (Logic) ในการหาความสัมพันธ์ระหว่างตัวแปรต้นและตัวแปรตาม จึงสามารถหาตัวแปรตามจากตัวแปรต้นได้ แต่ในกรณีของการเรียนรู้แบบมีผู้สอนนั้นเพียงแค่มีตัวแปรต้น และตัวแปรตาม เพื่อให้คอมพิวเตอร์หาความสัมพันธ์ระหว่างทั้ง 2 ตัวแปร เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นเมื่อมีตัวแปรต้นที่การเรียนรู้ของเครื่องไม่เคยเจอ ก็สามารถทำนายตัวแปรตามได้ โดยขึ้นอยู่กับความแม่นยำของการเรียนรู้ [19] การเรียนรู้แบบมีผู้สอนมีโมเดลหรืออัลกอริทึมสำหรับหาความสัมพันธ์ระหว่าง 2 ตัวแปรหลากหลายวิธี สำหรับโมเดลที่ผู้วิจัยเลือกใช้ในงานวิจัยนี้คือ “Random Forest (RF)”, “Support Vector Regression (SVR)” และ “Multilayer Perceptron (MLP)”

2.5 อัลกอริทึมการเรียนรู้ของเครื่อง

เป็นขั้นตอนหรือกระบวนการทางคณิตศาสตร์ที่ใช้ในการสร้างโมเดลเพื่อวิเคราะห์และทำนายข้อมูล โดยใช้ตัวอย่างข้อมูลในการฝึกฝนและปรับปรุงประสิทธิภาพของโมเดล ในงานวิจัยนี้ได้มีการใช้งาน 4 อัลกอริทึม ประกอบด้วย Random Forest, Support Vector Regression, Multilayer Perceptron และ Extreme Gradient Boosting

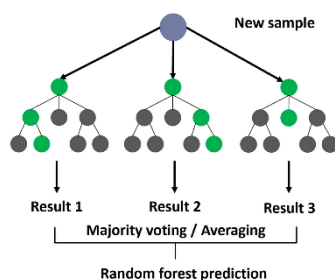
2.5.1 Random Forest (RF)

เป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของเครื่อง (Machine Learning) ที่มีความหลากหลายในการประมวลผลข้อมูล หลักการทำงานของ Random Forest ประกอบด้วย

1. การสร้างต้นไม้ (Tree Creation): การสร้างต้นไม้ใน Random Forest จะใช้ข้อมูลที่ถูกรวบรวมมาจากชุดข้อมูลทั้งหมด โดยการสุ่มชุดข้อมูลและคุณสมบัติที่จะใช้ในการสร้างต้นไม้เพื่อให้แต่ละต้นไม้มีความหลากหลายในการทำงาน
2. การผลักดันข้อมูล (Data Pushing): ข้อมูลทดสอบจะถูกผลักดันผ่านทุกต้นไม้ใน Random Forest เพื่อให้แต่ละต้นไม้ทำการคาดเดาผลลัพธ์ และแต่ละต้นไม้จะคืนค่าทำนายที่แตกต่างกันกลับมา
3. การโหวต (Voting): ผลลัพธ์จากแต่ละต้นไม้จะถูกรวบรวมและนับคะแนน จากนั้นเลือกผลลัพธ์ที่ได้คะแนนสูงสุดเป็นผลลัพธ์สุดท้ายของ Random Forest
4. การลด overfitting: ด้วยการใช้การสุ่มข้อมูลและคุณสมบัติในการสร้างต้นไม้แต่ละต้น และการรวมผลลัพธ์จากหลายต้นไม้ จะช่วยลดปัญหาการ overfitting ที่อาจเกิดขึ้นในต้นไม้แต่ละต้น การ overfitting คือการที่โมเดลทำนายตัวแปรตามออกมาได้ดีเกินไป ซึ่งเป็นเหตุมาจากการที่นำข้อมูลที่มีอิทธิพลจากสิ่งรบกวนมาคิดด้วย

ผลลัพธ์จากโมเดล Random Forest มีลักษณะเป็นแผนภาพต้นไม้ดังแสดงในรูปที่

2.6 ประกอบด้วยต้นไม้หลายๆต้น แต่ละต้นสามารถทำนายผลลัพธ์ที่ได้ ซึ่งโมเดลนี้จะนำผลลัพธ์ทั้งหมดนี้มารวมโหวตเพื่อค่าที่เหมาะสมที่สุด



รูปที่ 2.6 Random Forest (RF)

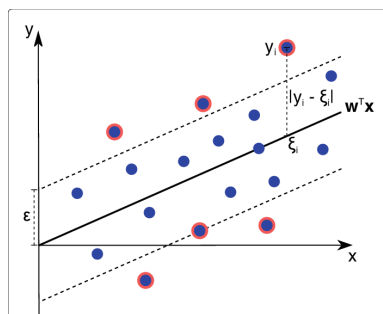
2.5.2 Support Vector Regression (SVR)

เป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของเครื่อง (Machine Learning) โดยมุ่งเน้นการพยากรณ์ค่าต่อเนื่อง (Continuous Value Prediction) โดยหลักการทำงานของ SVR ประกอบด้วย

1. การค้นหาแผนผังเส้นในพื้นที่หลัก (Feature Space): SVR จะพยากรณ์ค่าโดยการค้นหาแผนผังเส้นในพื้นที่หลักที่ทำให้ข้อมูลถูกแบ่งออกเป็นกลุ่มต่างๆ โดยการใช้ข้อมูลตัวอย่างที่เรียกว่า Support Vectors.
2. การกำหนดเส้นหรือแผนผังที่เหมาะสม (Fitting the Best-Fit Line or Hyperplane): SVR จะพยากรณ์ค่าโดยการกำหนดเส้นหรือแผนผังที่เหมาะสมในพื้นที่หลัก ที่มีระยะห่างระหว่างข้อมูลตัวอย่างและเส้นหรือแผนผังน้อยที่สุด
3. การกำหนด Margin และค่าความคลาดเคลื่อน (Margin and Error Tolerance): SVR จะใช้ Margin เพื่อกำหนดระยะห่างระหว่างเส้นหรือแผนผังกับข้อมูลตัวอย่าง และใช้ค่าความคลาดเคลื่อน (Error Tolerance) เพื่อกำหนดขอบเขตของความผิดพลาดที่ยอมรับได้
4. การหาฟังก์ชันเชิงเส้นที่เหมาะสม (Finding the Best-Fit Linear Function): โดยการใช้การเลือกแผนผังที่มี Margin ของข้อมูลตัวอย่างที่สูงที่สุด และค่าความคลาดเคลื่อนที่ต่ำที่สุด จากนั้นใช้ฟังก์ชันเชิงเส้นที่เหมาะสมเพื่อทำนายค่าต่อเนื่อง

องค์ประกอบทั้งหมดแสดงดังรูปที่ 2.7 ประกอบด้วยฟังก์ชันที่เหมาะสม $w^T x$ และเส้นปะเป็นบริเวณที่ห่างออกไปจากฟังก์ชันระยะ Margin และข้อมูลที่ห่างจากระยะ Margin “Support Vector”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



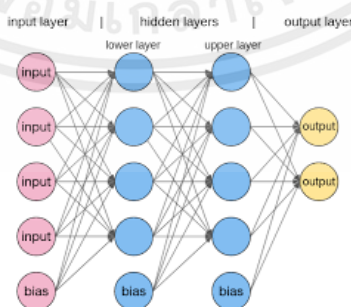
รูปที่ 2.7 Support Vector Regression (SVR)

2.5.3 Multilayer Perceptron (MLP)

เป็นโมเดลที่ใช้ในการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งเป็นประเภทหนึ่งของโครงข่ายประสาทเทียม (Neural Network) โดยหลักการทำงานของ MLP ประกอบด้วย

1. การส่งออกและการคำนวณ: ข้อมูลจะถูกส่งเข้าสู่ MLP ผ่านชั้นนำเข้า (Input Layer) ซึ่งจะทำการคำนวณและส่งต่อผ่านชั้นซ่อน (Hidden Layer) และมีการใช้ฟังก์ชันเชื่อมโยง (Activation Function) เพื่อกำหนดค่าเอาต์พุตของแต่ละโหนด
2. การเรียนรู้แบบถดถอย (Backpropagation Learning): MLP ใช้วิธีการถดถอยในการปรับค่าน้ำหนักและการเชื่อมโยงภายในโครงข่าย โดยใช้ข้อมูลผลลัพธ์ที่ได้จากการคาดเดาเพื่อปรับปรุงการทำนายในครั้งต่อไป
3. การจำแนกและทำนายข้อมูล: เมื่อการเรียนรู้เสร็จสิ้นแล้ว MLP จะสามารถใช้ค่าน้ำหนักที่ได้ปรับแล้วในการทำนายหรือจำแนกข้อมูลใหม่ที่เข้าถึง [20]

ผลลัพธ์จากโมเดล Multilayer Perceptron มีลักษณะเป็นโครงข่ายเส้นประสาท ดังแสดงในรูปที่ 2.8 ประกอบด้วยเซลล์หลายๆชั้น แบ่งเป็นชั้น input layer มีจำนวนเซลล์เท่ากับจำนวนของตัวแปรต้น, hidden layer ที่เป็นเซลล์ประมวลผลข้อมูล และ output layer ผลลัพธ์ซึ่งเป็นตัวแปรตาม



รูปที่ 2.8 Multilayer Perceptron (MLP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) เป็นอัลกอริธึมการเรียนรู้แบบ ensemble ที่ใช้ decision tree ในการจำแนกประเภทและการถดถอย XGBoost พัฒนาโดย Tianqi Chen จากมหาวิทยาลัยคาร์เนกีเมลลอน XGBoost มีจุดเด่น ดังนี้:

- มีประสิทธิภาพสูง: XGBoost เร็วกว่าอัลกอริธึม boosting อื่นๆ เช่น GBDT
- แม่นยำ: XGBoost ให้ผลลัพธ์ที่มีความแม่นยำสูง
- ยืดหยุ่น: XGBoost รองรับการปรับแต่ง parameters หลายตัว
- ใช้งานง่าย: XGBoost มี API ที่ใช้งานง่าย

XGBoost ทำงานโดยการสร้าง decision tree หลายๆ ต้น (weak learners) แต่ละต้นจะเรียนรู้จากข้อมูลที่ผิดพลาดจากต้นก่อนหน้า ensemble ของ decision tree เหล่านี้จะทำนายผลลัพธ์สุดท้าย [21]

2.6 การเพิ่มประสิทธิภาพ

จากกระบวนการทั้งหมดที่ได้กล่าวไปในกระบวนการเรียนรู้ด้วยตัวเองของเครื่องนั้น จำเป็นจะต้องมีการตรวจสอบประสิทธิผลของการเรียนรู้ของเครื่องโดยใช้วิธีการทางสถิติ ประกอบด้วย

2.6.1 MAPE (Mean Absolute Percentage Error)

MAPE เป็นค่าเฉลี่ยของร้อยละความคลาดเคลื่อนที่คำนวณจากค่าผิดพลาดแต่ละข้อมูลเทียบกับค่าจริง มีประโยชน์ในการประเมินประสิทธิภาพของแบบจำลองทางสถิติ ดังสมการที่ 4

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - F_i}{A_i} \right| \times 100\% \quad (4)$$

เมื่อ	n	คือ	จำนวนของข้อมูล
	A_i	คือ	ค่าจริงที่ตำแหน่ง i
	F_i	คือ	ค่าทำนายที่ตำแหน่ง i

2.6.2 MAE (Mean Absolute Error)

MAE ค่าเฉลี่ยของความคลาดเคลื่อนที่มีมาตรฐานเดียวกันกับข้อมูลต้นฉบับ ใช้ในการประเมินความถูกต้องของแบบจำลองอย่างเป็นทางการ ดังสมการที่ 5

$$MAE = \frac{1}{n} \sum_{i=1}^n |A_i - F_i| \quad (5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ	n	คือ	จำนวนของข้อมูล
	A_i	คือ	ค่าจริงที่ตำแหน่ง i
	F_i	คือ	ค่าทำนายที่ตำแหน่ง i

2.6.3 R-square (Coefficient of Determination)

R-square ใช้ในการวัดว่าแบบจำลองสามารถอธิบายข้อมูลตัวอย่างในแบบจำลองที่ใช้ค่าเฉลี่ยของตัวแปรตามเป้าหมายหรือไม่ มีความสำคัญเป็นอย่างมากในการประเมินความเหมาะสมของแบบจำลอง ดังสมการที่ 6

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (6)$$

เมื่อ	n	คือ	จำนวนของข้อมูล
	y_i	คือ	ค่าจริงที่ตำแหน่ง i
	\hat{y}_i	คือ	ค่าทำนายที่ตำแหน่ง i
	\bar{y}	คือ	ค่าเฉลี่ย

2.6.4 MSE (Mean Squared Error):

MSE เป็นค่าเฉลี่ยของความผิดพลาดที่ถูกยกกำลังสอง เป็นตัวชี้วัดที่นิยมใช้ในการประเมินความถูกต้องของแบบจำลองทางสถิติ ดังสมการที่ 7

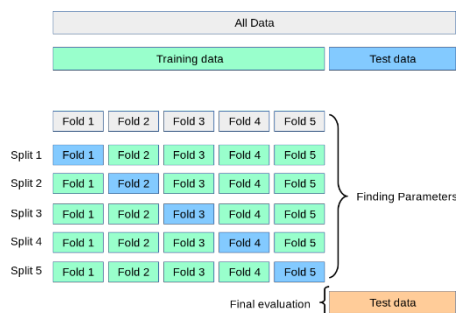
$$MSE = \frac{1}{n} \sum_{i=1}^n (A_i - F_i)^2 \quad (7)$$

เมื่อ	n	คือ	จำนวนของข้อมูล
	A_i	คือ	ค่าจริงที่ตำแหน่ง i
	F_i	คือ	ค่าทำนายที่ตำแหน่ง i

2.6.5 Cross Validation

การ Cross validation เป็นเทคนิคสำคัญที่ใช้ในการเรียนรู้ของเครื่องเพื่อประเมินประสิทธิภาพของแบบจำลองการคาดการณ์กับข้อมูลที่ไม่เคยเห็น โดยเกี่ยวข้องกับการแบ่งส่วนชุดข้อมูลออกเป็นหลายชุดย่อย เพื่อฝึกและทดสอบแบบจำลองซ้ำๆ วิธีการที่พบบ่อยที่สุดคือการตรวจสอบความถูกต้อง Cross validation แบบ k-fold โดยที่ข้อมูลจะถูกแบ่งออกเป็นชุดย่อย k และแบบจำลองจะได้รับการฝึก k ครั้ง ในแต่ละครั้งใช้ชุดย่อยที่แตกต่างกันเป็นชุดทดสอบ และส่วนที่เหลือเป็นชุดการฝึกดังแสดงในรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การ Cross Validation แบบ k-fold

การตรวจสอบความถูกต้อง cross validation แบบแบ่งชั้น k-fold จะรักษาการกระจายคลาสเดียวกันในแต่ละส่วน การตรวจสอบความถูกต้องแบบ cross-one-out (LOOCV) เป็นกรณีพิเศษของการตรวจสอบความถูกต้องแบบ cross-fold แบบ k-fold โดยที่แต่ละส่วนจะมีตัวอย่างเพียงตัวอย่างเดียว การตรวจสอบข้ามแบบ Leave-p-out เป็นลักษณะทั่วไปของ LOOCV โดยที่ตัวอย่าง p จะถูกละไว้ใน การวนซ้ำแต่ละครั้ง การตรวจสอบความถูกต้องข้ามช่วยในการประเมินประสิทธิภาพของแบบจำลอง หลีกเลี่ยงการติดตั้งมากเกินไป การเลือกพารามิเตอร์ที่ดีที่สุด และปรับปรุงความสามารถทั่วไปของแบบจำลอง เป็นเครื่องมือสำคัญในการประเมินประสิทธิภาพของอัลกอริธึมการเรียนรู้ของเครื่องต่างๆ และเพื่อรับรองความแข็งแกร่งของโมเดลที่ผ่านการฝึกอบรม [22]

2.7 งานวิจัยที่เกี่ยวข้อง

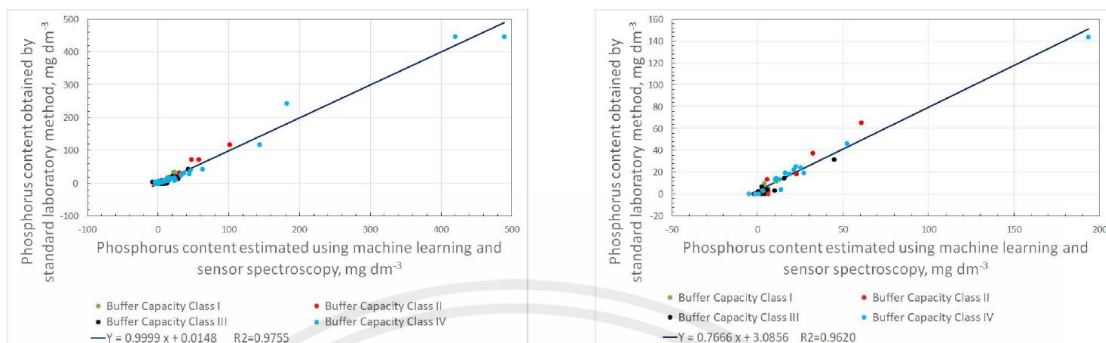
จากงานวิจัยต้องการจะทำนายความเข้มข้นด้วยการเรียนรู้ของเครื่องเพื่อเพิ่มประสิทธิภาพของสเปกโตรโฟโตมิเตอร์แบบพกพา สำหรับงานวิจัยที่เกี่ยวข้องในด้านการใช้การเรียนรู้ของเครื่องเพื่อทำนายความเข้มข้นร่วมกับสเปกโตรโฟโตมิเตอร์ ประกอบด้วย

2.7.1 งานวิจัยตรวจสอบปริมาณฟอสฟอรัสในดินด้วยสเปกโตรโฟโตมิเตอร์ต้นทุนต่ำและการเรียนรู้ของเครื่อง

งานวิจัยนี้มุ่งเน้นไปที่การพัฒนาวิธีทดสอบฟอสฟอรัสในดินเป็นหลัก ผ่านการใช้เทคโนโลยีพรีอิกซิมัลสเปกโทรสโกปีและการเรียนรู้ของเครื่อง โดยเกี่ยวข้องกับการใช้เรซินแลกเปลี่ยนไอออน ทั้งในสภาวะความอิ่มตัวมาตรฐานและสถานะดัดแปลง เพื่อแยกไอออนฟอสฟอรัสจากตัวอย่างดิน ขั้นตอนการปรับเปลี่ยนใช้รีเอเจนต์ของโครโมฟอร์เพื่อเพิ่มสารละลายความอิ่มตัว ซึ่งนำไปสู่ความสัมพันธ์ที่สูงระหว่างการสะท้อนกลับและความเข้มข้นของฟอสฟอรัส คอมพิวเตอร์ฝังตัวต้นทุนต่ำซึ่งประกอบด้วยไมโครคอนโทรลเลอร์ ATMEGA328P และเซ็นเซอร์สเปกโตรโฟโตมิเตอร์ SparkFun AS7265x ได้รับการพัฒนาสำหรับการรวบรวมข้อมูล มีการใช้แบบจำลองการเรียนรู้ของเครื่อง โดยเฉพาะการถดถอยเชิงเส้นพหุคูณเพื่อสร้างแบบจำลองการคาดการณ์ที่แข็งแกร่งโดยใช้ข้อมูลการสะท้อนกลับ การศึกษาเน้นถึงผลกระทบในทางปฏิบัติของวิธีการในการเกษตรกรรมที่แม่นยำ โดยเน้นถึงศักยภาพในการลดต้นทุนและแรงงานในการวิเคราะห์ดิน การค้นพบนี้เน้นย้ำถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

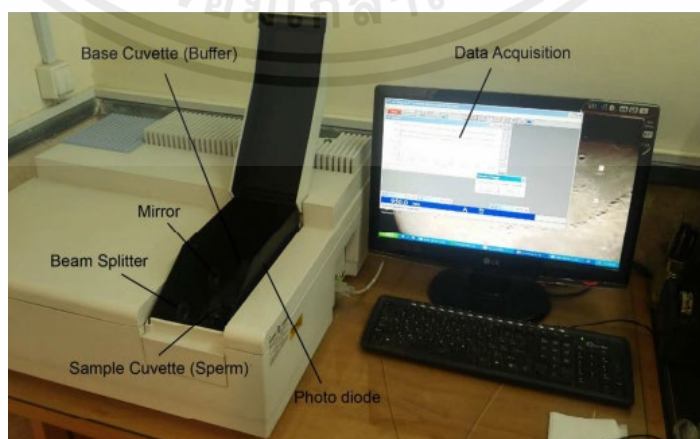
ความเป็นไปได้ในการใช้เทคนิคที่นำเสนอในการประมาณความเข้มข้นของฟอสฟอรัสในดินอย่างแม่นยำ ซึ่งจะช่วยอำนวยความสะดวกในการจัดการปุ๋ยอย่างมีประสิทธิภาพและเพิ่มประสิทธิภาพผลผลิตพืชผลและได้กราฟการทำนายในชุดข้อมูลฝึกฝนและทดสอบดังรูปที่ 2.10 [23]



รูปที่ 2.10 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set จากงานวิจัยตรวจสอบปริมาณฟอสฟอรัสในดินด้วยสเปกโตรโฟโตมิเตอร์ต้นทุนต่ำและการเรียนรู้ของเครื่อง

2.7.2 งานวิจัยการวัดปริมาณความเข้มข้นของอสุจิของมนุษย์โดยใช้เครื่องสเปกโตรโฟโตมิเตอร์แบบอาศัยการเรียนรู้ของเครื่อง

การศึกษานี้แนะนำการประยุกต์ใช้โครงข่ายประสาทเทียมในสเปกโตรโฟโตเมตรีเพื่อวัดปริมาณความเข้มข้นของอสุจิของมนุษย์ ซึ่งให้วิธีการที่รวดเร็วและแม่นยำเมื่อเปรียบเทียบกับวิธีการทั่วไป ด้วยการวิเคราะห์สเปกตรัมการดูดกลืนแสงแบบเต็ม โมเดลโครงข่ายประสาทเทียมที่พัฒนาแล้วมีความแม่นยำในการทำนายมากกว่า 93% และแสดงให้เห็นถึงข้อตกลงกับการประเมินทางคลินิก ซึ่งบ่งชี้ถึงศักยภาพในการจัดการกับภาวะมีบุตรยากในผู้ชาย ความคุ้มค่าและประสิทธิภาพที่เหนือกว่าของวิธีการนี้ชี้ให้เห็นถึงความเหมาะสมในการนำไปใช้อย่างแพร่หลาย และความสามารถในการเปิดช่องทางใหม่สำหรับการวิจัยในสาขานี้ และมีรูปของอุปกรณ์ที่ใช้ในการวิเคราะห์ดังรูปที่ 2.11 [24]



รูปที่ 2.11 อุปกรณ์ที่ใช้ในการวิเคราะห์และเก็บค่าความเข้มข้นของอสุจิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงาน

จากการศึกษาสามารถแบ่งกระบวนการในการการปรับปรุงประสิทธิภาพการทำงาน ความเข้มข้นของเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพาด้วยการเรียนรู้ของเครื่องเป็น 2 ส่วน ได้แก่ 1) การเตรียมรูปภาพและการแปลงรูปภาพเป็นข้อมูลตัวเลข 2) การประยุกต์ใช้การเรียนรู้ของเครื่อง ซึ่งเขียนเป็นผังงานดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ผังวิธีการดำเนินงาน

3.1 การเตรียมรูปภาพและการแปลงรูปภาพเป็นข้อมูลตัวเลข

ในการทำนายหาความเข้มข้นของสารด้วยการเรียนรู้ของเครื่อง ด้วยการวิเคราะห์จากแถบสเปกตรัมรูปภาพที่ได้จากอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพา ไม่สามารถนำภาพไปใช้เพื่อฝึกฝนการเรียนรู้ของเครื่องได้โดยตรง ต้องเป็นข้อมูลประเภทตัวเลข ก่อนหน้านั้นยังจำเป็นต้องมีการเลือกส่วนของแถบสเปกตรัมที่จะใช้ในการวิเคราะห์ แล้วจึงค่อยนำไปแปลงเป็นตัวเลข

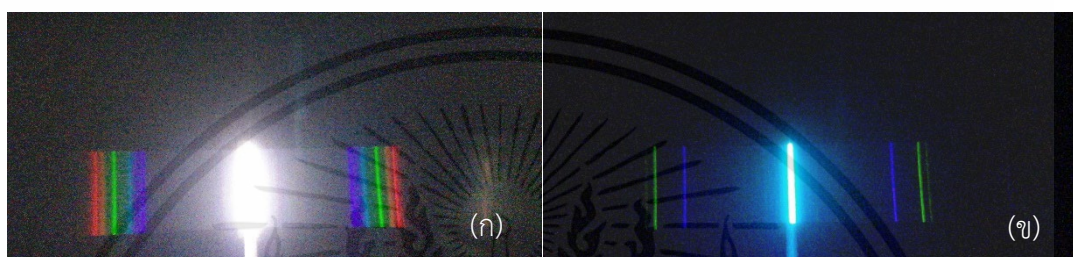
ดังนั้นการเตรียมรูปภาพและการแปลงรูปภาพเป็นข้อมูลตัวเลขก่อนการวิเคราะห์ จำเป็นต้องมีการปรับใช้ให้เข้ากับชนิดข้อมูล ซึ่งทั้งสองชุดการทดลองมีลักษณะทางกายภาพของรูปที่ต่างกันอย่างมาก ผู้วิจัยจึงแบ่งส่วนการเตรียมรูปภาพและแปลงเป็นตัวเลขออกตามชุดการทดลองที่ได้วิเคราะห์ ดังนี้

3.1.1 งานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมต โดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล”

สำหรับชุดทดลองนี้มีขั้นตอนการเตรียมรูปภาพและการแปลงรูปภาพเป็นข้อมูลตัวเลขมีชุดข้อมูลดังตารางที่ 3.1 และมีตัวอย่างรูปดังรูปที่ 3.2

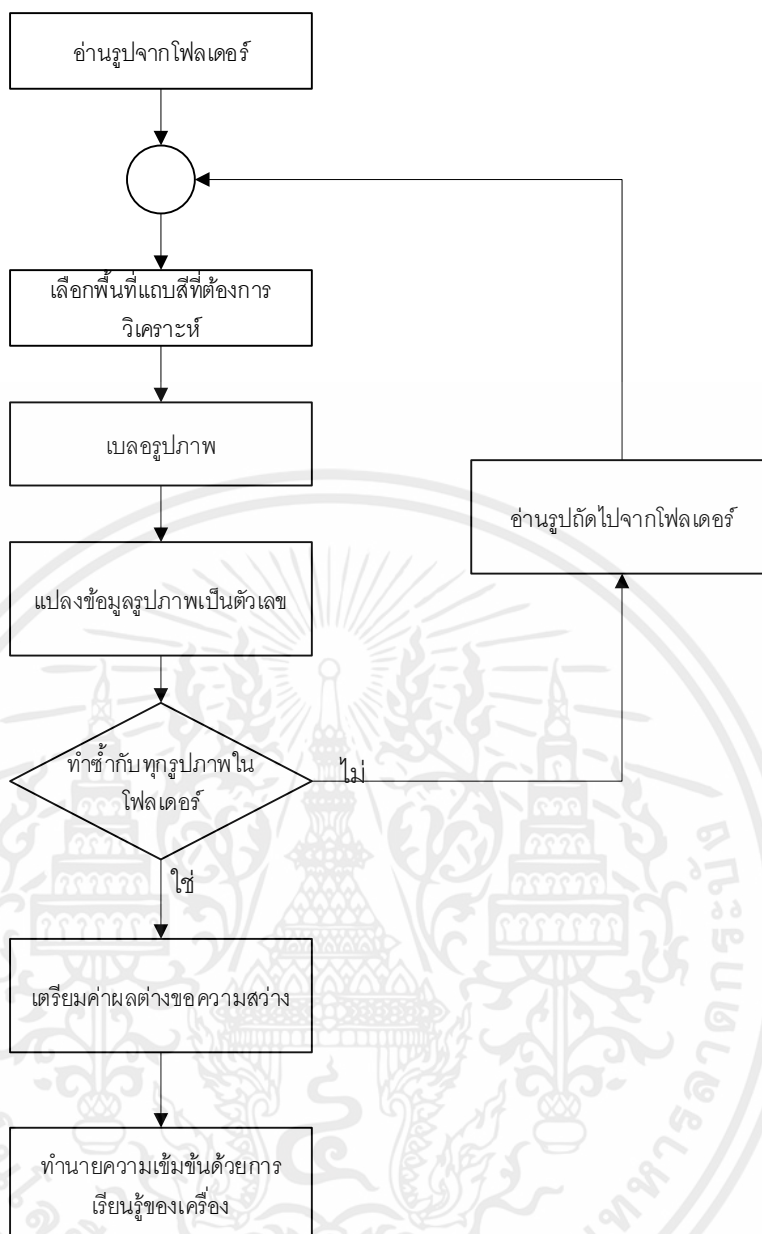
ตารางที่ 3.1 สรุปชุดข้อมูลของงานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล”

ชุดการทดลอง	ความเข้มข้น (ppm)	จำนวนรูปต่อความเข้มข้น
โพแทสเซียมเปอร์แมงกาเนต	5, 8, 10, 15 และ 20	10
คาร์โบซิลแฟน	1, 3, 5 และ 7	10



รูปที่ 3.2 ตัวอย่างรูปจากชุดการทดลอง (ก)โพแทสเซียมเปอร์แมงกาเนต และ(ข)คาร์โบซิลแฟน แสดงเป็นแผนภาพได้ดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แผนภาพแสดงขั้นตอนการเตรียมรูปภาพและแปลงเป็นตัวเลขชุดการทดลอง “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล”

3.1.1.1 อ่านรูปจากฟิล์มเดออร์

ภายในฟิล์มเดออร์ของข้อมูลดิบที่ใช้ในการวิเคราะห์ค่าความเข้มข้นของสารประกอบด้วยชุดรูปภาพ โดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล ซึ่งใช้สารสองชนิดคือ สารละลายโพแทสเซียมเปอร์แมงกาเนต (KMnO_4) และ สารกำจัดศัตรูพืชกลุ่มคาร์บาเมต “คาร์โบซัลแฟน” โดยข้อมูลดิบจะประกอบด้วยชื่อของรูปภาพที่บ่งบอกถึงความเข้มข้นของสารนั้น และรูปภาพสเปกตรัมของสารนั้น โดยจะแบ่งเป็นรูปภาพของสารละลายที่รู้ความเข้มข้น และสารละลายไร้สิ่งตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.2 เลือกพื้นที่แถบสีที่ต้องการวิเคราะห์

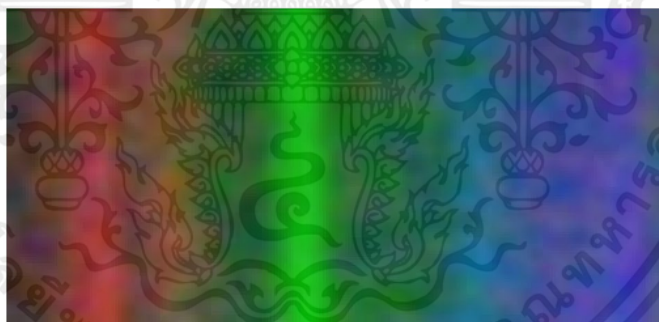
ภาพถ่ายภายในชุดทดลองตำแหน่งของแถบสีภายในภาพถ่ายทุกๆรูปนั้นมีตำแหน่งที่ใกล้เคียงกันดังนั้นในการตัดภาพจึงใช้ชุดคำสั่ง จึงเลือกตัดพื้นที่ของแถบสีตามตำแหน่ง ดังรูปที่ 3.4



รูปที่ 3.4 ชุดการทดลอง KMnO₄ “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” หลังการตัดรูปภาพ

3.1.1.3 เบลอรูปภาพ

เมื่อเลือกตัดพื้นที่แถบสีที่เหมาะสมกับการวิเคราะห์แล้วจะเห็นได้ว่าภายในแถบสีมีแสงรบกวน (Noise) มาก จึงเลือกใช้คำสั่ง Gaussian blur จาก OpenCV เพื่อลดแสงรบกวน ได้ดังรูปที่ 3.5



รูปที่ 3.5 ชุดการทดลอง KMnO₄ “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” หลังการเบลอรูปภาพ

3.1.1.4 การแปลงรูปภาพเป็นข้อมูลตัวเลข

เมื่อรูปภาพมีความเหมาะสมแก่การวิเคราะห์แล้ว จึงแปลงรูปภาพเป็นตัวเลขด้วยค่า L จากระบบสี LAB เพื่อแสดงถึงค่าความสว่างของพิกเซลนั้นๆ และนำค่า L ของสารละลายไว้สิ่งตัวอย่างลบด้วยค่า L ของรูปสารละลายที่รู้ความเข้มข้น ด้วยชุดคำสั่ง Pandas, NumPy และ OpenCV

เริ่มต้นใช้ชุดคำสั่ง OpenCV และ Pandas อ่านค่าสีในแต่ละพิกเซลและเก็บเป็น ดาต้าเฟรม ขนาด 207x100 ตามขนาดของภาพ ก่อนจะใช้ NumPy และ Pandas ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลี่ยค่า L ในแต่ละแถวเพื่อลดสรีบกววน ได้ผลลัพธ์เป็น ดาต้าเฟรม ขนาด 1x207 และใช้กระบวนการดังกล่าวกับทุกๆ แถบสีสำหรับภาพถ่ายในชุดทดลองด้วยภาษา Python กับชุดคำสั่ง Pandas ได้เป็น ดาต้าเฟรม ขนาด 50x207 ซึ่งมีแถวและหลักเป็น ความเข้มข้นของสารซึ่งมีจำนวนเท่ากับรูปภาพ และค่า L ในแต่ละพิกเซลตามลำดับ สุดท้ายนำ ดาต้าเฟรม ดังกล่าวไปลบจากค่า L ของสารละลายไว้ สิ่งตัวอย่างในทุกๆหลักเป็น ดาต้าเฟรม ขนาด 50x207 ซึ่งมีแถวและหลักเป็น ความเข้มข้นของสาร และค่า ΔL ของสารละลายไว้สิ่งตัวอย่างและสารละลายที่รู้ความเข้มข้น

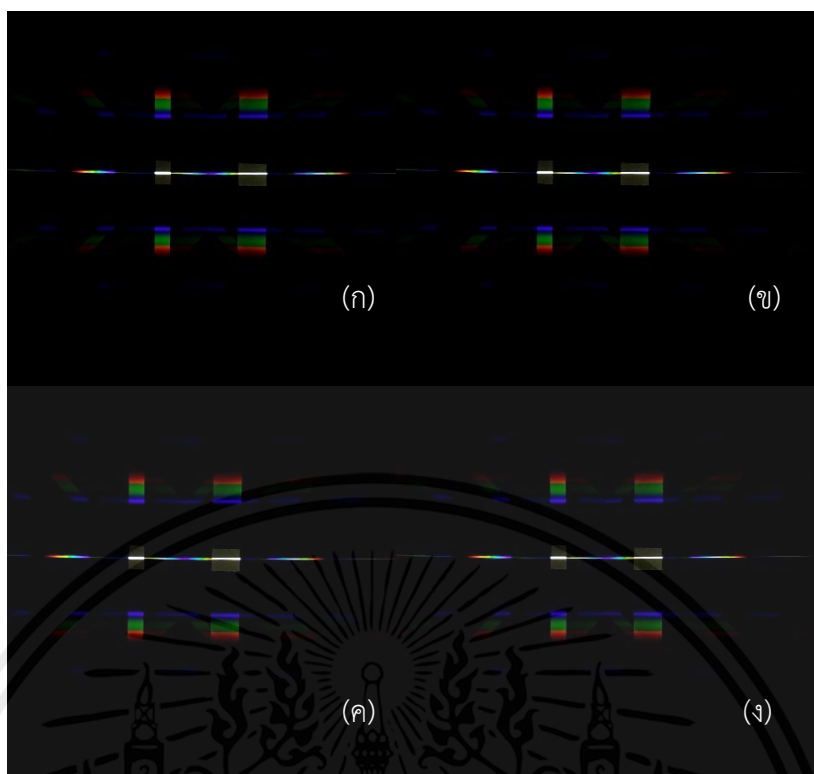
3.1.2 งานวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต”

ในงานวิจัยนี้ได้มีการประดิษฐ์และพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาขึ้นมาใหม่ ทำให้มีลักษณะของข้อมูลรูปภาพที่แตกต่างจากงานวิจัย การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล” จึงทำให้ผู้วิจัยพัฒนากระบวนการเตรียมรูปภาพและแปลงเป็นตัวเลขแสดงใหม่ มีชุดข้อมูลดังตารางที่ 3.2 มีตัวอย่างรูปดังรูปที่ 3.6 และแผนภาพกระบวนการดังรูปที่ 3.7

ตารางที่ 3.2 สรุปชุดข้อมูลของงานวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต”

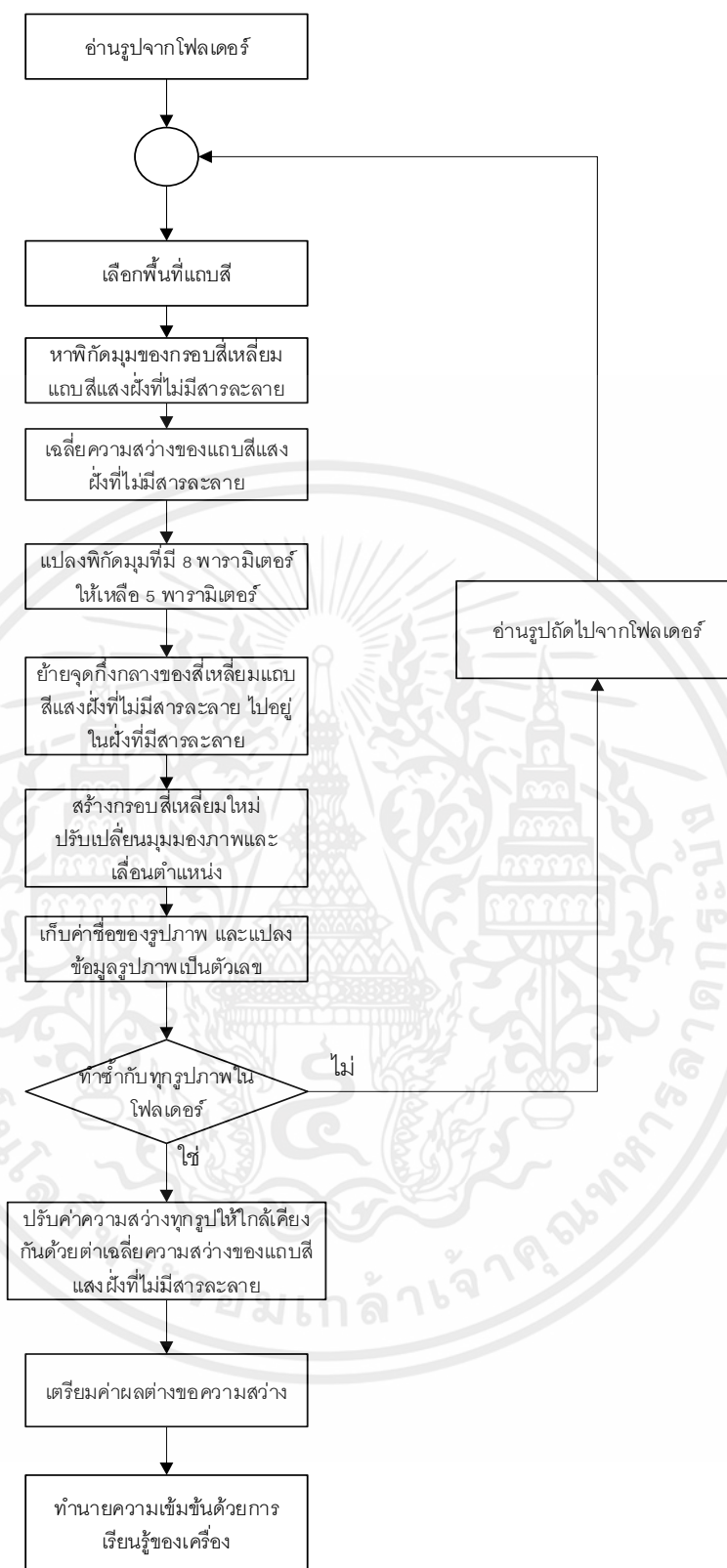
ชุดการทดลอง	ความเข้มข้น	หน่วยความเข้มข้น	จำนวนรูปต่อความเข้มข้น
คาร์โบซัลแฟน	300, 600, 900, 1200, 1500	ppm	10
เกลือ	5, 10, 12, 15, 20	%w/v	10
น้ำตาล	5, 8, 10, 15, 20	%w/v	10
เกลือ-น้ำตาล	(2.5,5),(2.5,7.5),(5,2.5),(5,5),(7.5,5) และ (5,7.5)	%w/v	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ตัวอย่างรูปจากชุดการทดลอง (ก)คาร์โบซิลแฟน (ข)เกลือ (ค)น้ำตาล และ (ง)เกลือ-น้ำตาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แผนภาพแสดงขั้นตอนการเตรียมรูปภาพและแปลงเป็นตัวเลขชุดการทดลอง “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพหุราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต”

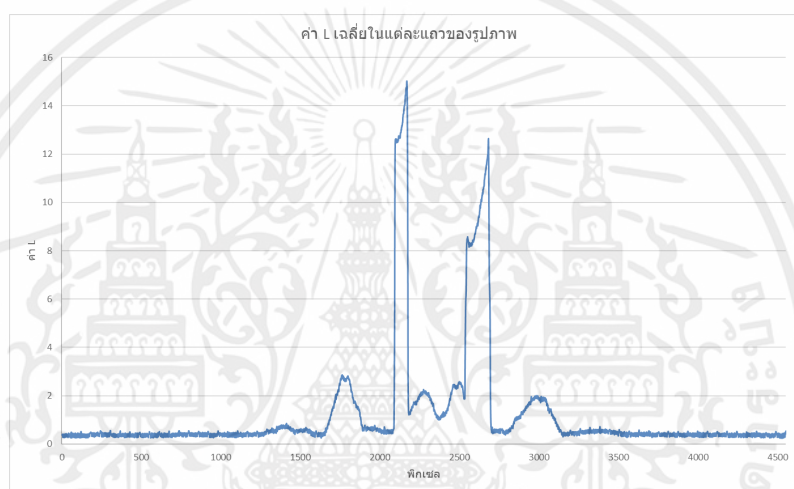
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.1 อ่านรูปจากโฟลเตอร์

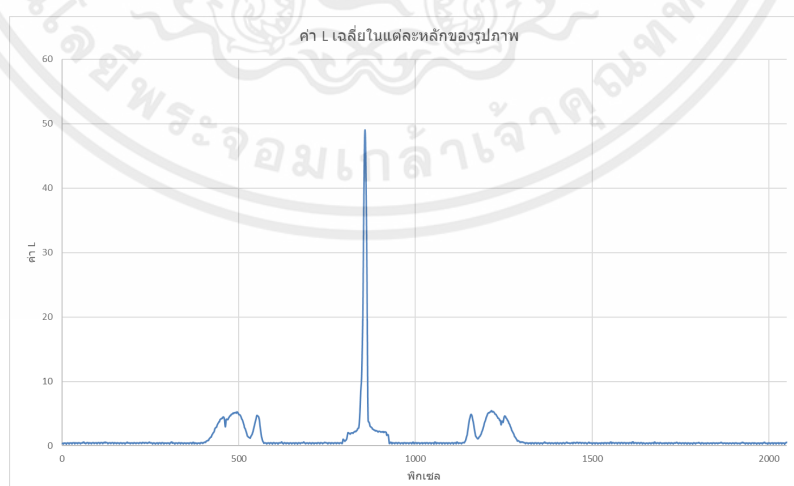
ภายในโฟลเตอร์จะประกอบด้วยชื่อของรูปภาพที่บ่งบอกถึงวันที่ที่ตกลงชนิดของแหล่งกำเนิดแสง ชื่อของสารที่ต้องการจะทำนายความเข้มข้น ชื่อของตัวทำละลาย ความเข้มข้นของสารละลาย และรูปภาพสเปกตรัมของสารนั้น โดยจะแบ่งเป็นรูปภาพของสารละลายที่รู้ความเข้มข้น และสารละลายไร้สิ่งตัวอย่าง

3.1.2.2 การเลือกพื้นที่แถบสี

ในขั้นตอนแรกของการเลือกตัดรูปภาพ จะมีการวิเคราะห์รูปภาพก่อนว่ามีบริเวณใดบ้างที่มีแถบสีที่จำเป็นต่อการวิเคราะห์ โดยเลือกการเฉลี่ยรูปในแต่ละแถวและแต่ละหลัก ดังรูปที่ 3.8 และ 3.9



รูปที่ 3.8 กราฟแสดงค่า L เฉลี่ยในแต่ละแถวของรูปภาพ



รูปที่ 3.9 กราฟแสดงค่า L เฉลี่ยในแต่ละหลักของรูปภาพ

จากภาพจะเห็นได้ว่าจะมีการแบ่งค่าอย่างชัดเจนว่าส่วนที่เป็นพื้นหลังจะมีค่าที่ใกล้เคียง 0 และส่วนที่มีแถบสีจะมีค่ามากกว่า 0 ผู้วิจัยจึงนำจุดที่บ่งบอกถึงแถบสีอ้างอิงมาเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนด และสร้างกรอบสีเหลี่ยมที่มีจุดศูนย์กลางเป็นจุดอ้างอิง เพื่อเลือกแค่เพียงบริเวณที่มีแถบสี
อ้างอิงได้ดังรูปที่ 3.10



รูปที่ 3.10 รูปภาพที่ผ่านการเลือกพื้นที่แถบสีอ้างอิง

3.1.2.3 หาพิกัดมุมของกรอบสีเหลี่ยมแถบสีแสงฟุ้งที่ไม่มีสารละลาย

เมื่อได้ภาพที่มีเฉพาะส่วนของแถบสีอ้างอิงแล้ว ผู้วิจัยได้เลือกใช้การเปลี่ยนจากระบบสี RGB เป็น Greyscale เพื่อให้มีการไล่ระดับสีจากดำถึงขาว แล้วใช้เทคนิคสัญญาณวิทยาเพื่อแบ่งรูปภาพที่สนใจออกจากพื้นหลัง โดยจะใช้ 2 กระบวนการ ประกอบด้วย การเปิดซึ่งเหมาะสมกับการลดแสงรบกวนในบริเวณโดยรอบของรูปภาพที่สนใจ และแบบปิดซึ่งเหมาะสมกับการปะจุดที่มีดในรูปภาพที่เราสนใจเพื่อให้ขอบของรูปภาพมีความคมชัด โดยมีการตั้งเกณฑ์ในเทคนิคสัญญาณวิทยาว่าหากในรูปภาพมีค่า Greyscale มากกว่า 6 จะถูกเปลี่ยนให้เป็นสีขาว และหากน้อยกว่าจะถูกเปลี่ยนให้เป็นสีดำ แสดงเป็นลำดับจากรูปที่ 3.11 ถึง 3.14



รูปที่ 3.11 รูปภาพหลังจากเปลี่ยนระบบสีจาก RGB เป็น Greyscale



รูปที่ 3.12 รูปภาพหลังจากตั้งเกณฑ์ของค่า Greyscale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 รูปภาพหลังจากใช้เทคนิคสัณฐานวิทยาแบบเปิด



รูปที่ 3.14 รูปภาพหลังจากใช้เทคนิคสัณฐานวิทยาแบบปิด

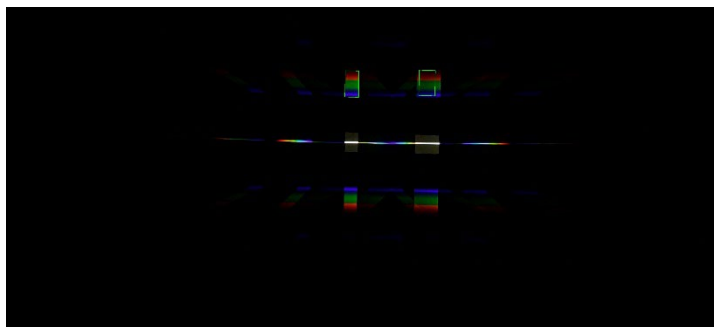
เมื่อได้รูปภาพจากเทคนิคสัณฐานวิทยาจึงใช้ฟังก์ชันของ OpenCV ชื่อว่า `findContour` เพื่อหาจุดมุมของแถบสีอ้างอิงซึ่งจะได้ออกมาเป็นพิกัด (x,y) ของมุมทั้ง 4 มุม รวมเป็น 8 พารามิเตอร์

3.1.2.4 เฉลี่ยความสว่างของแถบสีแสงฝั่งที่ไม่มีสารละลาย

เมื่อได้รูปภาพเฉพาะในส่วนของแถบสีอ้างอิงแล้วจึงเฉลี่ยค่าความสว่างหรือค่า L โดยตั้งสมมุติฐานว่า”ความสว่างของแถบสีอ้างอิงต้องมีความสว่างของแต่ละสีสม่ำเสมอใกล้เคียงกัน และมีแนวโน้มของความสว่างเดียวในทุกชุดการทดลอง” โดยจะเฉลี่ยความสว่างของทั้งรูปแถบสีอ้างอิงทุกรูป แล้วนำไปเทียบกับความสว่างของแถบสีอ้างอิงแต่ละรูปเพื่อหาค่าที่จำเป็นสำหรับการปรับความสว่างแต่ละรูปที่ต้องการ เพื่อลดความคลาดเคลื่อนจากค่าความสว่างที่อาจไม่คงที่

3.1.2.5 แปลงพิกัดมุมที่มี 8 พารามิเตอร์ ให้เหลือ 5 พารามิเตอร์

จากขั้นตอนที่ 3.1.2.3 ที่ได้ข้อมูลกำหนดมุมทั้งหมด 8 พารามิเตอร์ ผู้วิจัยจึงลดจำนวนของพารามิเตอร์ลงเหลือ 5 พารามิเตอร์ ประกอบด้วย พิกัด (x,y) ของจุดศูนย์กลางสี่เหลี่ยม ความกว้าง ความยาว แล้วมุมที่กระทำกับแนวระดับ เพื่อความแม่นยำในการฉายจากแถบสีอ้างอิงไปยังแถบสีที่ต้องการวิเคราะห์ ดังรูปที่ 3.15



รูปที่ 3.15 รูปภาพแสดงตำแหน่งกรอบสี่เหลี่ยมที่แถบสีอ้างอิง

3.1.2.6 ย้ายจุดกึ่งกลางของสี่เหลี่ยมแถบสีแสงฝั่งที่ไม่มีสารละลาย ไปอยู่ในฝั่งที่มีสารละลาย

เมื่อได้ 5 พารามิเตอร์ที่บ่งบอกถึงตำแหน่งและลักษณะของแถบสีอ้างอิง แล้วจึงทำการฉายกรอบสี่เหลี่ยมจากแถบสีอ้างอิงไปยังแถบสีที่ต้องการวิเคราะห์ โดยฉายให้จุดกึ่งกลางจากแถบสีอ้างอิงไปอยู่บริเวณแถบสีที่ต้องการวิเคราะห์ด้วยมุมที่ทำกับแนวระดับของสี่เหลี่ยม โดยเลื่อนจากระยะห่างคงที่ระหว่างแถบสีฝั่งซ้ายและฝั่งขวาตามลักษณะทางกายภาพของเครื่อง ดังรูปที่ 3.16



รูปที่ 3.16 รูปภาพแสดงตำแหน่งกรอบสี่เหลี่ยมจากแถบสีอ้างอิงและตำแหน่งกรอบสี่เหลี่ยมที่เกิดจากการฉาย

3.1.2.7 สร้างกรอบสี่เหลี่ยมใหม่ ปรับเปลี่ยนมุมมองภาพและเลื่อนตำแหน่ง

เมื่อย้ายจุดกึ่งกลางจากของสี่เหลี่ยมแถบสีแสงฝั่งที่ไม่มีสารละลาย ไปอยู่ในฝั่งที่มีสารละลาย จึงสร้างกรอบสี่เหลี่ยมจากจุดกึ่งกลางใหม่ด้วยข้อมูล 5 พารามิเตอร์ แล้วนำพิกัดมุมของสี่เหลี่ยมที่ได้จากการฉายภาพจากแถบสีอ้างอิงไปเข้าฟังก์ชัน Warp perspective จากชุดคำสั่ง OpenCV และกำหนดขนาดรูปภาพเป็น 100 x 84 พิกเซล ได้ผลลัพธ์ดังรูปที่ 3.17



รูปที่ 3.17 ตัวอย่างภาพหลังการปรับเปลี่ยนมุมมองชุดการทดลองคาร์โบซิลแฟน “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต”

เมื่อใช้ภาพหลังการปรับเปลี่ยนมุมมองมาวิเคราะห์พบว่าตำแหน่งของสีภายในภาพมีความเคลื่อนไหวไปหลายพิกเซลทำให้เกิดกรณีความสว่างของสีที่มีต่างกันมากในสารที่ต้องการทราบความเข้มข้นกับสีในสารละลายไร้ตัวอย่าง จึงมีการสร้างฟังก์ชันสำหรับเลื่อนแถบสีให้ตรงกัน โดยกำหนดจุดอ้างอิงจากจุดที่มีแสงน้อยที่สุดในภาพให้เป็นตำแหน่งอ้างอิง และเลื่อนพิกเซลของภาพให้ตรงกันตามตำแหน่งอ้างอิงเป็นผลลัพธ์ของภาพก่อนการแปลงรูปภาพเป็นตัวเลข

3.1.2.8 เก็บค่าชื่อของรูปภาพ และแปลงข้อมูลรูปภาพเป็นตัวเลข

เมื่อรูปภาพมีความเหมาะสมแก่การวิเคราะห์แล้ว จึงเก็บค่าชื่อของรูปภาพ และแปลงรูปภาพเป็นตัวเลขด้วยค่า L จากระบบสี LAB เพื่อแสดงถึงค่าความสว่างของพิกเซลนั้นๆ และนำค่า L ของสารละลายไร้สิ่งตัวอย่างลบด้วยค่า L ของรูปสารละลายที่รู้ความเข้มข้น ด้วยชุดคำสั่ง Pandas, NumPy และ OpenCV

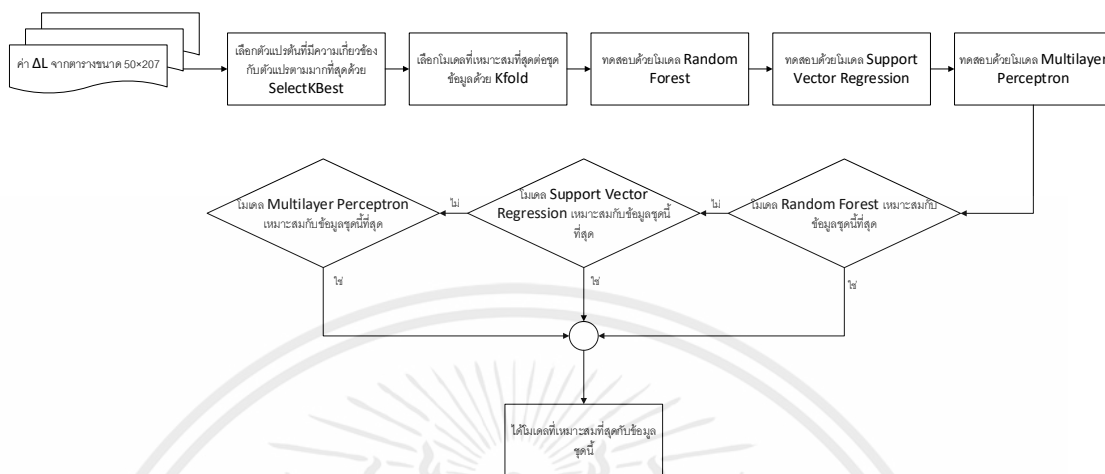
เริ่มต้นใช้ชุดคำสั่ง OpenCV และ Pandas อ่านค่าสีในแต่ละพิกเซลและเก็บเป็น ดาต้าเฟรม ขนาด 100x84 ตามขนาดของภาพ ก่อนจะใช้ NumPy และ Pandas ในการเฉลี่ยค่า L ในแต่ละแถวเพื่อลดสปีรบกวน ได้ผลลัพธ์เป็น ดาต้าเฟรม ขนาด 1x100 และใช้กระบวนการดังกล่าวกับทุกๆ แถบสีสำหรับภาพถ่ายในชุดทดลองด้วยภาษา Python กับชุดคำสั่ง Pandas ได้เป็น ดาต้าเฟรม ขนาด 60x100 ซึ่งมีแถวและหลักเป็น ความเข้มข้นของสารซึ่งมีจำนวนเท่ากับรูปภาพ และค่า L ในแต่ละพิกเซลตามลำดับ สุดท้ายนำ ดาต้าเฟรม ดังกล่าวไปลบจากค่า L ของสารละลายไร้สิ่งตัวอย่างในทุกๆ หลักเป็น ดาต้าเฟรม ขนาด 60x100 ซึ่งมีแถวและหลักเป็น ความเข้มข้นของสาร และค่า ΔL ของสารสารละลายไร้สิ่งตัวอย่างและสารละลายที่รู้ความเข้มข้น

3.2 การเรียนรู้ของเครื่องและการเลือกใช้โมเดล

เมื่อได้รับข้อมูลตัวเลขที่แสดงถึงตัวแปรต้นและตัวแปรตามหรือ ค่า ΔL ของสารจำนวน 154 พิกเซล และความเข้มข้นของสารตามลำดับ สามารถนำข้อมูลไปเข้าสู่การเรียนรู้ของเครื่องได้โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วยขั้นตอนดังนี้ การเลือกชุดข้อมูลตัวแปรต้นด้วย SelectKBest, การสร้างความเท่าเทียมให้กับโมเดลด้วย K-fold, การทำนายความเข้มข้นด้วยการเรียนรู้ของเครื่อง ซึ่งสรุปเป็นผังงานได้ดังรูปที่ 3.18



รูปที่ 3.18 ผังงานการเรียนรู้ของเครื่องและการเลือกใช้โมเดล

3.2.1 การเลือกชุดข้อมูลตัวแปรต้นด้วย SelectKBest

เมื่อได้ข้อมูลตัวเลขที่แบ่งเป็นตัวแปรต้นคือค่า ΔL และ ตัวแปรตามคือความเข้มข้นของสารละลาย แต่เนื่องจากว่าค่า ΔL นั้นมีถึง 100 ค่าต่อหนึ่งรูป ตามหลักการพื้นฐานของการเรียนรู้ของเครื่องคือจำเป็นต้องมีชุดข้อมูลอย่างน้อยสิบเท่าของจำนวนตัวแปรต้น ซึ่งนั่นทำให้จำเป็นต้องมีการทดลองอย่างน้อย 1000 การทดลองที่ความเข้มข้นต่างกัน จึงได้มีการใช้ชุดคำสั่งจาก sklearn ชื่อ “SelectKBest” โดยใช้วิธีการทางสถิติคือค่า “f_regression” เพื่อหาความสัมพันธ์ระหว่างตัวแปรต้นและตัวแปรตามโดยผู้วิจัยเลือกมาทั้งหมด 4 ข้อมูลจากค่าความสัมพันธ์ที่มากที่สุด 4 ลำดับแรก

3.2.2 การสร้างความเท่าเทียมให้กับโมเดลด้วย K-fold

เนื่องด้วยการทำการทำนายความเข้มข้นโดยการใช้การเรียนรู้ของเครื่อง มีโมเดลหลากหลาย สำหรับในงานวิจัยนี้ได้แก่ Random forest, support vector machine และ Multilayer perceptron ซึ่งในแต่ละโมเดลมีกระบวนการให้ได้มาซึ่งผลลัพธ์ที่ต่างกัน จึงเรียกใช้ชุดคำสั่ง sklearn ชื่อ K-fold มาสร้างความเท่าเทียมให้กับโมเดล ซึ่งเป็นกระบวนการที่ใช้เลือกชุดข้อมูลสำหรับเรียนรู้ให้เหมาะกับชนิดโมเดลแต่ละแบบ เพื่อให้การเปรียบเทียบประสิทธิภาพของโมเดลมีความเท่าเทียม โดยเลือก $K = 5$

3.2.3 การทำนายความเข้มข้นด้วยการเรียนรู้ของเครื่อง

หลังจากที่สร้างความเท่าเทียมให้กับแต่ละโมเดลแล้วนั้นจึงทำการฝึกโมเดลให้รู้ความสัมพันธ์ระหว่างตัวแปรต้นและตัวแปรตาม หลังจากนั้นจึงนำข้อมูลที่โมเดลไม่เคยเห็นมาก่อนมาทดสอบประสิทธิภาพ โดยแต่ละโมเดลมีการตั้งค่าเพื่อหา Hyperparameter ที่ดีที่สุดดังตารางที่ 3.3

ตารางที่ 3.3 การปรับ Hyperparameter ของแต่ละโมเดล

Model	การตั้งค่า
SVR	'C': [0.1, 1, 10, 100, 1000], 'kernel': ['linear', 'rbf', 'sigmoid']
RF	'n_estimators': [50, 100, 200, 300, 400, 500], 'max_depth': [None, 10, 20, 30, 40, 50], 'min_samples_split': [2, 5, 10, 15, 20]
MLP	'hidden_layer_sizes': [(50, 50), (75, 75), (100,100), (25,25)], 'activation': ['relu', 'tanh'], 'alpha': [0.0001, 0.001, 0.01]
XGBoost	'n_estimators': [100, 200, 300, 400, 500], 'max_depth': [3, 4, 5, 6, 7, 8], 'learning_rate': [0.01, 0.05, 0.1, 0.5, 1]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

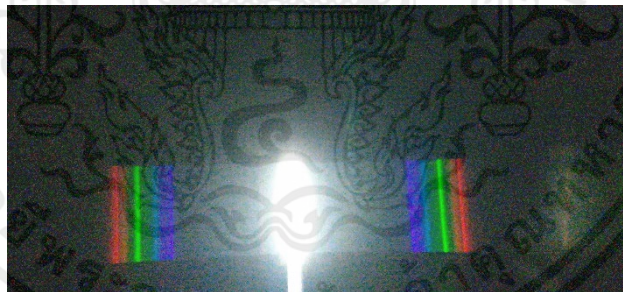
ผลการทดลองและวิเคราะห์ผลการทดลอง

ในส่วนผลการทดลองและวิเคราะห์ผลการทดลองเป็นการนำผลลัพธ์ที่ได้หลังกระบวนการเตรียมรูปภาพและแปลงเป็นตัวเลขผ่านส่วนการเรียนรู้ของเครื่อง โมเดลของการเรียนรู้ของเครื่องที่ใช้ในการทดลองจะประกอบด้วยโมเดลจากสองชุดคำสั่ง คือ Support Vector Regression (SVR), Random Forest (RF), Multilayer Perceptron (MLP) จากชุดคำสั่ง Scikit-Learn และ Extreme Gradient Boosting จากชุดคำสั่ง XGBoost ซึ่งทุกโมเดลมีการเลือกจำนวนของตัวแปรต้นด้วยชุดคำสั่ง Selectkbest, การแบ่งจำนวนของชุดข้อมูลเป็น Train Set และ Test Set และการปรับ Hyperparameter ของ Model ด้วยการทำ Cross Validation ผ่านชุดคำสั่ง Kfold และ Gridsearchcv มารายงานผลลัพธ์และวิเคราะห์ผลที่ได้จากกระบวนการข้างต้น

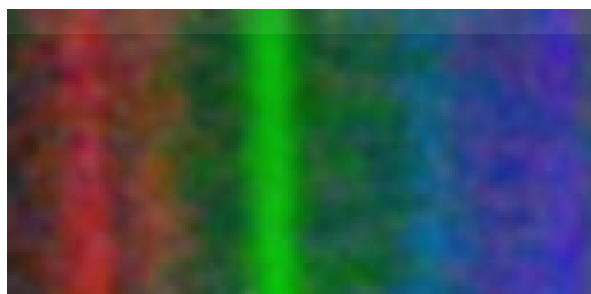
4.1 ผลการทดลองข้อมูลจากงานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล”

4.1.1 ผลจากกระบวนการเตรียมรูปภาพ

จากกระบวนการเตรียมรูปภาพเพื่อให้ได้รูปที่เหมาะสมในการวิเคราะห์ ด้วยกระบวนการดังรูปที่ 3.1.1 มีรูปภาพตัวอย่างก่อนกระบวนการดังรูปที่ 4.1



รูปที่ 4.1 รูปภาพสารละลายโพแทสเซียมเปอร์แมงกาเนต 5 ppm ก่อนกระบวนการ และรูปภาพตัวอย่างหลังกระบวนการดังรูปที่ 4.2

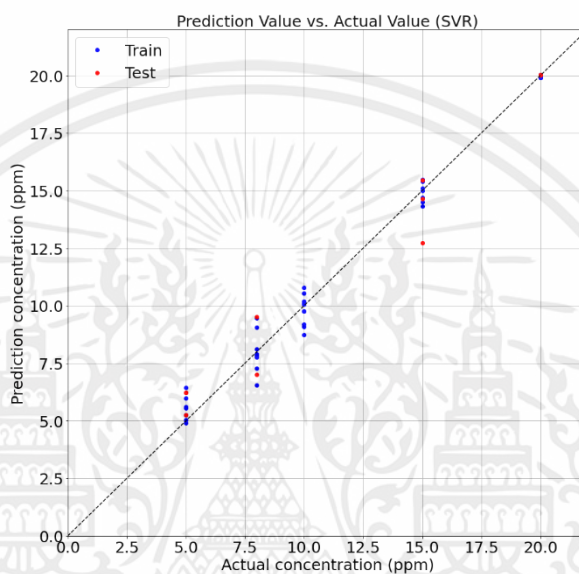


รูปที่ 4.2 รูปภาพสารละลายโพแทสเซียมเปอร์แมงกาเนต 5 ppm หลังกระบวนการ

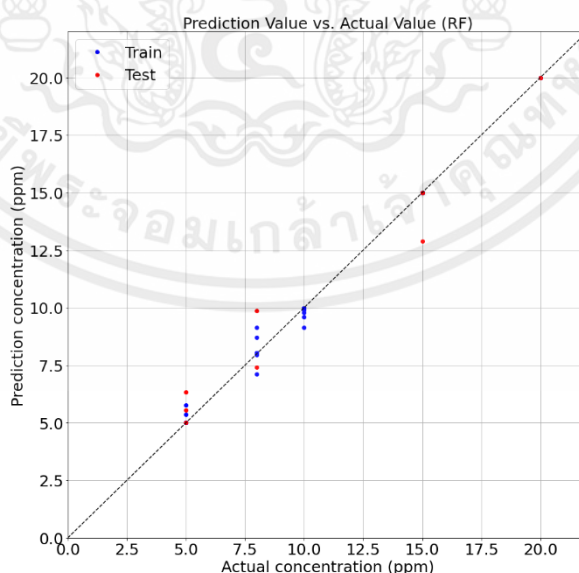
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลจากการประยุกต์ใช้การเรียนรู้ของเครื่อง

ผลการทดลองการพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมต โดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล ชุดข้อมูลโพแทสเซียมเปอร์แมงกาเนต (KMnO_4) จาก 3 โมเดล ได้แก่ SVR, RF และ MLP กำหนด SelectKbest; $k = 5$, KFold; $k = 5$ ผ่านการปรับ Hyperparameter ด้วย GridSearchCV กำหนดเกณฑ์การปรับด้วย Mean Absolute Error (MAE) ได้ผลลัพธ์จากการทำนาย, ตัวแปรต้นที่ใช้ และค่า Hyperparameter ดังรูปที่ 4.3 ถึง 4.5

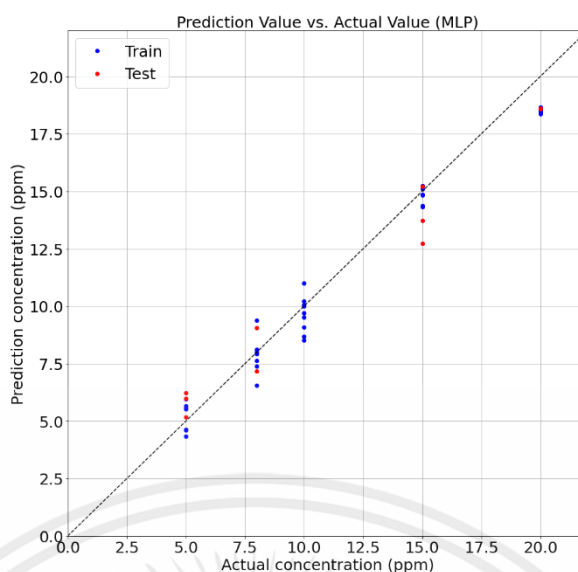


รูปที่ 4.3 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลโพแทสเซียมเปอร์แมงกาเนตจากโมเดล SVR



รูปที่ 4.4 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลโพแทสเซียมเปอร์แมงกาเนตจากโมเดล RF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลโพแทสเซียมเปอร์แมงกาเนตจากโมเดล MLP

จากการทำ cross-validation และ GridsearchCV แต่ละโมเดลมีการใช้ Hyperparameter ดังนี้ โมเดล SVR ใช้ kernel = rbf และ C = 10 โมเดล RF ใช้ n_estimators = 50, min_samples_split = 5 และ max_depth = 10 และโมเดล MLP ใช้ hidden_layer_sizes = (50, 50), alpha = 0.001 และ activation = tanh และทุกโมเดลใช้ pixel ที่ 85-89 เป็นตัวแปรต้น จากรูปที่ 4.3 ถึง 4.5 สามารถจัดรวม Evaluation metrics เป็นตารางได้ดังตารางที่ 4.1

ตารางที่ 4.1 Evaluation metrics ของแต่ละโมเดล

Evaluation Matrix \ Model	MAPE (%)		MAE		R-Square		MSE	
	Train	Test	Train	Test	Train	Test	Train	Test
SVR	5.39	10.58	0.43	0.83	0.99	0.96	0.38	1.18
RF	1.52	6.03	0.12	0.5	1.00	0.98	0.10	0.73
MLP	7.59	12.60	0.76	1.12	0.97	0.96	0.83	1.47

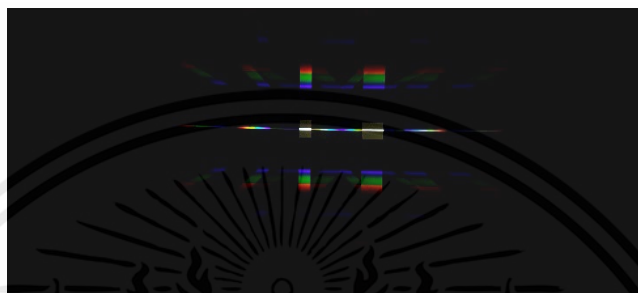
จากตารางพบว่าโมเดลที่มีค่าความคลาดเคลื่อนหรือ Mean Absolute Percentage Error น้อยที่สุดคือโมเดล RF ซึ่งมีความคลาดเคลื่อนอยู่ที่ 1.52% ใน train set และ 6.03% ใน test set

4.2 ผลการทดลองข้อมูลจากงานวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต”

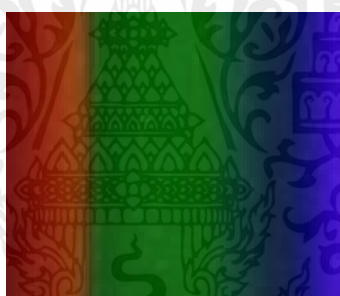
4.2.1 ชุดการทดลองสารละลายคาร์โบซัลแฟน

4.2.1.1 ผลจากกระบวนการเตรียมรูปภาพ

จากกระบวนการเตรียมรูปภาพเพื่อให้ได้รูปที่เหมาะสมในการวิเคราะห์ด้วยกระบวนการดังบทที่ 3.1.2 มีรูปภาพตัวอย่างก่อนกระบวนการดังรูปที่ 4.6



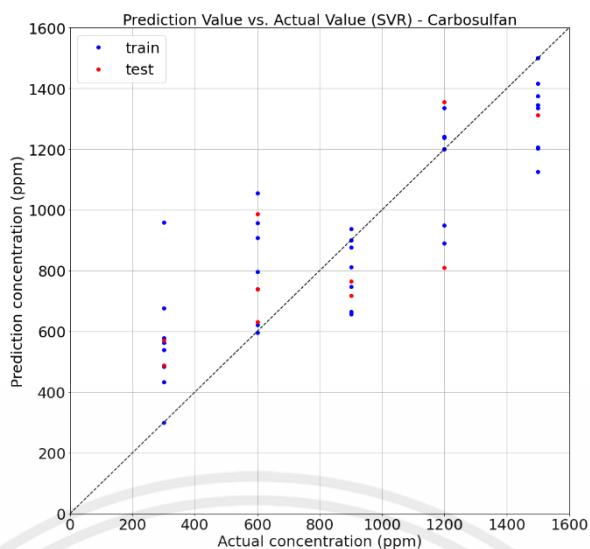
รูปที่ 4.6 รูปภาพสารละลายคาร์โบซัลแฟน 300 ppm ก่อนกระบวนการ
และรูปภาพหลังกระบวนการดังรูปที่ 4.7



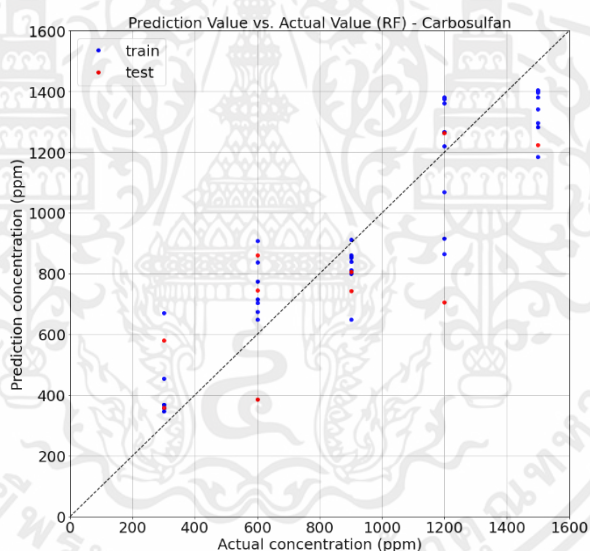
รูปที่ 4.7 รูปภาพสารละลายคาร์โบซัลแฟน 300 ppm หลังกระบวนการ

4.2.1.2 ผลจากการประยุกต์ใช้การเรียนรู้ของเครื่อง

ผลการทดลองการพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต ชุดข้อมูลคาร์โบซัลแฟน (Carbosulfan) จาก 4 โมเดล ได้แก่ SVR, RF, MLP และ XGBoost กำหนด SelectKbest; k = 4, KFold; k = 10 ผ่านการปรับ Hyperparameter ด้วย GridSearchCV กำหนดเกณฑ์การปรับด้วย Mean Absolute Percentage Error (MAPE) ได้ผลลัพธ์จากการทำนาย, ตัวแปรต้นที่ใช้ และค่า Hyperparameter ดังรูปที่ 4.8 ถึง 4.11

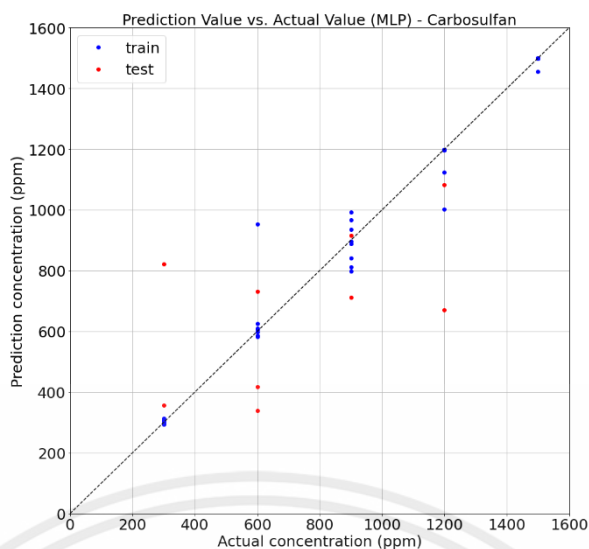


รูปที่ 4.8 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลคาร์โบซัลแฟนจากโมเดล SVR

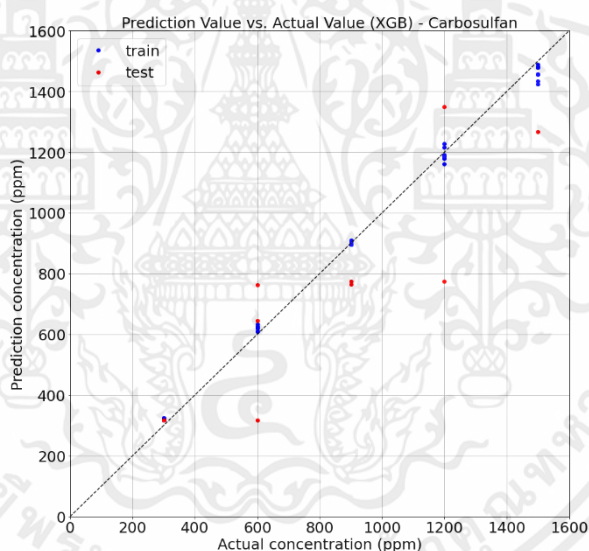


รูปที่ 4.9 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลคาร์โบซัลแฟนจากโมเดล RF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลคาร์โบซัลแฟนจากโมเดล MLP



รูปที่ 4.11 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลคาร์โบซัลแฟนจากโมเดล XGBoost

จากการทำ cross-validation และ GridsearchCV แต่ละโมเดลมีการใช้ Hyperparameter ดังนี้ โมเดล SVR ใช้ kernel = 'rbf' และ C = 1000 โมเดล RF ใช้ n_estimators = 300, min_samples_split = 2 และ max_depth = 50 โมเดล MLP ใช้ hidden_layer_sizes = (25, 25), alpha = 0.01 และ activation = relu และโมเดล XGBoost ใช้ n_estimators = 400, max_depth = 6 และ learning_rate = 0.01 และทุกโมเดลใช้ pixel ที่ 96-99 เป็นตัวแปรต้น

จากรูปที่ 4.8 ถึง 4.11 สามารถจัดรวม Evaluation Metrics เป็นตารางได้ดังตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 Evaluation metrics ของแต่ละโมเดล

Evaluation Metrics Model	MAPE (%)		MAE		R-Square		MSE	
	Train	Test	Train	Test	Train	Test	Train	Test
SVR	30.21	33.82	166.52	206.28	0.73	0.63	50835.54	54024.61
RF	11.09	25.75	80.00	173.80	0.94	0.66	11837.45	49971.40
MLP	2.52	22.66	20.47	173.25	0.98	0.70	3788.69	43624.83
XGBoost	2.73	18.52	19.87	159.09	1.00	0.72	632.54	40028.94

จากตารางพบว่าโมเดลที่มีค่าความคลาดเคลื่อนหรือ Mean Absolute Percentage Error น้อยที่สุดคือโมเดล XGBoost ซึ่งมีความคลาดเคลื่อนอยู่ที่ 2.73% ใน train set และ 18.52% ใน test set

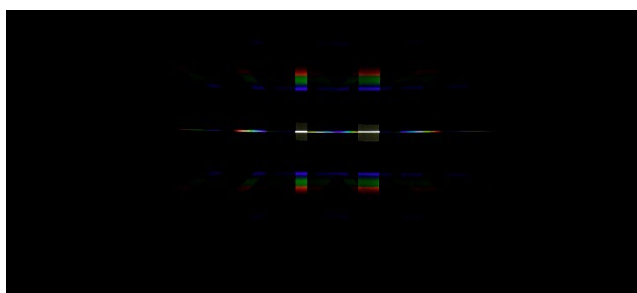
4.2.2 ชุดการทดลองสารละลายเกลือ, น้ำตาล และ เกลือ-น้ำตาล

4.2.2.1 ผลจากกระบวนการเตรียมรูปภาพ

จากกระบวนการเตรียมรูปภาพเพื่อให้ได้รูปที่เหมาะสมในการวิเคราะห์ด้วยกระบวนการดังบทที่ 3.1.2 มีรูปภาพตัวอย่างก่อนกระบวนการดังรูปที่ 4.12 ถึง 4.14

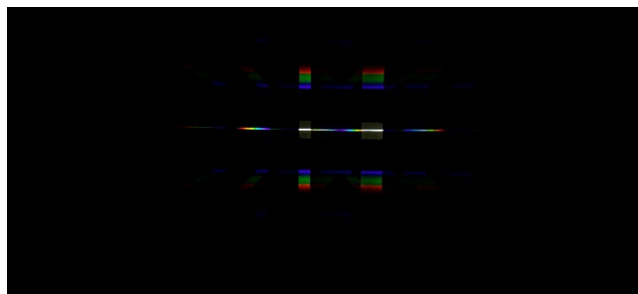


รูปที่ 4.12 สารละลายเกลือ 5 %w/v



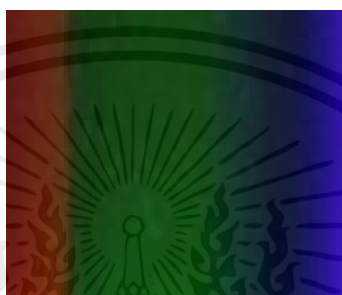
รูปที่ 4.13 สารละลายน้ำตาล 5 %w/v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

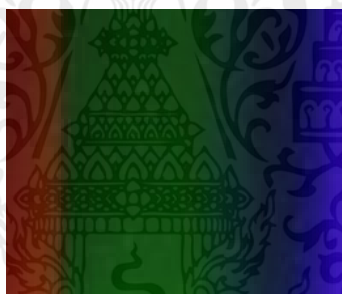


รูปที่ 4.14 สารละลายเกลือ 5 %w/v และ น้ำตาล 2.5 %w/v

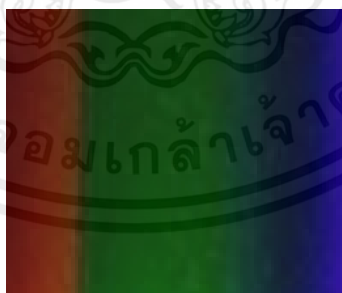
และรูปภาพหลังกระบวนการดังรูปที่ 4.15 ถึง 4.17



รูปที่ 4.15 รูปภาพสารละลายเกลือ 5 %w/v หลังกระบวนการ



รูปที่ 4.16 รูปภาพสารละลายน้ำตาล 5 %w/v หลังกระบวนการ



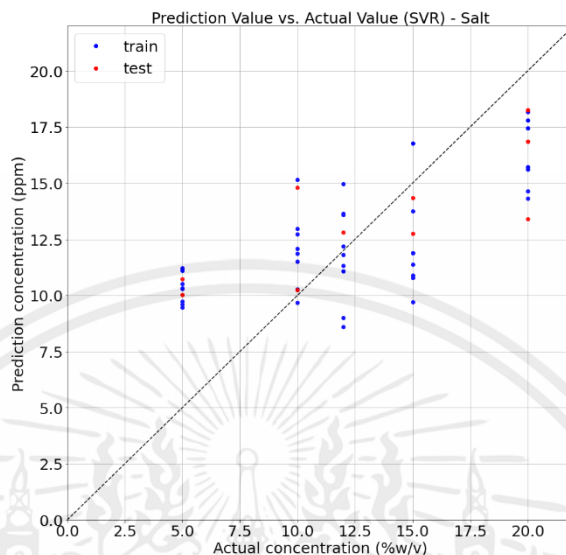
รูปที่ 4.17 รูปภาพสารละลายเกลือ 5 %w/v และ น้ำตาล 2.5 %w/v หลังกระบวนการ

4.2.2.2 ผลจากการประยุกต์ใช้การเรียนรู้ของเครื่อง

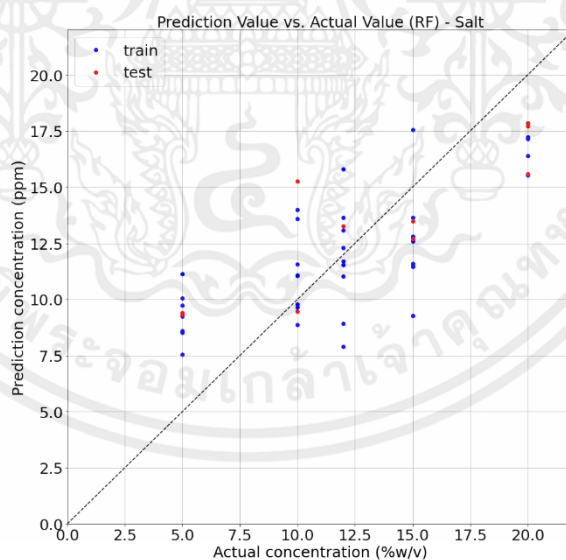
ผลการทดลองการพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารละลายเกลือ จาก 4 โมเดล ได้แก่ SVR, RF, MLP และ XGBoost กำหนด SelectKbest; k = 4, KFold; k = 10 ผ่านการปรับ Hyperparameter ด้วย GridSearchCV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดเกณฑ์การปรับด้วย Mean Absolute Percentage Error (MAPE) ได้ผลลัพธ์จากการทำนาย ,ตัวแปรต้นที่ใช้ และค่า Hyperparameter ดังรูปที่ 4.18 ถึง 4.21

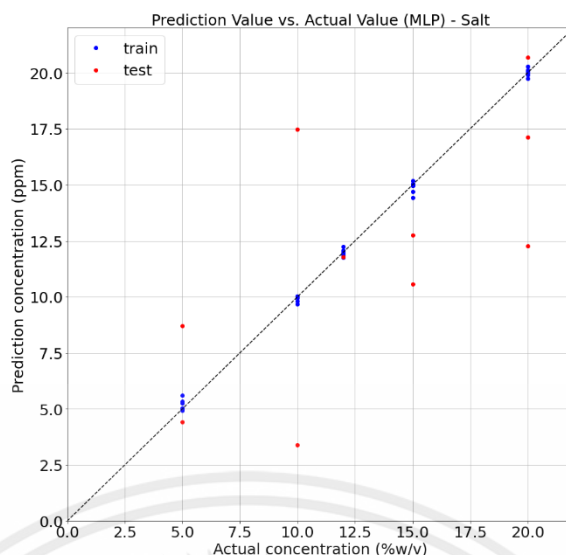


รูปที่ 4.18 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือจาก โมเดล SVR

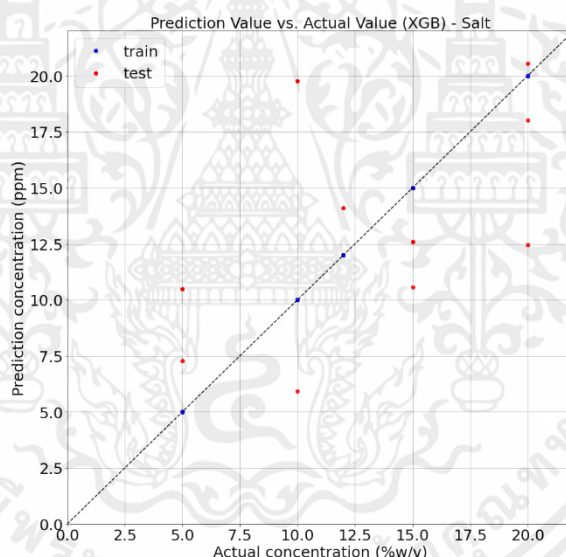


รูปที่ 4.19 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือจาก โมเดล RF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือจากโมเดล MLP



รูปที่ 4.21 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือจากโมเดล XGBoost

จากการทำ Cross-Validation และ GridsearchCV แต่ละโมเดลมีการใช้ Hyperparameter ดังนี้ โมเดล SVR ใช้ kernel = 'sigmoid' และ C = 1 โมเดล RF ใช้ n_estimators = 300, min_samples_split = 2 และ max_depth = 50 โมเดล MLP ใช้ hidden_layer_sizes = (50, 50), alpha = 0.001 และ activation = tanh และโมเดล XGBoost ใช้ n_estimators = 400, max_depth = 5 และ learning_rate = 0.1 และทุกโมเดลใช้ pixel ที่ 41-44 เป็นตัวแปรต้น

จากรูปที่ 4.18 ถึง 4.21 สามารถจัดรวมเป็นตารางได้ดังตารางที่ 4.3

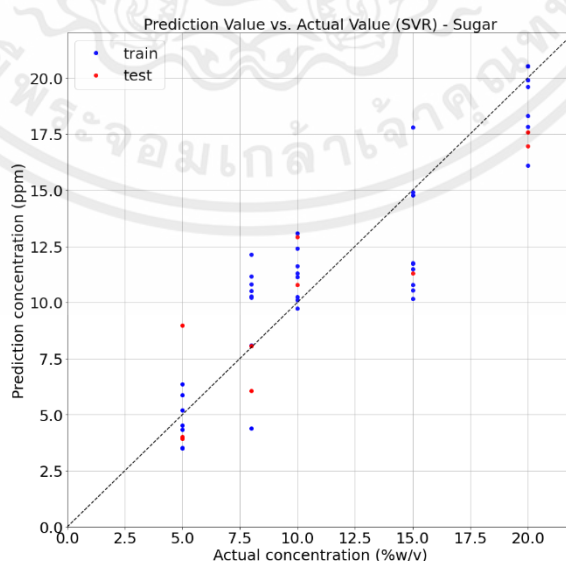
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 Evaluation metrics ของแต่ละโมเดล

Model	Evaluation Metrics		MAPE (%)		MAE		R-Square		MSE	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
SVR	36.10	34.92	3.16	3.10	0.44	0.52	13.21	14.38		
RF	14.88	32.34	1.34	3.14	0.90	0.58	2.44	12.81		
MLP	1.65	35.79	0.17	4.02	1.00	0.15	0.05	25.61		
XGBoost	0.01	40.70	0.00	4.06	1.00	0.21	0.00	23.82		

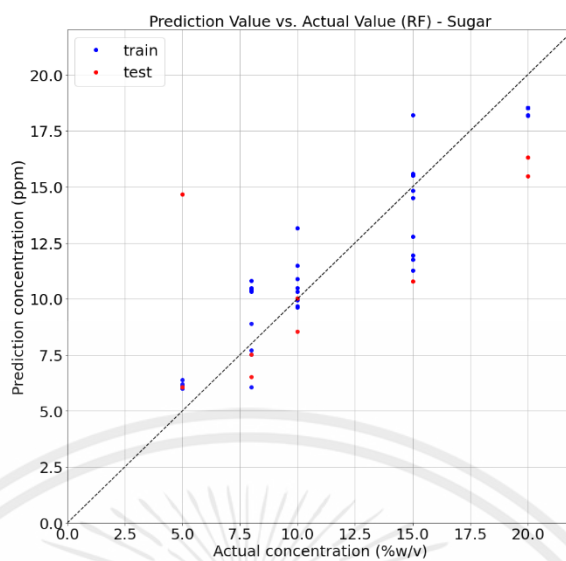
จากตารางพบว่าโมเดลที่มีค่าความคลาดเคลื่อนหรือ Mean Absolute Percentage Error น้อยที่สุดสำหรับการทำนายความเข้มข้นของสารละลายเกลือคือโมเดล RF ซึ่งมีความคลาดเคลื่อนอยู่ที่ 14.88% ใน train set และ 32.34% ใน test set

ผลการทดลองการพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารละลายน้ำตาล จาก 4 โมเดล ได้แก่ SVR, RF, MLP และ XGBoost กำหนด SelectKbest; k = 4, KFold; k = 10 ผ่านการปรับ Hyperparameter ด้วย GridSearchCV กำหนดเกณฑ์การปรับด้วย Mean Absolute Percentage Error (MAPE) ได้ผลลัพธ์จากการทำนาย ,ตัวแปรต้นที่ใช้ และค่า Hyperparameter ดังรูปที่ 4.22 ถึง 4.25

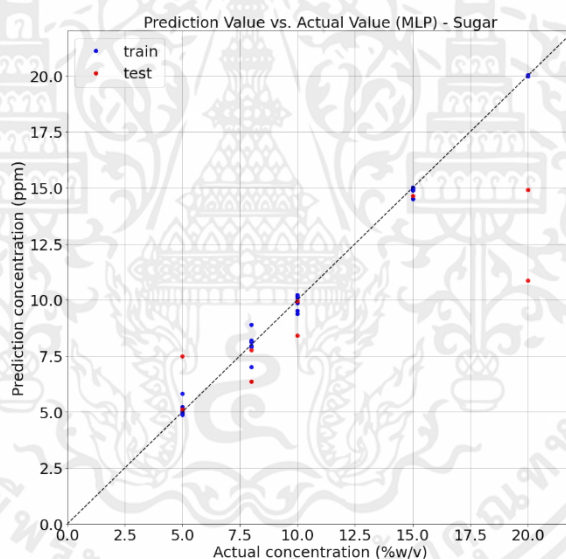


รูปที่ 4.22 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลน้ำตาลจากโมเดล SVR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

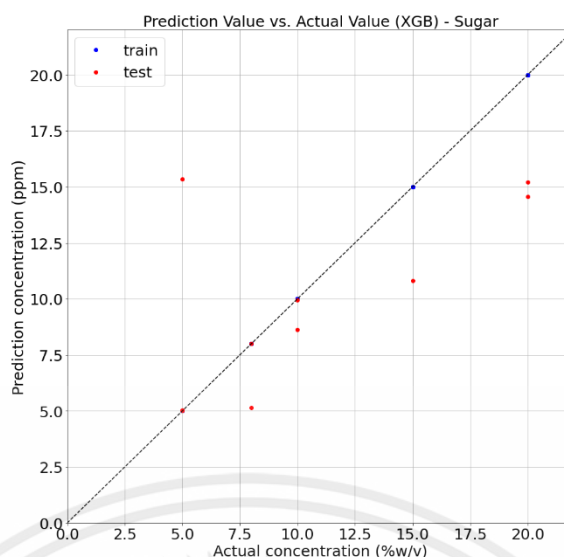


รูปที่ 4.23 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลน้ำตาลจาก
โมเดล RF



รูปที่ 4.24 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลน้ำตาลจาก
โมเดล MLP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลน้ำตาลจากโมเดล XGBoost

จากการทำ Cross-Validation และ Gridsearchcv แต่ละโมเดลมีการใช้ Hyperparameter ดังนี้ โมเดล SVR ใช้ Kernel = 'Linear' และ C = 100 โมเดล RF ใช้ n_estimators = 300, min_samples_split = 2 และ max_depth = 50 โมเดล MLP ใช้ hidden_layer_sizes = (100, 100), alpha = 0.001 และ activation = tanh และโมเดล XGBoost ใช้ n_estimators = 400, max_depth = 5 และ learning_rate = 0.1 และทุกโมเดลใช้ pixel ที่ 18-20 และ 80 เป็นตัวแปรต้น

จากรูปที่ 4.22 ถึง 4.25 สามารถจัดรวมเป็นตารางได้ดังตารางที่ 4.4

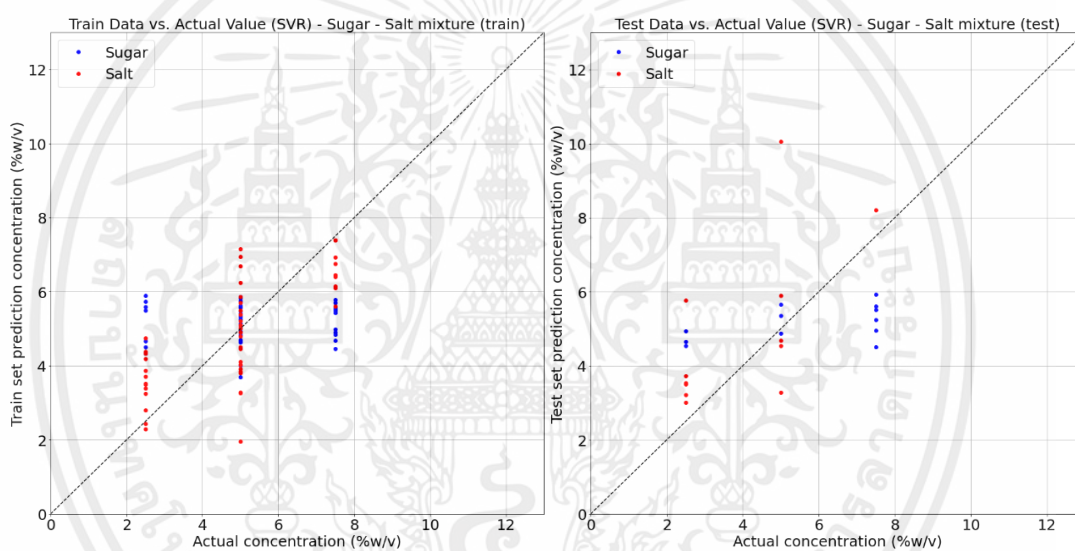
ตารางที่ 4.4 Evaluation metrics ของแต่ละโมเดล

Model	Evaluation Metrics		MAPE (%)		MAE		R-Square		MSE	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
SVR	17.92	23.45	1.84	2.09	0.80	0.80	5.47	5.96		
RF	8.17	31.34	0.85	2.80	0.96	0.53	1.10	14.44		
MLP	3.76	21.65	0.30	2.25	0.99	0.71	0.17	8.98		
XGBoost	0.01	33.77	0.00	2.92	1.00	0.38	0.00	18.76		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

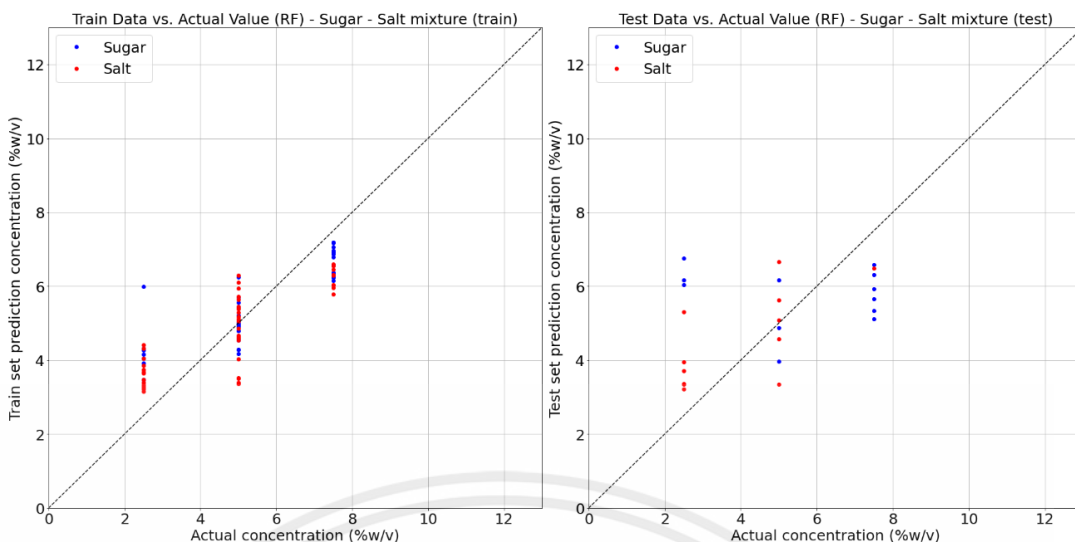
จากตารางพบว่าโมเดลที่มีค่าความคลาดเคลื่อนหรือ Mean Absolute Percentage Error น้อยที่สุดสำหรับการทำนายความเข้มข้นของสารละลายน้ำตาลคือโมเดล MLP ซึ่งมีความคลาดเคลื่อนอยู่ที่ 3.76% ใน Train Set และ 21.65% ใน Test Set

ผลการทดลองการพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารละลายเกลือ-น้ำตาลจาก 4 โมเดล ได้แก่ SVR, RF, MLP และ XGBoost กำหนด SelectKbest; k = 4, KFold; k=10 ผ่านการปรับ Hyperparameter ด้วย GridSearchCV กำหนดเกณฑ์การปรับด้วย Mean Absolute Percentage Error (MAPE) ได้ผลลัพธ์จากการทำนาย,ตัวแปรต้นที่ใช้ และค่า Hyperparameter ดังรูปที่ 4.26 ถึง 4.29

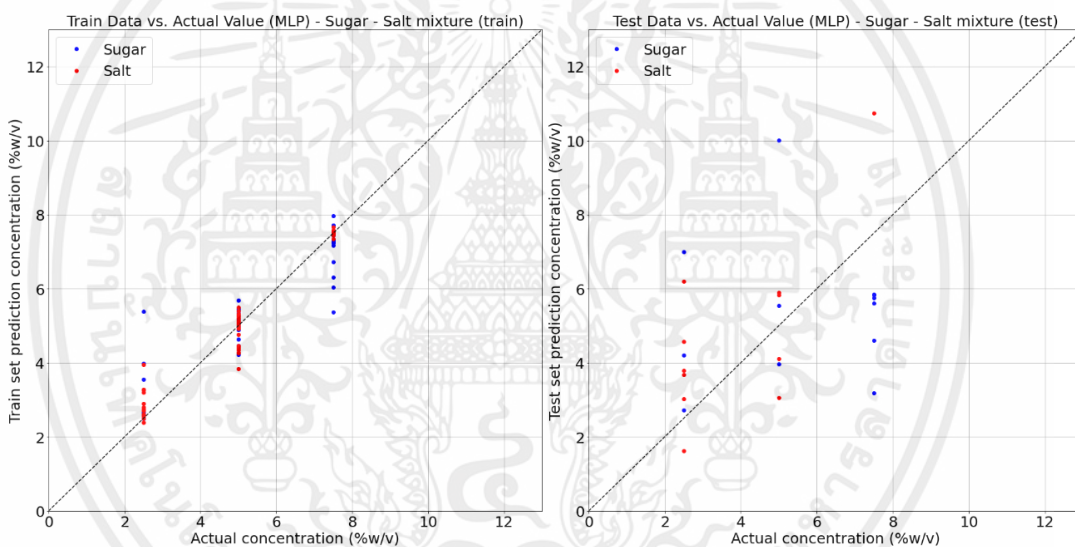


รูปที่ 4.26 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือ-น้ำตาลจากโมเดล SVR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

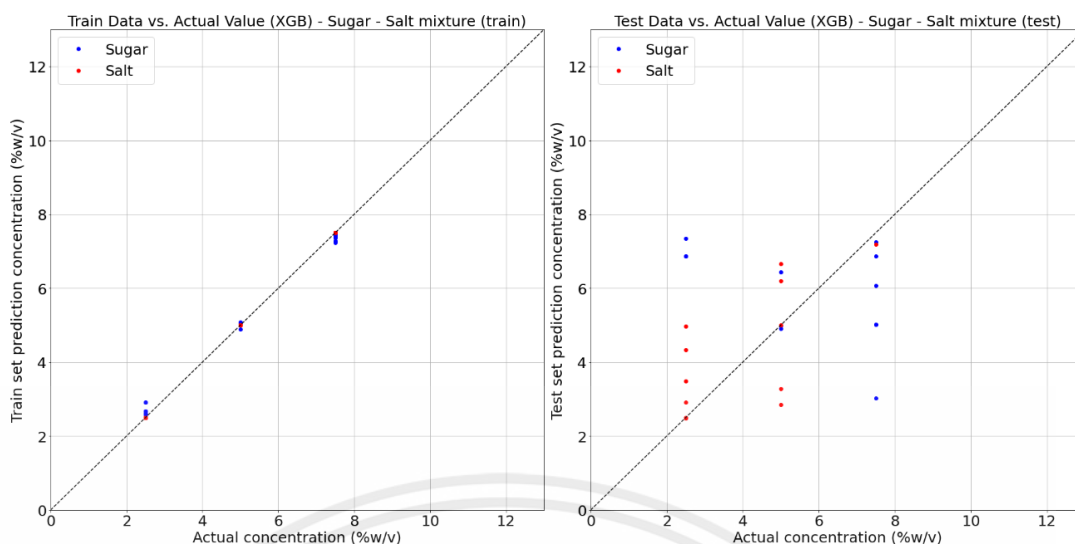


รูปที่ 4.27 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือ-น้ำตาลจากโมเดล RF



รูปที่ 4.28 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือ-น้ำตาลจากโมเดล MLP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 กราฟเปรียบเทียบระหว่างค่าที่ทำนายและค่าจริงของ Train set และ Test set ของชุดข้อมูลเกลือ-น้ำตาลจากโมเดล XGBoost

จากการทำ Cross-Validation และ Gridsearchcv แต่ละโมเดลมีการใช้ Hyperparameter ดังนี้ โมเดล SVR ใช้ Kernel = 'Sigmoid' และ C = 1 โมเดล RF ใช้ n_estimators = 50, min_samples_split = 10 และ max_depth = 10 โมเดล MLP ใช้ hidden_layer_sizes = (50, 50), alpha = 0.0001 และ activation = tanh และโมเดล XGBoost ใช้ n_estimators = 400, max_depth = 6 และ learning_rate = 0.01 และทุกโมเดลใช้ pixel ที่ 18-29, 50 และ 51 เป็นตัวแปรต้น

จากรูปที่ 4.26 ถึง 4.29 สามารถจัดรวมเป็นตารางได้ดังตารางที่ 4.5

ตารางที่ 4.5 Evaluation Metrics ของแต่ละโมเดล

Evaluation Metrics	MAPE (%)		MAE		R-Square		MSE	
	Train	Test	Train	Test	Train	Test	Train	Test
SVR	27.15	39.77	1.14	1.58	0.22	-0.18	2.10	3.80
RF	21.38	45.43	0.85	1.62	0.58	-0.02	1.15	3.79
MLP	11.11	50.11	0.45	2.09	0.85	-0.88	0.44	7.37
XGBoost	0.90	45.81	0.03	1.65	1.00	-0.32	0.01	5.05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางพบว่าโมเดลที่มีค่าความคลาดเคลื่อนหรือ Mean Absolute Percentage Error น้อยที่สุดสำหรับการทำนายความเข้มข้นของสารละลายน้ำตาลคือโมเดล SVR ซึ่งมีความคลาดเคลื่อนอยู่ที่ 27.15% ใน Train Set และ 39.77% ใน Test Set

จากผลการทดลองสามารถสรุป Hyperparameter และโมเดลที่ดีที่สุดรวมถึงค่าความคลาดเคลื่อนของแต่ละการทดลองได้ดังตารางที่ 4.6 และ 4.7

ตารางที่ 4.6 Hyperparameter ที่ใช้ในแต่ละชุดการทดลอง

การทดลอง	โมเดล	Hyperparameter ที่ใช้
สารละลายโพแทสเซียมเปอร์แมงกาเนต	RF	n_estimators=50, min_samples_split=5 และ max_depth=10
สารละลายคาร์โบซิลแฟน	XGBoost	n_estimators = 400, max_depth = 6 และ learning_rate = 0.01
สารละลายเกลือ	RF	n_estimators = 300, min_samples_split = 2 และ max_depth = 50
สารละลายน้ำตาล	MLP	hidden_layer_sizes = (100, 100), alpha = 0.001 และ activation = tanh
สารละลายเกลือ-น้ำตาล	SVR	kernel = 'sigmoid' และ C = 1

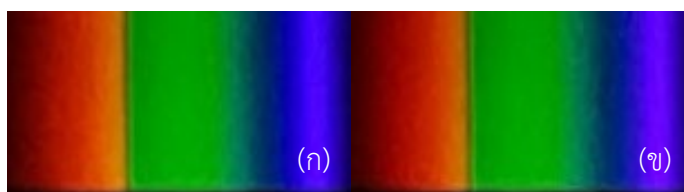
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 สรุปโมเดลที่ดีที่สุดและค่าความคลาดเคลื่อนของแต่ละการทดลอง

การทดลอง	โมเดล	MAPE (%)		MAE		R-Square		MSE	
		Train	Test	Train	Test	Train	Test	Train	Test
สารละลายโพแทสเซียมเปอร์แมงกาเนต	RF	1.52	6.03	0.12	0.5	1.00	0.98	0.10	0.73
สารละลายคาร์โบซิลแฟน	XGBoost	2.73	18.52	19.87	159.09	1.00	0.72	632.54	40028.94
สารละลายเกลือ	RF	14.88	32.34	1.34	3.14	0.90	0.58	2.44	12.81
สารละลายน้ำตาล	MLP	3.76	21.65	0.30	2.25	0.99	0.71	0.17	8.98
สารละลายเกลือ-น้ำตาล	SVR	27.15	39.77	1.14	1.58	0.22	-0.18	2.10	3.80

4.3 วิเคราะห์ผลการทดลอง

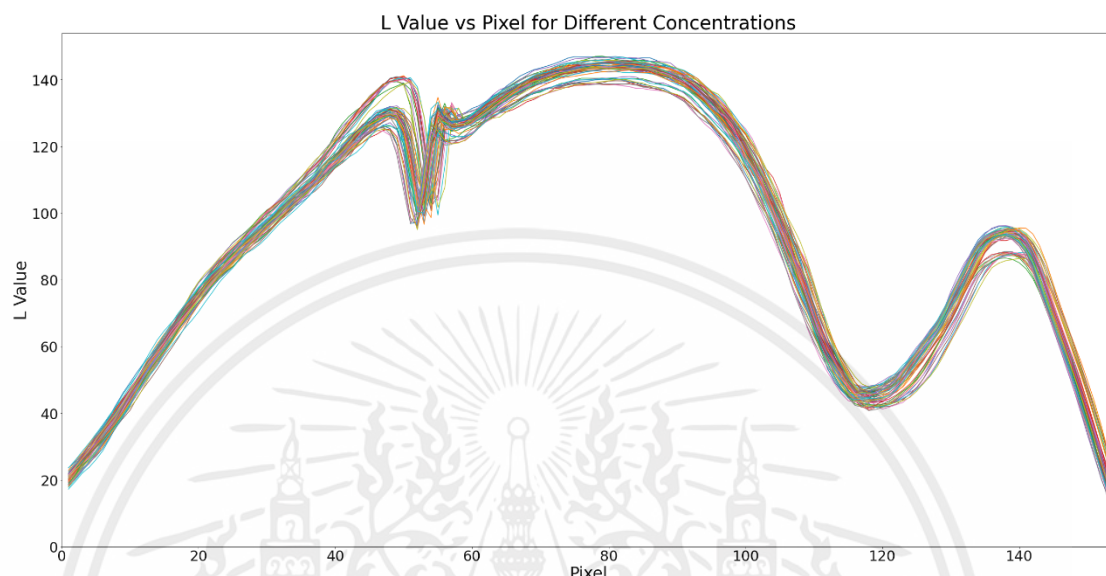
จากผลการทดลองพบว่า การประยุกต์ใช้การเรียนรู้ของเครื่องสามารถนำมาทำนายความเข้มข้นของสารด้วยเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพาได้ และให้ความคลาดเคลื่อนสำหรับชุดข้อมูลเกลือ, น้ำตาล และเกลือ-น้ำตาล ซึ่งเป็นสารที่ใส ไม่มีสี และวิเคราะห์ความเข้มข้นได้ยากหากใช้การดูดกลืนแสงในช่วงความยาวคลื่นของแสงที่มองเห็นได้ จากผลการประยุกต์ใช้การเรียนรู้ของเครื่องเข้ากับเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพา ได้ค่าความคลาดเคลื่อนเป็นร้อยละ 32.34, 21.65 และ 39.77 ตามลำดับ พบว่ามีความเป็นไปได้ในการทำนายหากมีการพัฒนาต่อ และชุดข้อมูลสารละลายคาร์โบซิลแฟนเป็น 18.52% เมื่อเทียบกับงานวิจัยก่อนหน้า 21.88 % พบว่าสามารถลดค่าความคลาดเคลื่อนได้เพียง 3.36 % เมื่อวินิจฉัยสาเหตุของปัญหาพบว่าเมื่อนำภาพของแถบสีอ้างอิงในชุดทดลองคาร์โบซิลแฟนมาเทียบกัน ซึ่งตามทฤษฎีควรมีความสว่างใกล้เคียงกันแต่พบว่าแถบสีอ้างอิงแต่ละความเข้มข้นมีสีต่างกัน ดังรูปที่ 4.30



รูปที่ 4.30 แถบสีอ้างอิงคาร์โบซิลแฟนความเข้มข้น 0 ppm (ก) และ 300 ppm (ข)

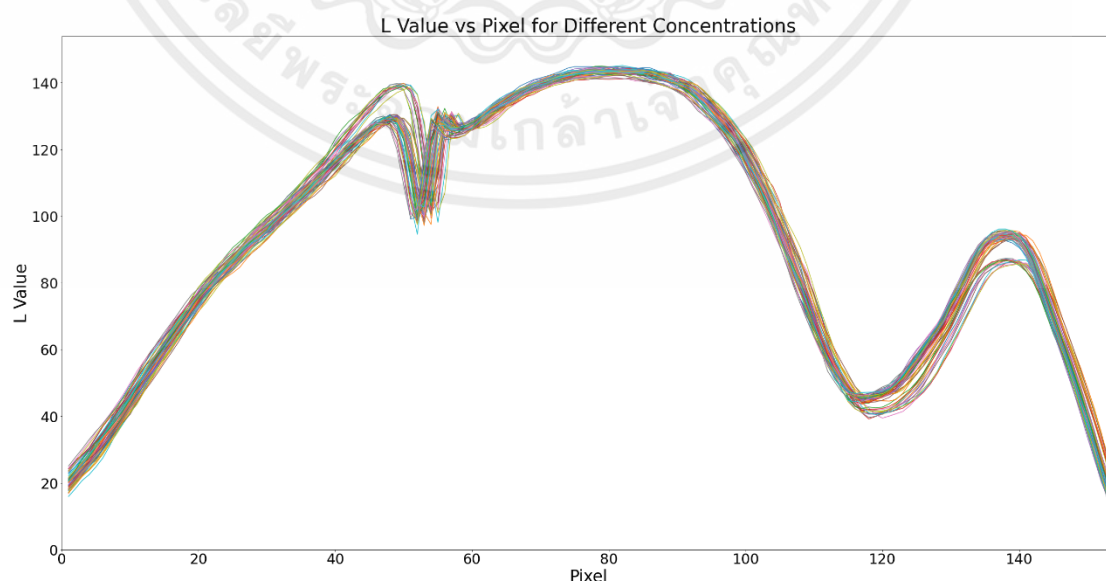
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแถบสีอ้างอิงที่ต่างความเข้มข้น พบว่าบริเวณแถบสีแดงของรูป (ก) มีความสว่างมากกว่ารูป (ข) แต่บริเวณแถบสีน้ำเงินของรูป (ก) กลับมีความสว่างที่น้อยกว่ารูป (ข) ซึ่งแสดงเป็นค่าความสว่างโดยเฉลี่ยจากแถบสีตั้งแต่ฝั่งซ้าย(สีแดง) จนถึงฝั่งขวา(สีม่วง) ของทุกรูปการทดลอง ดังรูปที่ 4.31



รูปที่ 4.31 ค่าความสว่างของแถบสีอ้างอิงจากการทดลองสารละลายคาร์โบซิลแฟน

ส่งผลให้ฟังก์ชัน”เฉลี่ยความสว่างของแถบสีแสงฝั่งที่ไม่มีสารละลาย” ภายใต้สมมุติฐานว่า “ความสว่างของแถบสีอ้างอิงต้องมีความสว่างของแต่ละสีสม่ำเสมอ ใกล้เคียงกัน และมีแนวโน้มของความสว่างเดียวในทุกชุดการทดลอง” ซึ่งมีหน้าที่ปรับความสว่างของแถบสีฝั่งที่ต้องการวิเคราะห์จากฝั่งแถบสีอ้างอิงทำงานได้ไม่เหมาะสมกับชุดทดลองนี้ ผู้วิจัยจึงทดลองใช้งานฟังก์ชันนี้ปรับความสว่างของฝั่งแถบสีอ้างอิงเพื่อวินิจฉัยที่มาของปัญหาแสดงดังรูปที่ 4.32



รูปที่ 4.32 ค่าความสว่างของแถบสีอ้างอิงจากการทดลองสารละลายคาร์โบซิลแฟนหลังการใช้ฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพหลังการใช้งานฟังก์ชันสังเกตว่าสามารถปรับให้ความสว่างใกล้เคียงกันได้ แต่เห็นได้ชัดว่ามี 2 แนวโน้มทั้งที่ตามสมมุติฐานต้องมีเพียงแนวโน้มเดียว ผู้วิจัยจึงไปวิเคราะห์ที่อุปกรณ์เพื่อหาสาเหตุของความสม่ำเสมอของแถบสีอ้างอิงพบว่า ขณะทดลองหลอดไฟ LED ของเครื่องสเปกโตรโฟโตมิเตอร์แบบพกพามีการเลื่อนตำแหน่ง ซึ่งมีลักษณะเป็นดวง เมื่อมีการเลื่อนตำแหน่งจึงส่งผลให้ความสว่างไม่สม่ำเสมอในแต่ละพื้นที่ของรูปภาพ ส่งผลให้แถบสีอ้างอิงมีแนวโน้มของสว่างแต่ละรูปที่ต่างกัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

ในส่วนนี้แบ่งออกสองส่วนได้แก่ 5.1 การสรุปผลที่ได้จากการทดลอง และ 5.2 ข้อเสนอแนะสรุปข้อควรกระทำจากความผิดพลาดที่เคยเกิดขึ้นรวมถึงสิ่งที่สามารถพัฒนาต่อไป

5.1 สรุปผลการทดลอง

การทดลองใช้ข้อมูลจากห้าชุดทดลอง ได้แก่ ชุดข้อมูลสารละลายโพแทสเซียมเปอร์แมงกาเนต (KMnO_4) ซึ่งเป็นสารที่มีสี จากงานวิจัย “การพัฒนาอุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล”, สารละลายคาร์โบซัลแฟน ($\text{C}_{20}\text{H}_{32}\text{N}_2\text{O}_3\text{S}$), สารละลายเกลือ, สารละลายน้ำตาล และสารเกลือ-น้ำตาล ซึ่งเป็นสารใสไม่มีสี จากงานวิจัย “การพัฒนาอุปกรณ์สเปกโตรโฟโตมิเตอร์ชนิดพกพาราคาประหยัดสำหรับการวิเคราะห์สารกำจัดศัตรูพืชประเภทคาร์บาเมต” พบว่าสามารถพัฒนาโปรแกรมสำหรับการจับภาพแถบสีอ้างอิง แถบสีที่ต้องการวิเคราะห์และจัดการรูปภาพให้เหมาะสมแก่การประยุกต์ใช้การเรียนรู้ของเครื่องได้

สำหรับสารมีสี สารละลายโพแทสเซียมเปอร์แมงกาเนต (KMnO_4) เมื่อใช้โมเดล Random Forest สามารถทำนายความเข้มข้นได้ที่ค่าความคลาดเคลื่อนเฉลี่ย (Mean Absolute Percentage Error) เพียง 6.03% ซึ่งดีกว่าผลงานวิจัยก่อนหน้าที่มีค่าความคลาดเคลื่อน 22.88%

สำหรับสารใสไม่มีสี สารละลายคาร์โบซัลแฟน ($\text{C}_{20}\text{H}_{32}\text{N}_2\text{O}_3\text{S}$) เมื่อใช้โมเดล Extreme Gradient Boosting สามารถทำนายความเข้มข้นของได้ที่ความคลาดเคลื่อนเฉลี่ย 18.52 % ซึ่งดีกว่าผลงานก่อนหน้าที่ 21.88% สารละลายเกลือ น้ำตาลและสารละลายเกลือ-น้ำตาลสามารถทำนายความเข้มข้นเมื่อใช้โมเดล Random Forest, Multilayer Perceptron และ Support Vector Regression ได้ค่าความคลาดเคลื่อนเฉลี่ยร้อยละ 32.34, 21.65 และ 39.77 ตามลำดับ

ผลการทดลองแสดงให้เห็นว่าการนำเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) มาใช้ ช่วยเพิ่มความแม่นยำในการทำนายความเข้มข้นของสารได้ทั้งสารที่มีสีและไม่มีสี เมื่อเปรียบเทียบกับงานวิจัยก่อนหน้า อย่างไรก็ตาม ระดับความคลาดเคลื่อนยังคงมีค่าสูงสำหรับสารที่ใสไม่มีสี ซึ่งเป็นข้อจำกัดที่ควรได้รับการปรับปรุงต่อไป

5.2 ข้อเสนอแนะ

1. ควรเขียนโปรแกรมให้เหมาะสมต่อการขยายและต่อยอดหากมีการเพิ่มจำนวนสารหรือทำสารละลายที่มีต้องการหาความเข้มข้นของสารมากกว่า 1 ชนิด
2. การจับมุมด้วยเทคนิคสัณฐานวิทยายังไม่สามารถจับมุมได้อย่างละเอียด จึงส่งผลให้การวิเคราะห์และเลือกแถบสีไม่ตรงกันในแต่ละรูป จึงเสนอแนะว่าหากใช้เทคโนโลยี Deep learning อาจจะสามารถแก้ไขปัญหาการจับมุมของแถบสีได้
3. เนื่องจากในงานวิจัยมีการใช้การทำซ้ำเพื่อสร้างรูปภาพหลังการวิเคราะห์ในแต่ละขั้นตอนและจัดเก็บในโฟลเดอร์ ส่งผลให้มีการใช้ทรัพยากรคอมพิวเตอร์มากและใช้เวลาในการประมวลผลนาน จึงเสนอแนะว่าให้ลองสืบค้นชุดคำสั่งอื่นๆที่ช่วยในการเก็บข้อมูลประเภทรูปภาพให้สามารถตรวจสอบได้โดยไม่ต้องสร้างใหม่และจัดเก็บภายในคอมพิวเตอร์
4. ปัจจุบันโปรแกรมที่ใช้ในการทำนายความเข้มข้นของสาร ใช้ได้เฉพาะกับสารที่ทราบชนิดแต่ไม่ทราบปริมาณเท่านั้น ดังนั้นจึงแนะนำว่าในอนาคตที่มีข้อมูลเพียงพอแล้วอาจมีการนำ Unsupervised learning มาช่วยในการจำแนกสารในกรณีที่ไม่ทราบชนิดของสารทดลอง
5. ปัจจุบันอุปกรณ์สเปกโตรโฟโตมิเตอร์แบบพกพามีตัวแปรที่ไม่สามารถควบคุมได้ในทุกชุดการทดลอง ไม่ว่าจะเป็นตำแหน่งของหลอดไฟ LED ระดับแบตเตอรี่ของหลอดไฟ LED ซึ่งส่งผลกับความสว่างโดยตรง ตำแหน่งการใส่โทรศัพท์มือถือและการใส่คิวเวท หากลดทอนตัวแปรที่ไม่สามารถควบคุมเหล่านี้ให้มีมาตรฐานมากขึ้นจะได้ผลลัพธ์ที่ดีขึ้น

เอกสารอ้างอิง

- [1] “สารบัญ สารก าจ คั ตรู พี ช”.
- [2] “65-2-DEVELOPMENT OF THE DEVICE FOR CARBAMATE PESTICIDE ANALYSIS USING THE LIGHT A”.
- [3] “สเปกโทรสโคปี (Spectroscopy) 1 1. บทน ำ 2. เทคนิค คั ตรำไวโอเลตและวิ ลี เบี ลสเปกโทรสโคปี (UV-VIS Spectroscopy) 3. ส วนประกอบของเครี' องอั ตรำไวโอเลตและวิ ลี เบี ลสเปกโทรโฟโตมิ เตอร์ (UV-VIS Spectrophotometer) 4. ก ำรประยุ กต์ ใช้ (Application) <http://www.blogspan.org/blogs/permalinks/6-2007/new-tool-for-spectroscopy.html>.” [Online]. Available: <http://www.blogspan.org/blogs/permalinks/6-2007/new-tool-for-spectroscopy.html>
- [4] “uv-vis spectroscopy.” Accessed: Nov. 08, 2023. [Online]. Available: https://il.mahidol.ac.th/e-media/color-light/page2_2.html
- [5] “Materials Innovation Center.” Accessed: Nov. 08, 2023. [Online]. Available: https://www.mic.eng.ku.ac.th/facilities-detail.php?id_sub=41&id=46
- [6] “ระบบสี RGB VS CMYKแตกต่างกันยังไง? - iop.” Accessed: Oct. 31, 2023. [Online]. Available: <https://www.iop.co.th/content/6222/%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%9A%E0%B8%AA%E0%B8%B5-rgb-vs-cmyk%E0%B9%81%E0%B8%95%E0%B8%81%E0%B8%95%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B9%84%E0%B8%87>
- [7] “ระบบ $L^*a^*b^*$ ซึ่งเป็นระบบการบรรยายสีแบบ 3 มิติ โดยที่แกน L^* จะบรรยายถึงความสว่าง (lightness) จากค่า $+L^*$ แสดงถึงสีขาว จนไปถึง $-L^*$ - AB All Techno : Inspired by LnwShop.com.” Accessed: Oct. 31, 2023. [Online]. Available: <https://www.aballtechno.com/article/18/%E0%B9%80%E0%B8%84%E0%B8%A3%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%87%E0%B8%A7%E0%B8%B1%E0%B8%94%E0%B8%AA%E0%B8%B5%E0%B9%81%E0%B8%9A%E0%B8%9>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A%E0%B8%9E%E0%B8%81%E0%B8%9E%E0%B8%B2-colorimeter-spectrophotometer

- [8] “มารู้จักประวัติของ ‘ภาษาไพธอน’ กันเถอะ! | TrueID Creator.” Accessed: Oct. 31, 2023. [Online]. Available: https://intrend.trueid.net/article/%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B8%A7%E0%B8%B1%E0%B8%95%E0%B8%B4%E0%B8%82%E0%B8%AD%E0%B8%87-%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2%E0%B9%84%E0%B8%9E%E0%B8%98%E0%B8%AD%E0%B8%99-%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B9%80%E0%B8%96%E0%B8%AD%E0%B8%B0-trueidintrend_77630
- [9] “Pandas Introduction What is Pandas | Python Tricks.” Accessed: Oct. 31, 2023. [Online]. Available: <https://python-tricks.com/pandas-introduction/>
- [10] “Introduction to NumPy — Definition and Brief History | by TechwithJules | Medium.” Accessed: Oct. 31, 2023. [Online]. Available: <https://medium.com/@techwithjules/introduction-to-numpy-definition-and-brief-history-db047e66a179>
- [11] “History of OpenCV from v1 to v4 - Mastering OpenCV 4 - Third Edition [Book].” Accessed: Oct. 31, 2023. [Online]. Available: <https://www.oreilly.com/library/view/mastering-opencv-4/9781789533576/2de0893d-e450-417d-b336-a4143799b43d.xhtml>
- [12] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, Accessed: Oct. 31, 2023. [Online]. Available: <http://jmlr.org/papers/v12/pedregosa11a.html>
- [13] “w2-morphological | group1.” Accessed: Nov. 09, 2023. [Online]. Available: <https://humancominteracg1.wixsite.com/group1/w2-morphological>

- [14] “OpenCV: Morphological Transformations.” Accessed: Nov. 10, 2023. [Online]. Available: https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html
- [15] R. Klette, *Concise Computer Vision: An Introduction into Theory and Algorithms*. Springer, 2014.
- [16] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Prentice Hall Professional Technical Reference, 2001.
- [17] N. Dwivedi, D. K. Singh, and D. S. Kushwaha, “An Approach for Unattended Object Detection through Contour Formation using Background Subtraction,” *Procedia Comput Sci*, vol. 171, pp. 1979–1988, 2020, doi: 10.1016/j.procs.2020.04.212.
- [18] “5 สิ่งที่คุณควรรู้เกี่ยวกับ Machine Learning.” Accessed: Oct. 31, 2023. [Online]. Available: <https://www.quickserv.co.th/knowledge-base/technology/5-%E0%B8%AA%E0%B8%B4%E0%B9%88%E0%B8%87%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B8%97%E0%B8%B8%E0%B8%81%E0%B8%84%E0%B8%99%E0%B8%84%E0%B8%A7%E0%B8%A3%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B9%80%E0%B8%81%E0%B8%B5%E0%B9%88%E0%B8%A2%E0%B8%A7%E0%B8%81%E0%B8%B1%E0%B8%9A-Machine-Learning/>
- [19] “Supervised Learning คืออะไร? ทำงานยังไง? | by Phuri Chalermkiatsakul | Medium.” Accessed: Oct. 31, 2023. [Online]. Available: <https://phuri.medium.com/supervised-learning-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B8%97%E0%B8%B3%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B9%84%E0%B8%87-1c0e411a40a2>
- [20] “1. Supervised learning — scikit-learn 1.3.2 documentation.” Accessed: Oct. 31, 2023. [Online]. Available: https://scikit-learn.org/stable/supervised_learning.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [21] “XGBoost - Wikipedia.” Accessed: Mar. 15, 2024. [Online]. Available: <https://en.wikipedia.org/wiki/XGBoost>
- [22] “3.1. Cross-validation: evaluating estimator performance — scikit-learn 1.4.1 documentation.” Accessed: Mar. 24, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html
- [23] G. O. Mayrink, D. M. de Queiroz, A. L. de F. Coelho, and D. S. M. Valente, “SOIL PHOSPHORUS TEST USING A LOW-COST SPECTROPHOTOMETER AND MACHINE LEARNING,” *Engenharia Agricola*, vol. 42, no. 6, 2022, doi: 10.1590/1809-4430-eng.agric.v42n6e20210228/2022.
- [24] A. Lesani *et al.*, “Quantification of human sperm concentration using machine learning-based spectrophotometry,” *Comput Biol Med*, vol. 127, p. 104061, Dec. 2020, doi: 10.1016/j.combiomed.2020.104061.



ภาคผนวก ก.

โปรแกรมสำหรับวิเคราะห์ข้อมูลจากงานวิจัย “การพัฒนา
อุปกรณ์ในการวิเคราะห์สารกำจัดศัตรูพืชกลุ่มคาร์บาเมตโดยใช้
หลักการการดูดกลืนแสงและการวิเคราะห์ภาพดิจิทัล”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรียกใช้ชุดคำสั่ง, กำหนดตัวแปร และเริ่ม loop ลบภาพ

```

import cv2

import numpy as np

import pandas as pd

import os

import time

from sklearn.model_selection import RandomizedSearchCV, KFold, train_test_split,
GridSearchCV, LeaveOneOut

from sklearn.ensemble import RandomForestRegressor

from sklearn.neural_network import MLPRegressor

from sklearn.linear_model import LinearRegression

from XGBoost import XGBRegressor

from sklearn.svm import SVR

from sklearn.feature_selection import SelectKBest, f_regression

from sklearn.metrics import mean_absolute_percentage_error,
mean_squared_error, r2_score

import matplotlib.pyplot as plt

import pickle

#Noted that all variable name are define by "ton"

start_time = time.time() #Track run time

# Define main_dir and name of each folder #Note: follow instruction 1 in msteam

main_dir = r"C:\Users\acer\Desktop\TEST_PROJECT4_new - lastst" #main folder

INPUT_DIR = os.path.join(main_dir, "test") #Input subfolder raw image

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OUTPUT_CROP_IMAGE = os.path.join(main_dir, "Air_Real_Crop") #Output crop area
of air image

OUTPUT_REAL_MARK = os.path.join(main_dir, "Air_Real_Mark") # Output full image
with mark Air and analysis image

OUTPUT_Crop_MARKED = os.path.join(main_dir, "Air_Crop_Mark")

OUTPUT_Crop_WARP = os.path.join(main_dir, "Air_Crop_Warp") # Output Air with
warp perspective

output_crop_Warp_add = os.path.join(main_dir, "Air_Crop_Warp_add") # Output
Warp air image with add condition

OUTPUT_Anal_real = os.path.join(main_dir, "Anal_real_mark") # Output full image
with mark analysis image

OUTPUT_ANAL_WARP = os.path.join(main_dir, "Anal_real_warp") # Output Analysis
with warp perspective

OUTPUT_ANAL_WARP_add = os.path.join(main_dir, "Anal_real_warp_add") #
Output Warp Analysis image with add conditio

MODEL = os.path.join(main_dir, "model")

#Define value for image pre-processing #Note: follow instruction 2 in msteam
height_solution_ratio = 1.2 #height of crop box for solution if not use put on 1
width_solution_ratio = 1.4 #width of crop box for solution if not use put on 1
distance_x = 475 #distance in x axis from air center to solution center neglect
elevation

more_red = -6#9.5 #adjust to give more red in solution image if not use put on 0
rotate_counter_clockwise = 1#1.2 #adjust number for image rotate cc if not use
put on 1

rotate_clockwise = 0#0.025 # adjust number for image rotate c if not use put on 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือนำไปใช้ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

angle_adjustment = 0#2 #adjust output solution image orientation if not use put
on 0

perfect_morph = 0#0.01 #for image that have rotation but after morph give
perfect no rotate rectangular if not use put on 0

new_angle = 0.00

#Delete all image in last running code

directories = [OUTPUT_CROP_IMAGE, OUTPUT_REAL_MARK,
OUTPUT_Crop_MARKED, OUTPUT_Crop_WARP, output_crop_Warp_add,
OUTPUT_Anal_real, OUTPUT_ANAL_WARP, OUTPUT_ANAL_WARP_add] #Define
subfolder to delete image inside

for directory in directories: #loop through each data in directories

    for filename in os.listdir(directory): #Loop each image

        file_path = os.path.join(directory, filename) # Define path for each image

        try: # Define condition for delete image

            if os.path.isfile(file_path):

                os.remove(file_path)

        except OSError as e:

            print(f"Error: {e.filename} - {e.strerror}")

```

ฟังก์ชัน

1) **Function name:** Crop_by_avg

Objective: Crop Reference spectral band image.

Input: Raw Image

Output: Reference Spectral band crop area

```
def Crop_by_avg(Raw_Image): #input: Full raw image from experiment
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lab_image = cv2.cvtColor(Raw_Image, cv2.COLOR_BGR2LAB) # Read image with
LAB system

L_channel = lab_image[:, :, 0] # Pick only axis of L value

avg_L_values_horizontal = L_channel.mean(axis=0).flatten() # Average L value in
axis = 0 (row)

avg_L_values_vertical = L_channel.mean(axis=1).flatten() # Average L value in
axis = 1 (Columns)

#horizontal pixels process

pixel_columns_horizontal_df = pd.DataFrame() # Crate Blank Dataframe to store
average L value

pixel_columns_horizontal_df ['avg_L_values_horizontal'] =
avg_L_values_horizontal # Store value in Blank DataFrame

pixel_columns_horizontal_df =
pixel_columns_horizontal_df.apply(pd.to_numeric).T # Transpose for loop find
pixel number

#vertical pixels process

pixel_columns_vertical_df = pd.DataFrame() # Crate Blank Dataframe to store
average L value

pixel_columns_vertical_df ['avg_L_values_vertical'] = avg_L_values_vertical #
Store value in Blank DataFrame

pixel_columns_vertical_df = pixel_columns_vertical_df.apply(pd.to_numeric).T #
Transpose for loop find pixel number

column_max_df_vertical = pd.DataFrame() # Define Blank DataFrame for
contain max vertical for each image

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for index, row in pixel_columns_horizontal_df.iterrows(): # loop through the
DataFraame

    max_column_number = row.idxmax() # find pixel with heighest average L
value

    for row_index, Column in pixel_columns_vertical_df.iterrows(): # loop through
the DataFraame

        max_column_number_ver = Column.idxmax() # find pixel with heighest
average L value

        max_column_number_ver = int(max_column_number_ver) #int

        max_column_number_new = max_column_number_ver - 100 # DeFine
heightest for only left side of image

        column_max_df_vertical.at [row_index, 'y'] = max_column_number_new #
get Heightest DataFrame pixel for each image

    for row_index, column in pixel_columns_vertical_df.iterrows(): # Same
Horizontal

        max_column_number_new = int(column_max_df_vertical.at [row_index, 'y'])
# Same Horizontal

        new_data_slice = column [:max_column_number_new + 1] # Same
Horizontal

        new_max_column_number_vertical = new_data_slice.idxmax() # Same
Horizontal

        x,y = max_column_number, new_max_column_number_vertical # get position
for center point area of crop image

        width = 600 # Define area for crop

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

height = 336 # Define area for crop

half_width= width // 2 # Define area for crop

half_height = height // 2 # Define area for crop

top_left_x = x - half_width # Define area for crop

top_left_y = y - half_height # Define area for crop

bottom_right_x = x + half_width # Define area for crop

bottom_right_y = y + half_height # Define area for crop

crop_box = (top_left_x, top_left_y, bottom_right_x, bottom_right_y) # Define
area for crop

cropped_img = Raw_Image [crop_box [1]:crop_box [3], crop_box [0]:crop_box
[2]] # Define area for crop

return cropped_img, crop_box # Two output is Crop image and crop position
respectively

```

2) **Function name:** preprocess_image

Objective: Crop Reference spectral band image.

Input: Full raw image from experiment

Output: contour or desired crop image

```

def preprocess_image(img): # #input: Full raw image from experiment
    cropimage,Crop_box = Crop_by_avg(img) #Sent full raw image to Crop_by_avg
function for cropimage

    blurred = cv2.GaussianBlur(cropimage, (5, 5), 0) #Blur

    gray = cv2.cvtColor(blurred, cv2.COLOR_BGR2GRAY) # turn to gray scale system
    _, thresh = cv2.threshold(gray,12, 255, cv2.THRESH_BINARY) # Defind Threshold
for seperate foreground and background

    kernel = np.ones((5,5), np.uint8) # Define kernal for mophology

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OPEN = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel, iterations=3) #use
opening method
closing = cv2.morphologyEx(OPEN, cv2.MORPH_CLOSE, kernel, iterations=3)
#use Colseing method
return closing # Output: contour or desired crop image

```

3) **Function name:** find_corners

Objective: Find corners of the Desired square image

Input: Full raw image from experiment

Output: array of position in each corners of square contour

```

def find_corners(img): # input: Full raw image from experiment
    processed_img = preprocess_image(img) # Sent image to processed_img for
contour of dessired image
    contours, _ = cv2.findContours(processed_img, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE) #Use find contour to find corners
    contours = sorted(contours, key=cv2.contourArea, reverse=True)
    contour = contours [0]
    rect = cv2.minAreaRect(contour)
    box = cv2.boxPoints(rect)
    box = np.intp(box)
    return box # Output array of position in each corners of square contour

```

4) **Function name:** Linedistance

Objective: Find distance between each corners

Input: array of position in each corners of square contour

Output: Array of distance

```

def Linedistance(corners): # input: array of possition in each corners of square
contour

```

```
# Find distance 1 #Noted: Value change everytime the image have angle. Check
variable in corners to know each distance.

distance1 = np.linalg.norm(corners [1] - corners [0])
distance2 = np.linalg.norm(corners [2] - corners [1])
distance3 = np.linalg.norm(corners [3] - corners [2])
distance4 = np.linalg.norm(corners [3] - corners [0])
distances = np.array( [distance1, distance2, distance3, distance4]) # Combine all
distance to array

return distances # Output: Array of distance
```

5) **Function name:** corner_reorder

Objective: Reorder the corners of square contour

Input: array of corners

Output: array of reordered corners

```
def corner_reorder(corners): # input: array of corners
    corners = find_corners(img) # Find corners of the square contour
    distances = Linedistance(corners) # find distance between corners
    if distances [0] < distances [1]: # Define condition based on the square contour
        recorners = np.array( [corners [1], corners [2], corners [3], corners [0]])
    else:
        recorners = np.array( [corners [2], corners [3], corners [0], corners [1]])
    return recorners # OutputL array of reordered corners
```

6) **Function name:** FiveParam

Objective: turn reordered_corners(8 parameters) in to 5 parameter

Input: array of reordered_corners

Output: get all 5 parameter

```
def FiveParam(reordered_corners): # input: array of reordered_corners
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

distanceheight = np.linalg.norm(reordered_corners [0] - reordered_corners [1]) #
Find height of square image
distancewidth = np.linalg.norm(reordered_corners [0] - reordered_corners [3]) #
Find Width of square image
center = (reordered_corners [3] + ([ [distancewidth/2, distanceheight/2]))) # Find
Center point of square
ref_distance = reordered_corners [1] [0] - reordered_corners [2] [0] # Find
Horizontal reference line for angle finding
angledistance = np.linalg.norm(reordered_corners [1] - reordered_corners [2]) #
Find distance of angle finding
if reordered_corners [0] [1] > reordered_corners [3] [1]: # Define when angle are
negative
    anglevalue = -np.arccos(ref_distance/angledistance) #*180/np.pi
else :
    anglevalue = np.arccos(ref_distance/angledistance) # Define angle when
positive
return center,distancewidth,distanceheight,anglevalue # get all 5 parameter

```

7) **Function name:** CenterToFiveParamFormRef

Objective: Find 5 parameter in analysis image with center point of analysis image

Input: center point of analysis image, and all 3 parameter from Air image with adjusted values

Output: corners of analysis image

```

def CenterToFiveParamFormRef(new_center_point, old_reordered_corners,
old_angle, height_solution_ratio, width_solution_ratio, more_red): #Input: center
point of analysis image, and all 3 parameter from Air image with adjusted values
    distanceheight = np.linalg.norm(old_reordered_corners [0] -
old_reordered_corners [1]) # Define height
    distanceheight = distanceheight*height_solution_ratio + more_red # Adjusted
for ratio of height

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

distancewidth = np.linalg.norm(old_reordered_corners [0] -
old_reordered_corners [3]) # Define width
distancewidth = distancewidth*width_solution_ratio # Adjusted number for ratio
of width
#old_angle = old_angle #if need can give adjusted number for change angle
# find new corner from old angle
x0_new = new_center_point [0] [0] + distancewidth/2 * np.cos(old_angle) -
distanceheight/2 * np.sin(old_angle)
y0_new = new_center_point [0] [1] - distancewidth/2 * np.sin(old_angle) -
distanceheight/2 * np.cos(old_angle)
x1_new = new_center_point [0] [0] + distancewidth/2 * np.cos(old_angle) +
distanceheight/2 * np.sin(old_angle)
y1_new = new_center_point [0] [1] - distancewidth/2 * np.sin(old_angle) +
distanceheight/2 * np.cos(old_angle)
x2_new = new_center_point [0] [0] - distancewidth/2 * np.cos(old_angle) +
distanceheight/2 * np.sin(old_angle)
y2_new = new_center_point [0] [1] + distancewidth/2 * np.sin(old_angle) +
distanceheight/2 * np.cos(old_angle)
x3_new = new_center_point [0] [0] - distancewidth/2 * np.cos(old_angle) -
distanceheight/2 * np.sin(old_angle)
y3_new = new_center_point [0] [1] + distancewidth/2 * np.sin(old_angle) -
distanceheight/2 * np.cos(old_angle)
corners = np.array( [ [x0_new,y0_new], [x1_new,y1_new], [x2_new,y2_new],
[x3_new,y3_new]])
return corners # Output: corners of analysis image

```

8) **Function name:** warp_perspective_new

Objective: Warp perspective of desired image

Input: Full raw image , reordered cornerd of desired image

Output: Warp of desired image

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def warp_perspective_new(img,recorners ,output_size=(160, 104)): # Input: Full raw
image , reordered cornerd of desired image
    dst = np.array( [
        [output_size [0] - 1, 0],
        [0, 0],
        [0, output_size [1] - 1],
        [output_size [0] - 1, output_size [1] - 1]
    ], dtype='float32') # Define warp perspective parameter
    M = cv2.getPerspectiveTransform(np.array(recorners, dtype='float32'), dst)#
Define warp perspective parameter
    warped = cv2.warpPerspective(img, M, output_size) # Warped
    return warped # Warp of desired image

```

9) **Function name:** lightness_adjust

Objective: Get base value for further adjust

Input: Depend on which part or image use for base-adjustment, in this case use top 500 row from full image

Output: Average L value from top 500 row for full image

```

def lightness_adjust(image, image_name): # Input: Depend on which part or image
use for base-adjustment, in this case use top 500 row from full image
    lab_image = cv2.cvtColor(image, cv2.COLOR_BGR2LAB) # Read image in LAB
color space
    L_channel = lab_image[:, :, 0] # Choose only L value
    top_500_rows = L_channel[:500, :] # Choose L value top 500 row
    avg_L_values = top_500_rows.mean(axis=0).flatten() # Average to get result in
horizontral
    pixel_columns = [f"pixel {i+1}" for i in range(len(avg_L_values))] # Create column
name
    data = pd.DataFrame({"Image_Name": [image_name], **{pixel_columns [i]:
[avg_L_values [i]] for i in range(len(avg_L_values))}}) # Create dataframe which each
row represent each image which their name and each column with l value

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
return data # Average L value from top 500 row for full image
```

10) **Function name:** Average_lightness

Objective: Get average L value in horizontal

Input: Image of solution spectral band after warperspective

Output: Average L in horizontal value from solution spectral band after warperspective

```
def Average_lightness(image, image_name): # Input: Image of solution spectral
band after warperspective
    lab_image = cv2.cvtColor(image, cv2.COLOR_BGR2LAB) # Read image in LAB
color space
    L_channel = lab_image[:, :, 0] # Choose only L value
    avg_L_values = L_channel.mean(axis = 0).flatten() # Average to get result in
horizontal
    pixel_columns = [f"pixel {i+1}" for i in range(len(avg_L_values))] # Create column
name
    data = pd.DataFrame({"Image_Name": [image_name], **{pixel_columns [i]:
[avg_L_values [i]] for i in range(len(avg_L_values))}}) # Create dataframe which each
row represent each image which their name and each column with l value
    return data # Average L in horizontal value from solution spectral band after
warperspective
```

11) **Function name:** Read_Image

Objective: Get average hue value

Input: image

Output: average hue value

```
def Read_Image(img_array): # Input: image
    image_rgb = cv2.cvtColor(img_array, cv2.COLOR_BGR2RGB) # Read image in RGB
height, width, _ = image_rgb.shape # Get the size of the image
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

num_rows = 50 # Define the number of rows to read (in this case, 50)
left_column_hues = [] # Read the left column of pixels and convert to HSV
for row in range(min(num_rows, height)):
    rgb_pixel = image_rgb [row, 0, :] # Get RGB values
    hsv_pixel = cv2.cvtColor(np.uint8( [ rgb_pixel])), cv2.COLOR_RGB2HSV) [0, 0]
# Convert RGB to HSV
    left_column_hues.append(hsv_pixel [0]) # Append the hue value to the list
    average_hue = sum(left_column_hues) / len(left_column_hues) # Calculate the
average hue value
    print("Average Hue Value:", average_hue) # Display the average hue
    return average_hue

```

12) Function name: local_minimum_find

Objective: To find minimum value for shift process

Input: Image, image name, Blank dataframe to stack minimum value and column header which has minimum value of all image

Output: is minimum L value for each image in that interval, column which have the minimum L value and median minimum from every image

```

def local_minimum_find(img, img_name, Local_minimum_each_image_Df,
min_column_number_df): # Input: Image, image name, Blank dataframe to stack
minimum value and column header which has minimum value of all image
    lab_image = cv2.cvtColor(img, cv2.COLOR_BGR2LAB) # Read image in LAB color
space
    L_channel = lab_image[:, :, 0] # Choose only L value
    median_L_values_horizontal = np.median(L_channel, axis=0).flatten() # Get
median value in horizontal
    pixel_columns_horizontal_df = pd.DataFrame({'median_L_values_horizontal':
median_L_values_horizontal}) # Collect median L value with column name
    pixel_columns_horizontal_df_for_find_local_minimum =
pixel_columns_horizontal_df.iloc [85:100] # Choose only column 85 to 100

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

min_column_number = pixel_columns_horizontal_df_for_find_local_minimum
['median_L_values_horizontal'].idxmin() # Get minimum value with in those
threshold

min_column_numbers_df = pd.DataFrame({'min_column_number_+1_if_Pixel':
[min_column_number]}) # Create dataframe to store minimum value

min_column_number_df = pd.concat([min_column_number_df,
min_column_numbers_df]) # Stack data every time its loop

min_column_number_df = min_column_number_df.reset_index() # Reset index
min_column_number_df = min_column_number_df.drop(columns =
"index").astype(int) #3 Change type of value to int

Local_minimum_each_image_Df = pd.concat([Local_minimum_each_image_Df,
pd.DataFrame({'filename': [img_name], 'min_column_number_+1_if_Pixel':
[min_column_number]})]) # Stack data with their image name and minimum value

Local_minimum_each_image_Df = Local_minimum_each_image_Df.reset_index()
# Reset index

Local_minimum_each_image_Df =
Local_minimum_each_image_Df.drop(columns = "index") # Drop column name
index

median_minimum = np.median(min_column_number_df, axis=0).flatten() # Get
median value from each minimum value

median_minimum_scalar = median_minimum.item() # Change type to int
median_minimum = int(median_minimum_scalar)

return Local_minimum_each_image_Df,
min_column_number_df, median_minimum # Output is mini L value for each
image in that interval, column which have the minimum L value and median
menimum from every image

```

13) **Function name:** Shift_image_median_minimum_add_condition

Objective: Shift image base on minimum value

Input: warperpective image and median minimum value

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output: Warp perspective image with warp add condition and x_shift from each image

```

average_hue = Read_Image(warp_img) # use Read_Image function to get
average hue value

initial_crop_box = (40, 20, 140, 330) # Indentifine crop box which is (X position
of topleft, y position of top left, x position of bottom right, y position of bottom
right)

lab_image = cv2.cvtColor(warp_img, cv2.COLOR_BGR2LAB) # Read image in LAB
color space

L_channel = lab_image[:, :, 0] # Choose only L value

median_L_values_horizontal = np.median(L_channel, axis=0).flatten() # Get
median value in horzontral

pixel_columns_horizontal_df = pd.DataFrame({'median_L_values_horizontal':
median_L_values_horizontal}) # Collect median L value with column name

pixel_columns_horizontal_df_for_find_local_minimum =
pixel_columns_horizontal_df.iloc [85:100] # CHoose only column 85 to 100

min_column_number = pixel_columns_horizontal_df_for_find_local_minimum
['median_L_values_horizontal'].idxmin() # Get minimum value with in those
threshold

local_diff_med = int(median_minimum) - min_column_number # Get difference
between median minimum from all image and from each image

x_shift = local_diff_med # Store difference value in name x_shift

print(x_shift) # Print

if average_hue < 205: # Create condition that if average hue lower than 205

warp_img = cv2.rotate(warp_img, cv2.ROTATE_180) # Rotate warped image

if x_shift > 0:

    shifted_crop_box = (initial_crop_box [0] + x_shift, initial_crop_box [1],
initial_crop_box [2]+ x_shift , initial_crop_box [3])

    warp_img_crop_add = warp_img [int(shifted_crop_box
[1]):int(shifted_crop_box [3]), int(shifted_crop_box [0]):int(shifted_crop_box [2])]

    print(shifted_crop_box)

else: # Operate image if x_shift is negative

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        shifted_crop_box = (initial_crop_box [0] + x_shift, initial_crop_box [1],
initial_crop_box [2] + x_shift , initial_crop_box [3])
        warp_img_crop_add = warp_img [int(shifted_crop_box
[1]):int(shifted_crop_box [3]), int(shifted_crop_box [0]):int(shifted_crop_box [2])]
        print(shifted_crop_box)
        return warp_img_crop_add,x_shift # Output: Warp perspective image with warp
add condition and x_shift from each image

```

14) **Function name:** condition

Objective: Warp perspective with condition but not shift

Input: warp desired image

Output: Warp add image

```

def condition(img_array): #input: warp desired image
    average_hue = Read_Image(img_array) # sent image to read with Read_image
    crop_box = (6,20, 200, 330) # Desired Crop_box with x of top left, y of bottom
left, x of top right and y of bottom right
    if average_hue < 205: #Ddfine condition before Operate picture
        img_array = cv2.rotate(img_array, cv2.ROTATE_180) # Rotate
        img_array_Crop = img_array [crop_box [1]:crop_box [3], crop_box
[0]:crop_box [2]] # Crop image if not appreciate shape
    else:
        # Save the image to the output folder without rotation
        img_array_Crop = img_array [crop_box [1]:crop_box [3], crop_box
[0]:crop_box [2]] # Crop image if not appreciate shape
    return img_array_Crop # Output: Warp add image

```

15) **Function name:** adjust_pixel_values

Objective: make desired pixel has all same L value

Input: Data_for_train

Output: adjusted dataframe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

def adjust_pixel_values(df):
    columns_to_drop = ['Concentration_1_ppm', 'Concentration_2_ppm', 'Solute_1',
'Solute_2', 'Solvent', 'Light_Source', 'Picture_Number'] # Chosse column to drop
    dropped_columns = df [columns_to_drop] # Store choosed column
    df = df.drop(columns=columns_to_drop) # Drop choosed column
    pixel_adjust = "pixel 25" # specific pixel to analyze
    pixel_1_avg = df [pixel_adjust].mean()# Calculate the average of the "pixel 1"
column
    adjusted_df = pd.DataFrame() # Create a new DataFrame to store the
adjusted values
    for _, row in df.iterrows(): # Iterate over each row in the original DataFrame
        adjustment = row [pixel_adjust] - pixel_1_avg # Calculate the
adjustment value for the current row
        adjusted_row = row.to_frame().transpose()# Create a new row with adjusted
values
        adjusted_row [pixel_adjust] -= adjustment
        for col in adjusted_row.columns: # Apply the same adjustment to all other
columns with loop
            if col != pixel_adjust:
                adjusted_row [col] -= adjustment
        adjusted_df = pd.concat( [adjusted_df, adjusted_row], ignore_index=True) #
Append the adjusted row to the new DataFrame
        dropped_columns.reset_index(drop=True, inplace=True) # Reset indices of both
DataFrames before concatenating
        adjusted_df.reset_index(drop=True, inplace=True) # Reset index
        adjusted_df = pd.concat( [dropped_columns, adjusted_df], axis=1) #
Concatenate the DataFrames along the columns axis
    return adjusted_df # Output: adjusted dataframe

```

16) Function name: create_substance_dfs

Objective: Create dictionary that separate with each experiment base on solute name

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input: dataframe after substact blank

Output: dictionary after cleaning

```
def create_substance_dfs(data): # Input: dataframe after substact blank
    substance_dfs = {} # create blank dictionary
    unique_substances = data[['Solute_1', 'Solute_2']].drop_duplicates().values #
drop all duplicate data leave only unique one
    for substance in unique_substances: # loop through unique value
        Solute_1, Solute_2 = substance # get data form substance in to solute_1
and solute_2
        if pd.isna(Solute_2): # create condition that if solute_2 is Null it will store as
'None'
            Solute_2 = 'None'
        key = f"{Solute_1}_{Solute_2.replace(' ', '')}" # Create key in name of each
solute
        substance_dfs[key] = data[(data['Solute_1'] == Solute_1) & (data['Solute_2']
== Solute_2)] # store in dictionary
    return substance_dfs # Output: dictionary after cleaning
```

17) **Function name:** pure_mixture_seperation

Objective: Separate the data into pure and mixture components, and perform machine learning on each component.

Input: dictionary from create_substance_dfs

Output: Return the pure component models, combined models, combined dictionaries, and combined metrics

```
def pure_mixture_seperation(all_dataframe): # Input: dictionary
from create_substance_dfs
    # Convert dictionary keys to a list
    dict_keys = list(all_dataframe.keys()) # Convert the keys of the input dictionary
'all_dataframe' to a list
    # Create empty dictionaries to store the results
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dict_combined = {} # Create an empty dictionary to store the combined
dictionary results

dict_mix = {} # Create an empty dictionary to store the mixture dictionary
results

metric_combined = {} # Create an empty dictionary to store the combined
metric results

metric_mix = {} # Create an empty dictionary to store the mixture metric results

model_pure = {} # Create an empty dictionary to store the pure component
models

model_mix = {} # Create an empty dictionary to store the mixture component
models

model_combine = {} # Create an empty dictionary to store the combined
models

for key in dict_keys: # Iterate over the keys in the list 'dict_keys'
    # Split the key by '_'
    key_parts = key.split('_') # Split the current key by the underscore character
    # Check if the second part (index 1) is 'None'
    if key_parts [1] == 'None': # Check if the second part of the key (index 1) is
'None'
        # This is a pure component
        DATA = all_dataframe [key] # Retrieve the data for the current key from
'all_dataframe'
        columns_to_drop = ['Concentration_1_ppm', 'Concentration_2_ppm',
'Solute_1', 'Solute_2', 'Solvent', 'Light_Source', 'Picture_Number'] # Define a list of
columns to be dropped from the data
        X = DATA.drop(columns=columns_to_drop).values # Create the feature
matrix 'X' by dropping the specified columns
        y = np.array(DATA ['Concentration_1_ppm'].values).astype(float) # Create
the target vector 'y' from the 'Concentration_1_ppm' column
        best_model, dict_df, metrics_df = machine_learning(X, y, key) # Call the
'machine_learning' function and store the results
        # Store the results in the respective dictionaries

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dict_combined[key] = dict_df # Store the dictionary result for the pure
component in 'dict_combined'

metric_combined[key] = metrics_df # Store the metric result for the pure
component in 'metric_combined'

model_pure[key] = best_model # Store the best model for the pure
component in 'model_pure'

else: # If the condition in the previous 'if' statement is false, execute this
block

# This is a mixture component
DATA = all_dataframe[key] # Retrieve the data for the current key from
'all_dataframe'
columns_to_drop = ['Concentration_1_ppm', 'Concentration_2_ppm',
'Solute_1', 'Solute_2', 'Solvent', 'Light_Source', 'Picture_Number'] # Define a list of
columns to be dropped from the data
X_1 = DATA.drop(columns=columns_to_drop).values # Create the feature
matrix 'X_1' by dropping the specified columns
y_1 = np.array(DATA ['Concentration_1_ppm'].values).astype(float) # Create
the target vector 'y_1' from the 'Concentration_1_ppm' column
best_model_1, dict_df_1, metrics_df_1 = machine_learning(X_1, y_1, key)
# Call the 'machine_learning' function and store the results for the first component
X_2 = DATA.drop(columns=columns_to_drop).values # Create the feature
matrix 'X_2' by dropping the specified columns
y_2 = np.array(DATA ['Concentration_2_ppm'].values).astype(float) # Create
the target vector 'y_2' from the 'Concentration_2_ppm' column
best_model_2, dict_df_2, metrics_df_2 = machine_learning(X_2, y_2, key)
# Call the 'machine_learning' function and store the results for the second
component

# Store the results in the respective dictionaries
dict_mix = {'{key_parts [0]}_{key_parts [1]}': {key_parts [0]: dict_df_1,
key_parts [1]: dict_df_2}} # Store the dictionary results for the mixture components
in 'dict_mix'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        metric_mix = {f"{key_parts [0]}_{key_parts [1]}": {key_parts [0]: metrics_df_1,
key_parts [1]: metrics_df_2}} # Store the metric results for the mixture components
in 'metric_mix'

        model_mix = {f"{key_parts [0]}_{key_parts [1]}": {key_parts [0]:
best_model_1, key_parts [1]: best_model_2}} # Store the best models for the
mixture components in 'model_mix'

        model_combine [key] = model_mix # Store the combined model for the
current key in 'model_combine'

        # Combine the dictionaries

        metric_combined = {**metric_combined, **metric_mix} # Combine the
'metric_combined' and 'metric_mix' dictionaries

        dict_combined = {**dict_combined, **dict_mix} # Combine the
'dict_combined' and 'dict_mix' dictionaries

        return model_pure, model_combine, dict_combined, metric_combined #
Return the pure component models, combined models, combined dictionaries,
and combined metrics

```

18) **Function name:** machine_learning

Objective: Predict and get evaluation metric

Input: X,y and key

Output: Return the best model, prediction DataFrames, and metrics
DataFrame

```

def machine_learning(X, y, key):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=11) # Split the data into train and test sets

    prediction_data_train_df = pd.DataFrame(y_train, columns= ['Actual']) # Create a
DataFrame for actual train values

    prediction_data_test_df = pd.DataFrame(y_test, columns= ['Actual']) # Create a
DataFrame for actual test values

    models = {
        'SVR': SVR(),

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'MLR': LinearRegression(),
'RF': RandomForestRegressor(),
'MLP': MLPRegressor(max_iter=100000),
'XGB': XGBRegressor()
} # Define the models to be trained
param_grids = {
'SVR': {
'C': [0.1, 1, 10, 100, 1000],
'kernel': ['linear', 'rbf', 'sigmoid']
},
'MLR': {
'fit_intercept': [True, False],
},
'RF': {
'n_estimators': [50, 100, 200, 300, 400, 500],
'max_depth': [None, 10, 20, 30, 40, 50],
'min_samples_split': [2, 5, 10, 15, 20]
},
'MLP': {
'hidden_layer_sizes': [(50, 50), (75, 75), (100,100), (25,25)],
'activation': ['relu', 'tanh'],
'alpha': [0.0001, 0.001, 0.01]
},
'XGB': {
'n_estimators': [100, 200, 300, 400, 500],
'max_depth': [3, 4, 5, 6, 7, 8],
'learning_rate': [0.01, 0.05, 0.1, 0.5, 1]
}
} # Define the parameter grids for each model
selector = SelectKBest(score_func=f_regression, k=4 ) # Select the best 4
features using the f_regression score function
kf = KFold(n_splits=10, shuffle=True, random_state=42) # Define a 10-fold
cross-validation object

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

loo = LeaveOneOut() # Define a leave-one-out cross-validation object
results = {} # Initialize an empty dictionary to store results
best_scores = {} # Dictionary to keep track of the best score for each model
best_y_trains = {} # Dictionary to keep track of the y_train for the best fold for
each model

best_params = {} # Dictionary to store the best parameters for each model
# Initialize lists to store evaluation metrics for each model
mape_train_list = []
mape_test_list = []
mae_train_list = []
mae_test_list = []
r2_train_list = []
r2_test_list = []
mse_train_list = []
mse_test_list = []
model_names = []

for model_name, model in models.items(): # Iterate over the models
    print(f"Training {model_name} - {key}...") # Print the model name and key
    best_score = float('inf') # Initialize the best score to infinity
    X_train_selected = selector.fit_transform(X_train, y_train) # Select the best
features from the training data

    feature_scores = selector.scores_ # Get the feature scores
    feature_scores_tuples = [(i, feature_scores [i]) for i in
selector.get_support(indices=True)] # Get the feature indices and scores
    sorted_features = sorted(feature_scores_tuples, key=lambda x: x [1],
reverse=True) # Sort the features by score in descending order

    #random_search = GridSearchCV(estimator=model, param_grid=param_grids
[model_name],
    #scoring='r2', cv=kf) # Define a grid search object with leave-one-out cross-
validation

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

random_search =
RandomizedSearchCV(estimator=model,param_distributions=param_grids
[model_name],n_iter=5,
scoring='r2',cv=kf,random_state=42)
random_search.fit(X_train_selected, y_train) # Fit the grid search object to
the training data
best_model = random_search.best_estimator_ # Get the best model from
the grid search
best_params [model_name] = random_search.best_params_ # Store the best
parameters for the model

with open(f"{model_name}_model.pkl", "wb") as f: # Open a file to store the
best model
pickle.dump(best_model, f) # Save the best model to the file

X_train_selected = selector.transform(X_train) # Transform the training data
using the selected features
predictions_train = best_model.predict(X_train_selected) # Make predictions
on the training data
predictions_train_df = pd.DataFrame(predictions_train, columns=
[f"{model_name}_Train"]) # Create a DataFrame for train predictions
prediction_data_train_df = pd.concat( [prediction_data_train_df,
predictions_train_df], axis=1) # Concatenate train predictions and actual values

r2_train = r2_score(y_train, predictions_train) # Calculate R2 score for train
predictions
mse_train = mean_squared_error(y_train, predictions_train) # Calculate MSE
for train predictions
mape_train = mean_absolute_percentage_error(y_train,
predictions_train)*100 # Calculate MAPE for train predictions
mae_train = np.mean(np.abs(y_train - predictions_train)) # Calculate MAE for
train predictions

```



```

X_test_selected = selector.transform(X_test) # Transform the test data using
the selected features

predictions_test = best_model.predict(X_test_selected) # Make predictions
on the test data

predictions_test_df = pd.DataFrame(predictions_test, columns=
[f'{model_name}_Test']) # Create a DataFrame for test predictions

prediction_data_test_df = pd.concat( [prediction_data_test_df,
predictions_test_df], axis=1) # Concatenate test predictions and actual values

r2_test = r2_score(y_test, predictions_test) # Calculate R^2 score for test
predictions

mse_test = mean_squared_error(y_test, predictions_test) # Calculate MSE for
test predictions

mape_test = mean_absolute_percentage_error(y_test, predictions_test)*100
# Calculate MAPE for test predictions

mae_test = np.mean(np.abs(y_test - predictions_test)) # Calculate MAE for
test predictions

# Append evaluation metrics to respective lists
mape_train_list.append(mape_train)
mape_test_list.append(mape_test)
mae_train_list.append(mae_train)
mae_test_list.append(mae_test)
r2_train_list.append(r2_train)
r2_test_list.append(r2_test)
mse_train_list.append(mse_train)
mse_test_list.append(mse_test)
model_names.append(model_name)

print(f'{model_name} Train    - R^2: {r2_train:.2f}, MAPE: {mape_train:.2f}%,
MSE: {mse_train:.2f}, MAE: {mae_train:.2f}') # Print train evaluation metrics

print(f'{model_name} Test    - R^2: {r2_test:.2f}, MAPE: {mape_test:.2f}%,
MSE: {mse_test:.2f}, MAE: {mae_test:.2f}') # Print test evaluation metrics

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

print(f"Best parameters: {best_params}") # Print the best parameters for the
model

# Print selected features
print("SelectedKBest feature indices and scores (sorted from high to low):")
for feature_idx, score in sorted_features: # Iterate over the sorted features
    print(f"Feature Index: {feature_idx}, Score: {score}") # Print the feature
index and score

print("-" * 50)
Dict_with_Df = {"train": prediction_data_train_df, "test":
prediction_data_test_df} # Create a dictionary with train and test prediction
DataFrames
# Create a DataFrame from the lists
metrics_df = pd.DataFrame({
    'Model': model_names,
    'MAPE Train': mape_train_list,
    'MAPE Test': mape_test_list,
    'MAE Train': mae_train_list,
    'MAE Test': mae_test_list,
    'R2 Train': r2_train_list,
    'R2 Test': r2_test_list,
    'MSE Train': mse_train_list,
    'MSE Test': mse_test_list
}) # Create a DataFrame with evaluation metrics for all models

return best_model, Dict_with_Df, metrics_df # Return the best model,
prediction DataFrames, and metrics DataFrame

```

19) Function name: plot_data

Objective: Plot between actual and predict data

Input: Y predict dictionary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output: Display the plot

```
def plot_data(data_dict): # Input: Y predict dictionary

    plt.rc('font', size=25) # Set the default font size for text to 18

    plt.rc('axes', titlesize=25) # Set the font size for plot titles to 20

    plt.rc('axes', labelsiz=25) # Set the font size for axis labels to 18

    dict_keys = list(data_dict.keys()) # Convert the keys of the input dictionary
'data_dict' to a list

    for key in dict_keys: # Iterate over the keys in the list 'dict_keys'

        key_parts = key.split('_') # Split the current key by the underscore character

        solution = key_parts [0] # The first part of the key represents the solution

        sol_data = data_dict [key] # Retrieve the data for the current key from
'data_dict'

        if key_parts [1] == 'None': # Check if the second part of the key is 'None'

            for data_type, df in sol_data.items(): # Iterate over the data type and
DataFrame pairs

                for i in range(1, df.shape [1]): # Iterate over columns starting from the
second column

                    plt.figure(figsize=(15, 15)) # Create a new figure with a size of 15x15
inches

                    plt.plot(df.iloc[:, 0], df.iloc[:, i], 'o') # Plot the actual values (column
0) against the predictions (column i)

                    model_name = df.columns [i].split('_') [0] # Extract the model name
from the column header
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        set_name = df.columns [i].split('_') [1] # Extract the set name (Train
or Test) from the column header

        if key_parts [0] == 'Carbosulfan': # Check if first part of key is
'Carbosulfan"

            plt.xlim(0, 1600) # Create graph limit x axis at 1600

            plt.ylim(0, 1600) # Create graph limit y axis at 1600

            plt.plot( [0, 1600], [0, 1600], 'k--') # Create 45 degree line

            plt.xlabel('Actual concentration (ppm)') # Set the label for the x-
axis

            plt.ylabel(f"{set_name} set prediction concentration (ppm)") # Set
the label for the y-axis

            plt.title(f"{set_name} Data vs. Actual Value ({model_name}) -
{solution} ({data_type})") # Set the plot title

        else:

            plt.xlim(0, 22) # Create graph limit x axis at 22

            plt.ylim(0, 22) # Create graph limit y axis at 22

            plt.plot( [0, 22], [0, 22], 'k--') # Create 45 degree line

            plt.xlabel('Actual concentration (%w/v)') # Set the label for the x-
axis

            plt.ylabel(f"{set_name} set prediction concentration (%w/v)") #
Set the label for the y-axis

            plt.title(f"{set_name} Data vs. Actual Value ({model_name}) -
{solution} ({data_type})") # Set the plot title

        plt.grid(True) # Show the grid lines

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

plt.show() # Display the plot

else:

    sol_data_keys = list(sol_data.keys()) # Convert the keys of the input
dictionary 'sol_data' to a list

    # This line converts the keys of the 'sol_data' dictionary to a list, which
allows us to iterate over them easily.

    # Iterate over pairs of keys

    for i in range(len(sol_data_keys)-1):

        for j in range(i+1, len(sol_data_keys)):

            solution_mix_1 = sol_data_keys [i] # Get the first solution mix key
            solution_mix_2 = sol_data_keys [j] # Get the second solution mix
key
            # These lines create a nested loop to iterate over pairs of solution
mix keys.

            # Get data for each solution mix

            solution_data_1 = sol_data [solution_mix_1]
            solution_data_2 = sol_data [solution_mix_2]

            # These lines retrieve the data for the two solution mixes from the
'sol_data' dictionary.

            # Iterate over data types

            for data_type, df_1 in solution_data_1.items():

                df_2 = solution_data_2 [data_type]

                # This loop iterates over the data types in the first solution mix
data, and retrieves the corresponding data from the second solution mix.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# Iterate over columns

for col_index in range(1, df_1.shape [1]):

    plt.figure(figsize=(15, 15)) # Create a new figure with a size of
15x15 inches

    plt.plot(df_1.iloc[:, 0], df_1.iloc[:, col_index], 'bo',
label=f'{solution_mix_1}') # Plot the actual values (column 0) against the
predictions (current column) for the first solution mix

    plt.plot(df_2.iloc[:, 0], df_2.iloc[:, col_index], 'ro',
label=f'{solution_mix_2}') # Plot the actual values (column 0) against the
predictions (current column) for the second solution mix

    plt.xlim(0, 13) # Set the x-axis limits to 0-13
    plt.ylim(0, 13) # Set the y-axis limits to 0-13
    plt.plot([0, 13], [0, 13], 'k--') # Plot the 45-degree line
    model_name = df_1.columns [col_index].split('_') [0] # Extract
the model name from the column header
    set_name = df_1.columns [col_index].split('_') [1] # Extract the
set name (Train or Test) from the column header
    plt.xlabel('Actual concentration (%w/v)') # Set the label for the
x-axis

    plt.ylabel(f'{set_name} set prediction concentration (%w/v)') #
Set the label for the y-axis

    plt.title(f'{set_name} Data vs. Actual Value ({model_name}) -
{solution_mix_1} - {solution_mix_2} mixture ({data_type})') # Set the plot title

    plt.grid(True) # Show the grid lines

    plt.legend() # Show the legend

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
plt.show() # Display the plot
```

20) **Function name:** plot_trend

Objective: Plot trend of raw L data

Input: dataframe that not subtract and still have blank image data

Output: Display the plot

```
def plot_trend(df): # Input: dataframe that not subtract and still have blank image
data
    plt.rc('font', size=18) # Set the default font size for text to 18
    plt.rc('axes', titlesize=20) # Set the font size for plot titles to 20
    plt.rc('axes', labelsiz=18) # Set the font size for axis labels to 18
    plot_data = df.drop(columns= ["Light_Source", "Picture_Number"]) # Drop the
"Light_Source" and "Picture_Number" columns from the input DataFrame
    grouped_data_plot = plot_data.groupby( ['Solute_1', 'Concentration_1_ppm',
'Solute_2', 'Concentration_2_ppm', 'Solvent']) # Group the data by 'Solute_1',
'Concentration_1_ppm', 'Solute_2', 'Concentration_2_ppm', and 'Solvent'
    plot_data_group = grouped_data_plot.mean().reset_index() # Calculate the
mean of the grouped data and reset the index
    # Get unique combinations of 'Solute_1' and 'Solute_2' columns
    unique_solute_combinations = plot_data_group [ ['Solute_1',
'Solute_2']].drop_duplicates().values # Get unique combinations of 'Solute_1' and
'Solute_2'
    for solute_1, solute_2 in unique_solute_combinations: # Iterate over the unique
solute combinations
        # Filter data for the current solute combination
        solute_combination_data = plot_data_group [(plot_data_group ['Solute_1']
== solute_1) & (plot_data_group ['Solute_2'] == solute_2)] # Filter the data for the
current solute combination
        plt.figure(figsize=(20, 15)) # Create a new figure with a size of 20x15 inches
        pixels = range(1, 101) # Define a range of pixel values from 1 to 100
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# Plot each concentration's L values against pixel for the current solute
combination

for _, row in solute_combination_data.iterrows(): # Iterate over rows in the
solute combination data

    concentration_1 = row ['Concentration_1_ppm'] # Get the concentration
value for solute 1

    concentration_2 = row ['Concentration_2_ppm'] # Get the concentration
value for solute 2

    solvent = row ['Solvent'] # Get the solvent value

    columns = [col for col in solute_combination_data.columns if col not in
['Solute_1', 'Concentration_1_ppm', 'Solute_2', 'Concentration_2_ppm', 'Solvent']] #
Get the column names for L values

    l_values = row [columns] # Get the L values for the current row

    label =
f'{solute_1}_{concentration_1}_ppm_{solute_2}_{concentration_2}_ppm_{solvent}'
# Create a label for the plot

    plt.plot(pixels, l_values, label=label) # Plot the L values against pixels
with the corresponding label

    # Plot the 'Blank' series for the same solvent

    blank_data = plot_data_group [(plot_data_group ['Solute_1'] == 'Blank') &
(plot_data_group ['Solvent'] == solvent)] # Filter the data for 'Blank' solute and the
current solvent

    if not blank_data.empty: # Check if the blank data is not empty

        for _, row in blank_data.iterrows(): # Iterate over rows in the blank data

            concentration_1 = row ['Concentration_1_ppm'] # Get the
concentration value for solute 1 (should be 0 for 'Blank')

            concentration_2 = row ['Concentration_2_ppm'] # Get the
concentration value for solute 2 (should be 0 for 'Blank')

            columns = [col for col in blank_data.columns if col not in ['Solute_1',
'Concentration_1_ppm', 'Solute_2', 'Concentration_2_ppm', 'Solvent']] # Get the
column names for L values

            l_values = row [columns] # Get the L values for the current row

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

label =
f"Blank_{concentration_1}_ppm_{concentration_2}_ppm_{solvent}" # Create a label
for the 'Blank' plot

plt.plot(pixels, l_values, label=label) # Plot the 'Blank' L values against
pixels with the corresponding label

plt.xlabel('Pixel') # Set the label for the x-axis
plt.ylabel('L Value') # Set the label for the y-axis
plt.title(f'L Value vs Pixel for Different Concentrations (Solute_1: {solute_1},
Solute_2: {solute_2})') # Set the plot title

plt.legend() # Show the legend
plt.show() # Display the plot

```

21) **Function name:** plot_trend_diff_L

Objective: Plot data trend after subtract

Input: Data_For_train

Output: Display the plot

```

def plot_trend_diff_L(df):
plt.rc('font', size=18) # Set the default font size for text to 18
plt.rc('axes', titlesize=20) # Set the font size for plot titles to 20
plt.rc('axes', labelsz=18) # Set the font size for axis labels to 18
plot_data = df.drop(columns= ["Light_Source", "Picture_Number"]) # Drop the
"Light_Source" and "Picture_Number" columns from the input DataFrame
grouped_data_plot = plot_data.groupby( ['Solute_1', 'Concentration_1_ppm',
'Solute_2', 'Concentration_2_ppm', 'Solvent']) # Group the data by 'Solute_1',
'Concentration_1_ppm', 'Solute_2', 'Concentration_2_ppm', and 'Solvent'
plot_data_group = grouped_data_plot.mean().reset_index() # Calculate the
mean of the grouped data and reset the index

# Get unique combinations of 'Solute_1' and 'Solute_2' columns
unique_solute_combinations = plot_data_group [ ['Solute_1',
'Solute_2']].drop_duplicates().values # Get unique combinations of 'Solute_1' and
'Solute_2'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for solute_1, solute_2 in unique_solute_combinations: # Iterate over the unique
solute combinations

    # Filter data for the current solute combination
    solute_combination_data = plot_data_group [(plot_data_group ['Solute_1']
== solute_1) & (plot_data_group ['Solute_2'] == solute_2)] # Filter the data for the
current solute combination

    plt.figure(figsize=(20, 15)) # Create a new figure with a size of 20x15 inches
    pixels = range(1, 101) # Define a range of pixel values from 1 to 100
    # Plot each concentration's L values against pixel for the current solute
combination

    for _, row in solute_combination_data.iterrows(): # Iterate over rows in the
solute combination data
        concentration_1 = row ['Concentration_1_ppm'] # Get the concentration
value for solute 1
        concentration_2 = row ['Concentration_2_ppm'] # Get the concentration
value for solute 2
        solvent = row ['Solvent'] # Get the solvent value
        columns = [col for col in solute_combination_data.columns if col not in
['Solute_1', 'Concentration_1_ppm', 'Solute_2', 'Concentration_2_ppm', 'Solvent']] #
Get the column names for L values
        l_values = row [columns] # Get the L values for the current row
        label =
f'{solute_1}_{concentration_1}_ppm_{solute_2}_{concentration_2}_ppm_{solvent}'
# Create a label for the plot
        plt.plot(pixels, l_values, label=label) # Plot the L values against pixels
with the corresponding label

        plt.xlabel('Pixel') # Set the label for the x-axis
        plt.ylabel('L Value') # Set the label for the y-axis
        plt.title(f'L Value vs Pixel for Different Concentrations (Solute_1: {solute_1},
Solute_2: {solute_2})') # Set the plot title
        plt.legend() # Show the legend
        plt.show() # Display the plot

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการรวม. การใช้ฟังก์ชัน และการจบ loop ลบรูป

```

Local_minimum_each_image_Df = pd.DataFrame() # Initialize an empty
DataFrame to store local minimum data
min_column_number_df = pd.DataFrame() # Initialize an empty DataFrame to
store column numbers
Image_name_data_df_loop = [] # Initialize an empty list to store image name data
DataFrames

result_data = pd.DataFrame() # Initialize an empty DataFrame to store results
result_datafull_Blank = pd.DataFrame() # Initialize an empty DataFrame
result_anal_datafull = pd.DataFrame() # Initialize an empty DataFrame
for filename in os.listdir(INPUT_DIR):
    img_path = os.path.join(INPUT_DIR, filename)
    img = cv2.imread(img_path)

    try: # Try to execute the following code
        corners = find_corners(img) # Find the corners of the image using the
find_corners function
        distances = Linedistance(corners) # Calculate the distances between corners
using the Linedistance function
        reordered_corners = corner_reorder(corners) # Reorder the corners using the
corner_reorder function, passing corners as an argument
        fiveparam = FiveParam(reordered_corners) # Compute the five parameters
(center, width, height, angle, and aspect ratio) using the FiveParam function
        center = fiveparam [0] # Extract the center from the FiveParam result
        angle = fiveparam [3] # Extract the angle from the FiveParam result
        Air_crop_img, Air_crop_box = Crop_by_avg(img) # Crop the image using the
Crop_by_avg function
        marked_crop_img = Air_crop_img.copy() # Create a copy of the cropped
image for marking

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cv2.drawContours(marked_crop_img, [reordered_corners], 0, (0, 255, 0), 2) #
Draw the reordered corners on the marked cropped image
output_path_marked1 = os.path.join(OUTPUT_Crop_MARKED,
f"marked_{filename}") # Construct the output path for the marked cropped image
cv2.imwrite(output_path_marked1, Air_crop_img) # Save the cropped image
as "marked_{filename}" in the OUTPUT_Crop_MARKED directory
WARP_crop_img = Air_crop_img.copy() # Create a copy of the cropped image
for warping
warped_Air_image =
warp_perspective_new(WARP_crop_img,reordered_corners) # Warp the cropped
image using the warp_perspective_new function
output_path_warp = os.path.join(OUTPUT_Crop_WARP, f"warp_{filename}") #
Construct the output path for the warped image
#cv2.imwrite(output_path_warp, warped_Air_image) # Commented out code
for saving the warped image
crop_box = np.array( [Air_crop_box [0],Air_crop_box [1]]) # Create a numpy
array from the crop box coordinates
new_reordered_corners = crop_box + reordered_corners # Add the crop box
coordinates to the reordered corners
new_center = crop_box + center # Add the crop box coordinates to the
center
center_x, center_y = int(new_center [0, 0]), int(new_center [0, 1]) # Extract
the x and y coordinates of the new center
img_real_for_center = img.copy() # Create a copy of the original image for
marking the center
cv2.drawContours(img_real_for_center, [new_reordered_corners], 0, (0, 255,
0), 2) # Draw the new reordered corners on the marked image
cv2.circle(img_real_for_center, (center_x, center_y), radius=1, color=(0, 255,
0), thickness=3) # Draw a circle at the new center on the marked image
output_path_marked = os.path.join(OUTPUT_REAL_MARK,
f"marked_{filename}") # Construct the output path for the marked image
cv2.imwrite(output_path_marked, img_real_for_center) # Save the marked
image with the new center and reordered corners

```

```

Warp_add_Air_image = condition(warped_Air_image) # Apply the condition
function to the warped image
    output_path_warp_add = os.path.join(output_crop_Warp_add,
f"warp_{filename}") # Construct the output path for the conditioned warped image
    cv2.imwrite(output_path_warp_add, Warp_add_Air_image) # Save the
conditioned warped image
    Full_image = img.copy() # Create a copy of the original image
    image_name = os.path.splitext(filename) [0] # Extract the image name
without extension
    parts = image_name.split('_') # Split the image name into parts using
underscores
    Picture_number = parts [7] # Extract the picture number from the parts list
    Concentration_1_ppm = parts [4] # Extract the first concentration value from
the parts list
    Concentration_2_ppm = parts [6] # Extract the second concentration value
from the parts list
    solvent = parts [2] # Extract the solvent information from the parts list
    solute_1 = parts [3] # Extract the first solute information from the parts list
    solute_2 = parts [5] # Extract the second solute information from the parts
list
    Light_Source = parts [1] # Extract the light source information from the parts
list
    Image_name_data = {'Light_Source': [Light_Source], 'Solute_1':
[solute_1], 'Solute_2': [solute_2], 'Solvent': [solvent], 'Picture_Number':
[Picture_number], 'Concentration_1_ppm':
[Concentration_1_ppm], 'Concentration_2_ppm': [Concentration_2_ppm] } # Create
a dictionary with image name data
    Image_name_data_df = pd.DataFrame(Image_name_data) # Convert the
image name data dictionary to a DataFrame
    Image_name_data_df_loop.append(Image_name_data_df) # Append the
image name data DataFrame to the loop list
    if angle > 0: # If the angle is positive (counter-clockwise rotation)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

distance_y = -np.tan(angle*rotate_counter_clockwise)*distance_x +
more_red # Calculate the y distance based on the angle and other parameters
angle = angle + angle # Double the angle value
else: # If the angle is negative (clockwise rotation)
distance_y = np.tan(angle+rotate_clockwise)*distance_x - more_red #
Calculate the y distance based on the angle and other parameters
if angle == 0: # If the angle is zero (no rotation)
distance_y = -np.tan(angle*rotate_counter_clockwise)*distance_x #
Calculate the y distance based on the angle and other parameters
andgle = angle + perfect_morph # Adjust the angle value
distance_y_new = -6 # Set a new value for the y distance
angle = angle *180/np.pi # Convert the angle from radians to degrees
angle = angle + angle_adjustment # Apply an adjustment to the angle value
angle = angle/(180/np.pi) # Convert the angle back from degrees to radians
ref_distance = np.array( [ [distance_x,distance_y]] ) # Create a numpy array
with the x and y distances
new_Anal_center = new_center + ref_distance # Calculate the new analysis
center by adding the reference distance to the new center
new_Anal_Position = np.intp(CenterToFiveParamFormRef(new_Anal_center,
new_reordered_corners, angle, height_solution_ratio, width_solution_ratio,
more_red)) # Calculate the new analysis position using the
CenterToFiveParamFormRef function
img_Anal_real = img_real_for_center.copy() # Create a copy of the marked
image for analysis
cv2.drawContours(img_Anal_real, [new_Anal_Position], 0, (0, 255, 0), 2) #
Draw the new analysis position on the marked image
output_path_marked_anal = os.path.join(OUTPUT_Anal_real,
f"marked_{filename}") # Construct the output path for the marked analysis image
cv2.imwrite(output_path_marked_anal, img_Anal_real) # Save the marked
analysis image
anal_distances = Linedistance(new_Anal_Position) # Calculate the distances
for the analysis position using the Linedistance function
img_real_anal = img.copy() # Create a copy of the original image for analysis

```

```

warped_image_anal = warp_perspective_new(img_real_anal,
new_Anal_Position) # Warp the analysis image using the warp_perspective_new
function

output_path_warp_anal = os.path.join(OUTPUT_ANAL_WARP,
f"warp_{filename}") # Construct the output path for the warped analysis image

cv2.imwrite(output_path_warp_anal, warped_image_anal) # Save the warped
analysis image

Local_minimum_each_image_Df, min_column_number_df, median_minimum
= local_minimum_find(warped_image_anal, image_name,
Local_minimum_each_image_Df, min_column_number_df) # Find the local
minimum using the local_minimum_find function

image_data = lightness_adjust(Full_image, image_name) # Adjust the
lightness of the full image using the lightness_adjust function

result_data = pd.concat( [result_data, image_data], ignore_index=True) #
Concatenate the image data to the result_data DataFrame

print(f"Processed, marked, mark real, and warped: {output_path_marked},
{OUTPUT_REAL_MARK} ,{output_path_warp}") # Print a message indicating
successful processing, marking, and warping

except ValueError as e: # If a ValueError occurs during processing

print(f"Error processing {filename}: {e}") # Print an error message with the
filename and error description

for filename_warp_img in os.listdir(OUTPUT_ANAL_WARP): # Iterate over filenames
in the OUTPUT_ANAL_WARP directory

img_path = os.path.join(OUTPUT_ANAL_WARP, filename_warp_img) # Construct
the full image path

warp_img = cv2.imread(img_path) # Read the warped image

warped_image_anal_add,x_shift =
Shift_image_median_minimum_add_condition(warp_img,median_minimum) # Call
the Shift_image_median_minimum_add_condition function

image_namefull_anal = os.path.splitext(filename) [0] # Extract the image name
without extension

image_datafull_anal = Average_lightness(warped_image_anal_add,
image_namefull_anal) # Call the Average_lightness function

```

```

result_anal_datafull = pd.concat([result_anal_datafull, image_datafull_anal],
ignore_index=True) # Concatenate image_datafull_anal to result_anal_datafull
warp_img_crop_add,x_shift =
Shift_image_median_minimum_add_condition(warp_img,median_minimum) # Call
the Shift_image_median_minimum_add_condition function
print(filename_warp_img) # Print the current filename
output_path_warp_adjust = os.path.join(OUTPUT_ANAL_WARP_add,
f"marked_{filename_warp_img}") # Construct the output path for the marked
image
cv2.imwrite(output_path_warp_adjust, warp_img_crop_add) # Save the marked
image

Image_name_data_df_full = pd.concat(Image_name_data_df_loop,
ignore_index=True) # Concatenate all DataFrames in Image_name_data_df_loop
into a single DataFrame

image_name_df = result_data.iloc[:, [0]] # Extract the first column (image names)
from result_data
result_data_noname = result_data.drop(result_data.columns [0], axis=1) # Drop
the first column from result_data
result_data_noname_average = result_data_noname.mean(axis=1) # Calculate the
row-wise mean of result_data_noname
median_data = result_data_noname.mean(axis=0) # Calculate the column-wise
mean of result_data_noname
median_data = median_data.values # Convert median_data to a numpy array
median_data = np.median(median_data,axis=0) # Calculate the median across all
columns

adjust_data = median_data - result_data_noname_average # Assuming
adjust_data is a pandas Series
adjust_data_array = adjust_data.values # Convert adjust_data to a numpy array
reshaped_adjust_data = adjust_data_array[:, np.newaxis] # Reshape
adjust_data_array to a column vector

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

#Anal
result_data_full_noname_anal = result_anal_datafull.drop(result_data.columns [0],
axis=1) # Drop the first column from result_anal_datafull
result_data_full_noname_anal = result_data_full_noname_anal.values # Convert
result_data_full_noname_anal to a numpy array
new_result_data_anal = result_data_full_noname_anal + reshaped_adjust_data #
Add reshaped_adjust_data to result_data_full_noname_anal

new_result_data_anal_df = pd.DataFrame(new_result_data_anal) # Convert
new_result_data_anal to a DataFrame
Crude_result_df = new_result_data_anal_df #new_result_data_Blank_df # Assign
new_result_data_anal_df to Crude_result_df
Crude_result_df = Crude_result_df.astype(float) # Convert Crude_result_df to float
type
What_happen1 = Crude_result_df.insert(0,"Image_Name",image_name_df) # Insert
a new column "Image_Name" with image_name_df data

DATA_FOR_not_done = Crude_result_df # Assign Crude_result_df to
DATA_FOR_not_done
DATA_FOR_not_done.columns = image_datafull_anal.columns.values # Update
column names of DATA_FOR_not_done
DATA_FOR_not_done = DATA_FOR_not_done.drop(DATA_FOR_not_done.columns
[0], axis = 1) # Drop the first column from DATA_FOR_not_done
DATA_FOR_not_done = pd.concat( [Image_name_data_df_full,
DATA_FOR_not_done], axis = 1) # Concatenate Image_name_data_df_full with
DATA_FOR_not_done

DATA_with_Solute = DATA_FOR_not_done [DATA_FOR_not_done ['Solute_1'] !=
"Blank"] # Filter rows where 'Solute_1' is not "Blank"
DATA_BLANK = DATA_FOR_not_done [DATA_FOR_not_done.Solute_1 == "Blank"]
# Filter rows where 'Solute_1' is "Blank"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DATA_BLANK = DATA_BLANK.drop(columns =
["Picture_Number","Concentration_1_ppm","Solute_2","Concentration_2_ppm"]) #
Drop specified columns from DATA_BLANK
grouped_data = DATA_BLANK.groupby( ['Light_Source', 'Solute_1', 'Solvent',]) #
Group DATA_BLANK by specified columns
DATA_BLANK_grouped_average = grouped_data.median().assign(median=lambda x:
x.median()).drop(columns = ["median"]) # Calculate median for each group and
assign a 'median' column
DATA_BLANK_grouped_average = DATA_BLANK_grouped_average.reset_index() #
Reset the index of DATA_BLANK_grouped_average

DATA_FOR_TRAIN = DATA_with_Solute.copy() # Create a copy of
DATA_with_Solute and assign it to DATA_FOR_TRAIN
for index, row in DATA_BLANK_grouped_average.iterrows(): # Iterate over rows in
DATA_BLANK_grouped_average
    mask = (DATA_with_Solute ['Light_Source'] == row ['Light_Source']) & \
            (DATA_with_Solute ['Solvent'] == row ['Solvent']) # Create a mask based
on 'Light_Source' and 'Solvent'
    for col in DATA_with_Solute.filter(like='pixel').columns: # Iterate over columns
containing 'pixel'
        DATA_FOR_TRAIN.loc [mask, col] = row [col] - DATA_with_Solute.loc [mask,
col] # Perform operations on DATA_FOR_TRAIN

substance_dfs = create_substance_dfs(DATA_FOR_TRAIN) # Call the
create_substance_dfs function
plot_trend(DATA_FOR_not_done) # Call the plot_trend function
plot_trend_diff_L(DATA_FOR_TRAIN) # Call the plot_trend_diff_L function
best_model_pure,best_model_mixture,dict_combined, metric_combined =
pure_mixture_seperation(substance_dfs) # Call the pure_mixture_seperation
function
Crude_result_df.to_excel(r"C:\Users\ACER\OneDrive\Desktop\analyzee.xlsx",
sheet_name='Sheet2', index=False) # Save Crude_result_df to an Excel file
plot_data(dict_combined) # Call the plot_data function

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

metric_Sugar_Salt = metric_combined ["Sugar_Salt"]
metric_Salt = metric_Sugar_Salt ["Salt"]
metric_Sugar = metric_Sugar_Salt ["Sugar"]
metric_Sugar_Salt_df = pd.DataFrame({
    'Model': metric_Salt ['Model'],
    'MAPE Train': (metric_Salt ['MAPE Train'] + metric_Sugar ['MAPE Train']) / 2,
    'MAPE Test': (metric_Salt ['MAPE Test'] + metric_Sugar ['MAPE Test']) / 2,
    'MAE Train': (metric_Salt ['MAE Train'] + metric_Sugar ['MAE Train']) / 2,
    'MAE Test': (metric_Salt ['MAE Test'] + metric_Sugar ['MAE Test']) / 2,
    'R2 Train': (metric_Salt ['R2 Train'] + metric_Sugar ['R2 Train']) / 2,
    'R2 Test': (metric_Salt ['R2 Test'] + metric_Sugar ['R2 Test']) / 2,
    'MSE Train': (metric_Salt ['MSE Train'] + metric_Sugar ['MSE Train']) / 2,
    'MSE Test': (metric_Salt ['MSE Test'] + metric_Sugar ['MSE Test']) / 2
})
del metric_combined ["Sugar_Salt"]
metric_combined ["Sugar_Salt"] = metric_Sugar_Salt_df
end_time = time.time() # Record the end time
execution_time = end_time - start_time # Calculate the execution time
print("Run time: {:.2f} seconds".format(execution_time)) # Print the execution time

```

สามารถเข้าถึงข้อมูลเพิ่มเติมได้ที่ : https://github.com/TonNaphon/SpecPestML_Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้